

RESTRICTING SIDEWARD MOVEMENT:
THE COURSE OF THE DERIVATION AND THE DIRECTIONALITY OF MOVEMENT

By

Joseph Jalbert

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Linguistics-Doctor of Philosophy

2016

ABSTRACT

RESTRICTING SIDEWARD MOVEMENT: THE COURSE OF THE DERIVATION AND THE DIRECTIONALITY OF MOVEMENT

By

Joseph Jalbert

In this dissertation, an account is provided of the restrictions on sideward movement through normal derivational processes and well-motivated principles of the computational system. A syntactic operation such as sideward movement is powerful within a theory, so it must be properly restricted to avoid overgeneration, specifically regarding island constraints on movement. Parasitic gap constructions (Nunes 1995, 2001, 2004, 2012; Haida and Repp 2012) and relative clauses (Nunes 2004, Henderson 2007) have both been identified as structures which might be better analyzed using sideward movement. These analyses give a view of the explanatory power that sideward movement can provide, but also highlight the need to be able to restrict such an operation.

Chapter 2 examines previous accounts of sideward movement in parasitic gap constructions, and discusses how these theories fail to account for some of the restrictions necessary for sideward movement. In addition, the restrictions these theories do have can be shown to suffer from looking ahead in the derivation, as well as a lack of sufficient motivation from aspects of the computational system. In chapter 3 a new analysis for sideward movement in parasitic gaps is proposed which can solve these problems. This analysis of sideward movement is built upon the requirements of the cyclic, general application of Spell-out, which can be motivated independently through principles of efficient computation. From these principles, potential phase architectures are explored, where the timing and triggers of Spell-out are varied, including what syntactic categories must serve as phase heads. After removing those

architectures in which adjunct islandhood is predicted to be violated, as well as those in which a sideward movement account of parasitic gaps is impossible due to Spell-out, only one possible phase architecture remains. Within this architecture, restrictions on parasitic gap extractions are derived from cyclic application of Spell-out on phases. Additionally, extraction from adjuncts within the parasitic gap construction is ruled out by the demands on the course of the derivation and inaccessibility through the generic application of the Spell-out operation. Chapter 4 examines how several structural restrictions on parasitic gaps can be implemented in this new system using Nunes' (2004) Form Chain operation. An account of the asymmetries between parasitic gap constructions and Across-the-Board constructions is provided following Munn's (2001) account that parasitic gap constructions are essentially weak islands to movement due to the semantic properties of the prepositional phrase. Lastly, chapter 5 shows how these principles can be extended to understanding the derivation of relative clauses, another area in which sideward movement has been suggested to play a role (Nunes 2004, Henderson 2007). A unique analysis of the derivation of relative clauses is provided which incorporates principles of sideward movement, Sportiche's (2016) theory of the operation Neglect, and new data regarding reconstruction into relative clauses. Taken as a whole, this dissertation shows how our theory of the syntactic computation can be enriched with more explanatory power when the processes of the derivation are atomized and are well-motivated by principles of efficient computation.

ACKNOWLEDGMENTS

While my name is on the front page, anyone who has written a dissertation (or any large project) knows that this is a large undertaking involving far more than just one person. Many people helped me examine the data, helped me understand the concepts at hand, helped me learn the tools of linguistic analysis, helped me learn to think, and helped me survive the stresses and trials of graduate school. I am grateful to the institutions that have allowed me to complete this dissertation, but more importantly, I want to thank the individuals that have helped make this research possible.

First, I would like to thank my advisor Alan Munn, and my committee, Cristina Schmitt, Marcin Morzycki, and Suzanne Wagner. This dissertation is better for them having advised it. Their feedback over my graduate school career has been critical to my growth as a linguist and as a researcher. I first became interested in linguistics in Cristina's Introduction to Linguistics class, and she gave me the opportunity to grow that interest in her lab so many years ago as an undergraduate. Alan has helped me develop a critical analytical perspective that has served me well, both in research and more broadly. While the failings of this dissertation are my own, they are significantly less because of the members of my committee, and for that I am grateful.

Next, I would like to thank all my colleagues and friends in linguistics, who have helped me grow as a linguist and a person over the years. Specifically, I want to thank Drew Trotter, Karthik Durvasula, Alan Beretta, Greg Johnson, and Kali Morris. Thank you, Drew, for being a great friend from the beginning to the grad school to the end. Thank you, Karthik, for being a constant friend and argument sparring partner. Thank you, Alan, for helping me develop skills that will be of use to me for a long time and helping me find motivation when I truly had none.

Thank you, Greg and Kali, for all the fun times and all the long syntax talks over beers. And truly thank you to the rest of the graduate students at MSU for helping make life that much easier and bearable.

Next, I want to thank my friends that helped keep me sane throughout the process, especially Chris Wickman, Aaron Benmark, Austin Drouare, Dave Porath, and Jess Cole. Thanks for helping me deal with the stress, be it through having fun or long talks. Your friendships have meant so much to me.

I also want to thank my family, who has always been so supportive of me. Thank you to my mom, who frequently called to check up on me and see how things were going. Thank you for always being understanding of the changing timelines and uncertainties that come with graduate school. From a young age, she helped me make the most of my education and opportunities, and graduate school has been no exception. Thank you to my dad, who always showed an interest in what I was doing, even if I didn't do a very good job of explaining it for a long time. He inspired a lot of the curiosity I have in science and the world around me. Thank you to everyone else in my family, who always cared about what I was doing, hoped for the best for me, and were a wonderful support system that has allowed to me go out and do this research.

Lastly, I want to thank my wife, Ho-Hsin Huang. I am truly blessed to have been with someone who understands the stresses of graduate school, who I can share linguistics talks with, who can accept me for who I am, while simultaneously helping me become better. Thank you for all the support you have given me, thank you for pushing me, you for accepting me, and thank you for being by my side.

TABLE OF CONTENTS

LIST OF TABLES.....	viii
KEY TO ABBREVIATIONS.....	ix
1 Introduction.....	1
1.1 Building a theory from scratch	1
1.2 Charting the path.....	4
2 Minimalism and previous theories of parasitic gap constructions.....	7
2.1 Introduction	7
2.2 The Minimalist Program	8
2.2.1 The optimal solution	8
2.2.2 Evaluating the optimal solution	14
2.2.3 Minimalist conceptions of Merge and sideward movement	15
2.3 Parasitic gaps as a case study.....	22
2.3.1 Adjunct islands and parasitic gaps	22
2.3.2 Nunes on Parasitic Gaps	30
2.3.3 Haida and Repp on parasitic gaps	47
2.4 Conclusion	53
3 The course of the derivation and restricting sideward movement	55
3.1 Introduction	55
3.2 Problems with previous attempts	59
3.2.1 Using availability to restrict movement.....	62
3.2.2 Restricting movement out of adjunct islands.....	75
3.2.2.1 Agree and A' movement.....	75
3.3 Triggers and targets of Spell-out.....	77
3.3.1 Type 1 failure: No cyclic computation	82
3.3.2 Type 2 failure: Movement out of adjuncts.....	84
3.3.3 Type 3 failure: Movement out of adjuncts predicted.....	92
3.4 Justifying prepositions as a phase-heads.....	95
3.5 Applying the phase architecture.....	100
3.5.1 Deriving adjunct islands and parasitic gaps.....	101
3.5.2 Accounting for islands within the parasitic domain	105
3.5.2.1 Subject islands and Spell-out	109
3.6 Conclusion	114
4 Accounting for parasitic gap restrictions	117
4.1 Introduction	117
4.2 Structural restrictions on parasitic gaps	118
4.2.1 S-Structure licensing of parasitic gaps.....	120
4.2.2 C-command condition on parasitic gaps.....	123

4.2.3	Restriction against A-chains licensing parasitic gaps	129
4.2.4	Summary of structural restrictions	130
4.3	The NP restriction on parasitic gaps	132
4.4	ATB constructions and sideward movement	135
4.5	Conclusion	144
5	Relative clauses and sideward movement.....	146
5.1	Introduction	146
5.2	Reconstruction and the Copy Theory of Movement.....	148
5.2.1	Anti-reconstruction and late adjunction.....	149
5.3	Theories of relative clauses.....	151
5.3.1	The raising paradox and sideward movement	155
5.4	Neglect as an alternate account.....	161
5.4.1	Pseudo-complements and reconstruction.....	164
5.5	Implications for deriving relative clauses	168
5.5.1	Accounting for islands within the relative clause	178
5.5.2	Summarizing relative clause structure	183
5.6	Using Neglect for raising and matching effects.....	184
5.7	Conclusion	193
6	Conclusion	194
6.1	The theory as it stands.....	194
6.2	Contextualizing this research.....	202
	REFERENCES	205

LIST OF TABLES

Table 1: The four-dimensional possibility space for phase architectures	81
Table 2: Phase architectures eliminated due to a lack of cyclic computation	84
Table 3: All phase architectures eliminated so far, including those eliminated from predicting movement out of adjunct islands (without P phase-heads)	90
Table 4: All phase architectures eliminated so far, including those eliminated from predicting movement out of adjunct islands (with P phase-heads)	92
Table 5: All phase architectures eliminated so far, including those eliminated from predicting movement out of adjunct islands (with P phase-heads)	96

KEY TO ABBREVIATIONS

ATB	Across the Board
Acc	Accusative (case)
CED	Conditions on Extraction Domains
CI	Conceptual-Intentional
Dat	Dative (case)
LA	Lexical Array
LCA	Linear Correspondence Axiom
LF	Logical Form
MoM	Merge over Move
Nom	Nominative (case)
Num	Numeration
PF	Phonological Form
PIC	Phase Impenetrability Condition
SC	Small clause
Top	Topic

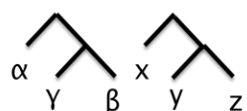
1 Introduction

1.1 Building a theory from scratch

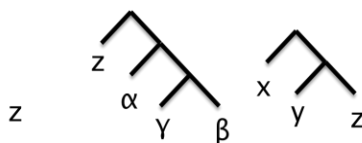
This dissertation is a case study how the vast amount of data, descriptions and analyses of syntactic phenomena can be more deeply understood through the methods of the Minimalist Program. Specifically, I will be looking at sideward movement (Nunes 1995, 2001, 2004), which is when a moved object does not move to a higher, c-commanding position in the structure being built, but instead moves ‘sideways’ to extend another structure being built in the derivational workspace. This can be seen in the example in (1), where we see z copied from within one structure in (1a) and merged to another structure in (1c).

(1) Numeration: $\{\alpha, \beta, \gamma, x, y, z, \dots\}$

a. Initial state



b. Select c. Merge



Such an operation is inherently powerful and should invoke a certain amount of conceptual concern for overgeneration. However, I will show that this operation can be reasonably constrained using well-motivated assumptions about the operations of the computational system, its interaction with the phonological and logical interfaces (LF and PF), and the constraints imposed upon the computational system by virtue of being part of a general cognitive mechanism that abides by principles of computational efficiency.

Methodologically, this dissertation will look at what can be understood by taking atomic operations of the syntactic computation and well-motivated assumptions about the system that surrounds it in order to obtain a deeper understanding of the phenomena at hand. Importantly, the goal is not primarily to obtain more empirical coverage or even come to a better description of the data. The measure of success here will be advancing the ability to understand how a handful of basic operations and assumptions about the grammar can account for the phenomena under consideration. Most of the data here is not new, and there are a great number of descriptions and analyses of that data. There is a great deal of value in finding new data and providing new descriptions and analyses of that data, to be sure. However, from those analyses and descriptions, we should ask another question. Why are these the descriptions and analyses that hold? It is this question I will focus on.

The Minimalist Program pursues the goal of finding deeper understanding by seeking fundamental, atomic parts of our theory and seeking to show that many, if not all of the analyses and descriptions follow from those atomic parts. One of the key aspects of the Minimalist Program can be summarized in what Chomsky (2001) calls the Strong Minimalist Thesis. This thesis states that the faculty of language that humans possess is in fact the optimal solution to the problem of language and its interaction with the cognitive systems it interfaces with. One result of this mode of inquiry is the understanding that a minimal conception of the Merge operation can be used to understand movement. Nunes (1995, 2001, 2004), Hornstein & Nunes (2002, 2008), and Hornstein (2001), and Boeckx et al. (2010) argue that this conception of Merge entails sideward movement as a possibility

Specifically, I will be looking at the predictions made by this minimalist conception of the Merge operation, and well as the Spell-out operation, the efficiency constraints placed on the

derivation that motivates cyclic phasal computation, and the operations which Nunes (2004, 2012) proposes govern the creation of chains and their pronunciations, Form Chain and Chain Reduction.

In order to show the predictive and explanatory ability of these atomic operations, as well as the ability for them to constrain one another, I examine several domains in which sideward movement has been used as an analysis. These include parasitic gap construction, Across-The-Board (ATB) constructions, and relative clauses. I will show that sideward movement provides us with a deeper understanding of these constructions, as well as better empirical coverage.

In showing how sideward movement can be restricted to analyze parasitic gap constructions, I will also show how the same constraints that restrict sideward movement can be used to account for adjunct islands. I will show that adjunct islands are the result of normal, cyclic applications of Spell-out, triggered by the introduction of phase heads into the derivation. Nunes' sideward movement theories of parasitic gap and ATB constructions uses sideward movement to obviate island effects through movement into another active phase. This happens in a manner very similar to typical cyclic WH movement through phase edges, which obviates the moved objects inaccessibility within spelled out material. These processes are restricted by the types of chains that are able to be created through the operation Form Chain. The creation of chains is crucially separated from the operations that are associated with movement. I will show that this separation is conceptually advantageous and provides us a better understanding of the phenomena at hand.

I will show that sideward movement and Spell-out are restricted by well-motivated principles of computation and by the inability of the computational system to look ahead at the

structure that will eventually be built. I will show that many of the restrictions proposed on sideward movement by Nunes (2004, 2012) require that the computational system look ahead in order to apply operations that will prevent overgeneration. I will argue that this cannot be allowed, and provide alternative explanations for the same phenomena using only the information at hand to build up the structures required. One example of this can be found in the directionality of sideward movement. Nunes argues that sideward movement can only bring objects from more embedded environments to less embedded environments. This is only reasonable if the relative embeddedness of two parts of a structure can be known. If the computational system has access to this information, then some other system has made a structure for the computational system, and the computation is redundant. I will account for the directionality of movement we observe in parasitic gap constructions by proposing the Spell-out of redundant moved material. Since this analysis does not explicitly restrict the directionality of movement generally, it predicts that under the right conditions, we might observe sideward movement that moves objects from less embedded structures to more embedded structures. I will demonstrate that this prediction is borne out in the domain of relative clause construction.

1.2 Charting the path

Chapter 2 will explore the restrictions on sideward movement by looking at the data surrounding parasitic gap constructions and their restrictions. This will include examining the details of Nunes' (2004, 2012) theory of parasitic gaps, as well as Haida and Repp's (2012) criticisms. Nunes shows that a great many of the restrictions that are required for sideward movement come from basic assumptions about motivating movement generally, the nature of the computational system, and the derivational workspace. However, I will show that there are conceptual and empirical problems with both Nunes' and Haida and Repp's sideward movement

theories of parasitic gap constructions. These problems largely relate to their interaction with movement islands produced by adjuncts.

Chapter 3 will focus on addressing the concerns raised with previous analyses of parasitic gap constructions. I will explore how an operation like sideward movement can be restricted largely through the use of a phase architecture that provides a derivation of parasitic gaps while also accounting for adjunct islands through the cyclic application of general Spell-out rules instead of specific rules for Spell-out that target structures that will prove problematic later in the derivation. In this way, I will show that these facts of parasitic gaps and adjunct islands can be accounted for without lookahead or explicit rules targeted at adjunct islands. I will also argue that this must be accomplished through cyclic movement within the parasitic domain, going against Nunes' account of islands within parasitic gap constructions.

Chapter 4 will show that Nunes' Form Chain can still account for the various structural conditions that exist on the formation of parasitic gap constructions. Given the departure from Nunes' theory of the structure of parasitic gaps, it is necessary to ensure that these restrictions are still accounted for. I will show that the atomized nature of Nunes' Form Chain operation is responsible for the ability to capture these structural requirements. This further enhances the argument that breaking down the theory into its essential, atomic parts can leave us with more explanatory and deeper explanations for the phenomena we investigate. In addition, an account of the restriction against non-referential NPs in parasitic gap constructions will be presented. This will be done through a comparison with ATB constructions, where a sideward movement analysis will also be shown to apply. In this analysis of ATB constructions, I will argue against Nunes' appeal toward motivating sideward movement through a knowledge of the conjoined nature of the eventual structure ahead of time. I will show that ATB constructions allow for free

sideward movement between the two conjuncts, and that the NP restrictions in parasitic gap constructions can be accounted for under Munn's (2001) theory of parasitic gap restrictions. Munn argues that parasitic gaps are weak islands, and therefore the NP restrictions we find can be understood under Szabolcsi and Zwarts' (1997) theory of weak islands.

Chapter 5 will extend the theory developed to another domain in which sideward movement has been used in the past, relative clauses. While many previous analyses have relied on late adjunction to account for the data surrounding relative clause structure, Sportiche (2016) argues that these phenomena can be better understood as a result of Neglect, an operation that deletes structural material at the interfaces. I will adopt Sportiche's theory of Neglect, but I will also show that sideward movement is necessary to properly account for the reconstruction facts that surround 'picture phrases', or what I will refer to as pseudo-complements. The theory provided shows that relative clauses are constructed with sideward movement into the relative clause are best equipped to account for the asymmetries we find in reconstruction into the relative clause between true complements and pseudo-complements. This theory, in combination with Sportiche's theory of Neglect provide a deeper understanding of the mechanisms that underlie relative clauses and also provides a principled account of the reconstruction facts that will be presented.

2 Minimalism and previous theories of parasitic gap constructions

2.1 Introduction

In order to set the stage for a discussion of the usefulness and the problems that arise from a sideward theory of movement, it is important to talk about the motivations and the underpinnings of such a theory. Sideward movement is, in my mind, one of the more interesting predictions made by theories created in a minimalist framework. In order to see that, we must understand what types of theories the Minimalist Program encourages, the predictions that follow from the few operations that are present, and how they lead to sometimes surprising conclusions. Many years of the study of syntactic structure in linguistics has shown that the complexity of the structures and patterns that we find are vast and can be challenging to nail down. From this point of view, it may seem optimistic to think that the rules, operations and constraints of previous waves of analysis that have brought to bear on the complexities of language can be easily replaced. However, it has been the track record of the Minimalist Program do so with just a few beginning assumptions and bring a deeper understanding of the phenomena in question.

This chapter will discuss the Minimalist Program's aims as they relate to our current understanding of Merge, Phases, movement, and ultimately sideward movement. Having laid the ground work for these basic principles and operations, I will look at one application of the sideward movement operation, parasitic gap constructions. I will examine previous theories from Nunes (1995, 2001, 2004, 2012) and Haida & Repp (2012). These theories, however

imperfect, show that a surprising amount of restriction is already present for a potentially powerful operation like sideward movement in the way the derivation proceeds.

2.2 The Minimalist Program

One of the key goals of the minimalist program is to understand what aspects of our theory of syntax are crucial, and what can be done away with. In considering this, Chomsky (2001) considers the optimal solution to the legibility constraints imposed upon the faculty of language by the other cognitive systems that it interfaces with, namely the articulatory/perceptual system, and the conceptual/intentional system. The claim that the faculty of language that humans possess is in fact this optimal solution is the Strong Minimalist Thesis. The Minimalist Program is therefore two-fold.

2.2.1 The optimal solution

Let us first consider the optimal solution to the language problem. What are the set of operations and constraints on that system such that all these operations and restrictions are necessary and irreducible? Chomsky (2001) states that any system from which larger units are created from independent atoms (lexical items) must have some process which creates a unit from smaller units. At the very least, such an operation must be able to take two objects, and make one new object out of them. From this irreducible necessity of the computational system, we can predict that a process like Merge, which takes two objects and creates a new object from them. Since restrictions placed on such an operation would push against the irreducible nature of this operation, Chomsky takes Merge to be simply the operation that takes two syntactic objects

(of which lexical items are included) and creates another syntactic object that is the set of both of the items merged.

It is also the case that the computational system must create objects that are legible to other cognitive systems, specifically the Conceptual-Intentional (CI) system to calculate meaning, and the Articulatory-Perceptual (AP) system, which externalizes language. These interfaces can provide restrictions on the possible outputs of the computational system. Since output representations must be legible to the interface between the computational system and these other systems, we can expect outputs that are able to be generated, but are rejected by the interfaces. Since we can motivate restrictions at the interface, it stands to reason that certain restrictions previously thought to hold on syntactic operations might instead hold on at the interfaces. For example, it is hard to state constraints like the Theta Criterion and the Case Filter as anything other than constraints on the output state of such an operation. However, checking these output states incrementally entails a new, syntax-internal operation that checks these states, as well as a level of representation that is incrementally checked.

This can be avoided if the Theta Criterion and the Case Filter both are a form of interface legibility requirement. If such restrictions can be stated as interface conditions, then the motivation for a syntax internal operation would be unfounded. One way in which this can be done is through the concept of uninterpretable features.¹ If certain features are illegible to the interfaces, then they must be eliminated before that evaluation takes place. The Case Filter can

¹Chomsky (2001) addresses the apparent imperfection of interpretable features through noting that in order for certain objects to make themselves visible in the structure for displacement, they must have features that reflect this. Uninterpretable features play both the role of marking a head as needing to be in a relation to another phrase, as well as marking those that will be related. If we take displacement to be one of the key facts of the language problem to be solved, then uninterpretable features are no longer an imperfection, but an efficient mechanism to facilitate such relationships and displacement generally.

be conceptualized in terms of uninterpretable features, as in Chomsky (2001). Uninterpretable case features, if left in the structure, would be illegible at the interfaces. These must be eliminated in the syntactic computation in order for the resulting structure to be legible at LF.

All of this is to say that we should not expect rules like the Theta Criterion or the Case Filter to be rules on the syntactic calculation, as these would be additional machinery. Instead, representational restrictions are the result of the requirements of the interfaces (or potentially the lexical items themselves). For example, (1) is an example of a sentence ruled out by the Case Filter.

(1) *It seems John to be angry.

If we take as a given that untensed T in the embedded clause cannot assign case, then *John* has not had any case assigned. Since, following the Strong Minimalist Thesis, there is no intermediate representation within the syntax to check that DPs have case assigned to them, the only place this can be checked is at the interfaces. But since no case has been assigned, uninterpretable features remain on *John* that need to be checked (in this case, *uCase*). Since this feature is uninterpretable to the CI interface, the derivation crashes.

Since representations at the interfaces are independently motivated, this leaves us with a more parsimonious state of affairs if the empirical facts can be accounted for with equal completeness. If other restrictions on the narrow syntax can be pushed out to the interfaces in a similar way, then the narrow syntactic computation is governed by only a few basic operations, and any restrictions come from the interface with other cognitive modules.

While limited operations and the interface with other cognitive systems are two ways in which the computational system can be restricted, we know that this system operates within a larger cognitive architecture. This architecture has its own constraints that it puts on the computational system, such as how memory is used and general principles of computational efficiency. One such implementation of this is the concept of phases. Phases were first introduced into the minimalist literature in by Chomsky (2000) as a way to account for the behavior of expletive subjects in certain sentences. For example, it is important to be able to derive (2a) while ruling out (2b).

(2) a. There are likely to be errors in this dissertation.

b. *There are likely errors to be in this dissertation.

(3) Num = (there₁, are₁, likely₁, to₀, be₀, errors₀, in₀, this₀, dissertation₀)

K = [_T to be errors in this dissertation]

In (3), where K represents a syntactic object in the derivational workspace and N represents the Numeration, we are at a state in the derivation where T needs to satisfy its EPP feature. In general, an EPP feature can be satisfied by either moving a DP already in the structure to the specifier, or an expletive can be merged into this position. In (3) there is a choice between merging *there* from the Numeration, or moving the DP *errors*. Merging *there* into that position will produce (2a), while moving *errors* up will produce (2b). One way to rule out (2b) is to say that Merge must be a preferred operation to Move. This Merge over Move (MoM) constraint is based in the assumption that Move is a more complex operation than Merge, and

should therefore be preferred. Moving *errors* to its position in (2b) is not allowed, since *there* could have merged into that position.

This approach needs additional help, however, since MoM, without any additional apparatus, will predict that (4a) is ungrammatical, and the structure should have instead produced something like (4b).

- (4) a. There is hope that someone is likely to read this dissertation.
b. *is hope that there is likely someone to read this dissertation.
- (5) Num = (there₁, hope₁, that₁, is₀, likely₀, someone₀, to₀, read₀, this₀, dissertation₀)
- K = [_T is likely someone to read this dissertation]

That is because at the derivational step in (5), MoM says that *there* must merge into this position. Therefore (4a) should be underivable. Since it is, there must be something else at play. We want to say that MoM still holds in (2). This can be done if *there* is only available for merger at a certain point in the derivation. Chomsky stated that if we assume that only smaller parts of the Numeration, what he called Lexical Arrays (LAs) are available in certain phases of the derivation, then we could still say that in (4) MoM forces *there* to merge since it is part of the LA available at that point in the derivation, but in (4b), *there* is part of a later LA associated with the higher clause.

These phases of the derivation begin as sections of the derivation over which particular LAs must be exhausted, but have been employed in a variety of uses, such as explaining the unavailability of syntactic objects in lower phases to operations in higher phases. This notion is

described in (6) as the Phase Impenetrability Condition (PIC) from Chomsky (2000). This is later refined in Chomsky (2001) to account for certain empirical facts, and stated in (7), which states that objects in the domain of H are only inaccessible to the next phase (ZP), but can be accessed by intermediate phrases.

- (6) Phase Impenetrability Condition (version 1) – In phase α with head H, the domain of H is not accessible to operations outside α , only H and its edge are accessible to such operations.
- (7) Phase Impenetrability Condition (version 2) – The domain of H is not accessible to operations at ZP [a higher phase]; only H and its edge are accessible to such operations.

In addition to the potential of explaining inaccessibility through the PIC, Richards (2011) notes the computational gains that can be made in understanding the syntax through the lens of phases.² If phases are units which can be incrementally spelled out (see Uriagereka's (1999) Multiple Spell-out), then phases can be considered a way of reducing the computational burden by culling the derivation by sending a portion of the structure to LF and PF. This reduces the amount of information the computational system must track, and also provides a motivation for the PIC in that syntactic material that has undergone Spell-out is also syntactically unavailable.

This first aspect of the Minimalist program brings a sort of optimization to our understanding of the faculty of language. Exactly what operations and restrictions can be motivated by the system alone, independent of the data that they hope to describe? This type of

² Richards (2011) discusses this in terms of both phases and Lexical (sub)arrays, though concludes that subarrays are preferable.

exploration gives us a set of well-motivated theories. So there is a sense in which the Minimalist Program is about possible computational systems, and not just the computational systems actually employed by the faculty of language. The hope is that we find crucial insight into the actual computational system employed through this exploration of its possible components.

2.2.2 Evaluating the optimal solution

The second aspect of the Minimalist Program to consider is seeing to what degree this optimal solution reflects the faculty of language as it actually is. While it is possible that the faculty of language is in fact the same as this optimal solution, there is no reason to assume this is so. For example, it has been argued (Chomsky 2001) that the operation Agree is not necessary to satisfy the interfaces. However, we find long distance dependencies ubiquitously across languages, which seem best accounted for by an operation such as Agree. There have been some attempts to find ways of incorporating agreement phenomena into the Merge operation, which might be ideal, but Agree potentially remains a bit of an imperfection in the faculty of language as it stands. It is important to recognize this, and if we find the inclusion of such operations necessary (as seems to be the case with Agree), then the same treatment of finding the minimum necessary machinery must be applied to these operations as well.

So this second aspect of the Minimalist Program is at least as crucial as the first. The first brings about a sort of optimization to our theory of language. From this exploration of theories, we now have a set of proposals which we can begin to compare to the language data we wish to explain. Understanding the degree to which the actual faculty of language matches the optimal solution to the challenge of language gives us keen insight into both the peculiarities of the system, and also questions for further research, such as understanding why the actual faculty of

language deviates from this optimal solution, or if there is in fact a better solution that would bring the true computational system into clearer perspective.

With these goals in mind, let us examine the operation of Merge, and see how a reductive definition of the operation makes predictions about the possible structures, specifically with regard to sideward movement.

2.2.3 Minimalist conceptions of Merge and sideward movement

If Merge and Agree are the only operations available in the narrow syntax, then movement must be understood in these terms as well. In Chomsky (1995), Merge is understood to Select two syntactic objects, either from the Numeration of the derivational workspace, and then create a set made up of those two objects. This can be thought of as External Merge.

We can see examples of this process in the derivations in (8-10).

(8) a. Initial state

$$\text{Num} = (\alpha_1, \beta_1)$$

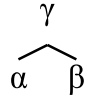
b. Select α and β from the Numeration

$$\text{Num} = (\alpha_0, \beta_0)$$

$$\alpha \quad \beta$$

c. Merge α and β

$$\text{Num} = (\alpha_0, \beta_0)$$



(9) a. Initial state

$$\text{Num} = (\alpha_0, \beta_0, \delta_1)$$

$$[\gamma \alpha, \beta]$$

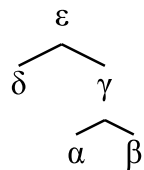
b. Select γ from the derivational workspace and δ from the Numeration

$$\text{Num} = (\alpha_0, \beta_0, \delta_0)$$

$$[\gamma \alpha, \beta] \quad \delta$$

c. Merge γ and δ

$$\text{Num} = (\alpha_0, \beta_0, \delta_0)$$



(10) a. Initial state

$$\text{Num} = (\alpha_0, \beta_0, \delta_0, \varepsilon_0)$$

$$[\gamma \alpha, \beta]$$

$$[\zeta \delta, \varepsilon]$$

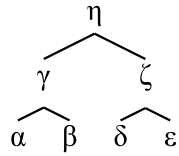
b. Select γ and ζ

$$\text{Num} = (\alpha_0, \beta_0, \delta_0, \varepsilon_0)$$

$$[\gamma \alpha, \beta] \quad [\zeta \delta, \varepsilon]$$

c. Merge γ and ζ

$$\text{Num} = (\alpha_0, \beta_0, \delta_0, \varepsilon_0)$$



In (8), we can see that two object (α and β) are selected from the Numeration (N). They are then merged to form a new, complex syntactic object (γ). In (9), we can see that there is a complex syntactic object (γ) in the derivation workspace, and a lexical item (δ) that must be merged from the Numeration. Both are selected, and merged to form another complex syntactic object (ε). Finally, in (10), we see two complex syntactic objects (γ and ζ) that are the target of select and are merged to form yet another syntactic object (η). We can see here that External Merge is equally happy to merge objects from the Numeration with each other and to complex objects already in the derivational workspace, as well as to merge two complex syntactic objects in the derivational workspace. Given these possible operations, Select is able to target any object, including the complete syntactic objects in the derivational workspace.

Movement is described as the result of Select targeting a syntactic object in the derivational workspace that already a member of a more complex syntactic object. Then a copy is made of a syntactic object, and that copy is then merged at the root of a structure. The lower, original copy is later phonologically deleted at PF. This can be thought of as Internal Merge. This eliminates the need for a specific Move operation and adds Copy, which can be justified

independently for use with agreement.³ In understanding movement as the concatenation of Select, Copy, and Merge, there is no need to understand Move as a separate operation within the computation system, but merely an extension of the idea of Merge.

We can see this sketched in (11) where in (11b) a target (α) is copied from inside of an existing syntactic object (β). Then the copy of α is merged with β to form a new syntactic object (γ).

(11) a. Select α from within β

[$\beta \dots \alpha \dots$]

b. Copy α

[$\beta \dots \alpha \dots$]

α

c. Merge the copy of α

[$\gamma \alpha [\beta \dots \alpha \dots]$]

In this way, movement is an extension of a minimal conception of Merge, since we are merely taking an object inside of a complex syntactic object and merging it (Internal Merge) instead of from the Numeration or one of the objects under construction in the derivational workspace (External merge).

Sideward movement seems to be a consequence of not stipulating this non-minimal limitation on the search space. Given the conception of Merge described above, an interesting prediction is made about what should be the possible inputs to the Merge operation. Select is not

³ This assumes a view of Agree in which some subset of the features of a syntactic object are copied and added to the features of another syntactic object.

limited to the Numeration or complete syntactic objects in the derivational workspace, as we can see it is able to target subsets of complex syntactic objects for Merge in (11). While this provides an encouraging boon to the explanatory power of Merge, there is no clear reason to expect that Merge should be limited to merging a complex syntactic object with its own constituents instead of the constituents of any object in the derivational workspace. Nunes (1995, 2001, 2004), Hornstein & Nunes (2002, 2008), and Hornstein (2001), and Boeckx et al. (2010) present sideward movement as an inevitable consequence of a minimalist conception of movement in which movement is simply a combination of the processes Select, Copy, and Merge.

In addition to Merge, it is a trivial and necessary assumption that the syntax must be able to have two complex syntactic structures with separate root nodes in the derivational workspace, such as when phrases are adjoined, or complex specifiers are merged to heads and their complements. When two (or more) structures are present in the derivational workspace, we might assume that only one is being acted on at a time, but the target of Select is unrestricted. Since we know that Select can target syntactic objects within a structure being extended as well as the Numeration, there is no clear motivation to restrict possible targets to only these two areas, but instead, it can be expected that any object in the derivational workspace is a potential target. This includes complex syntactic objects that are not currently being extended through Merge. It is therefore a consequence that Copy should be able to find a target in one structure and merge it to the root of another structure. It would be an extra stipulation on the process of Select and Copy to not allow the search of the other syntactic structure for a target.

These types of predictions are exactly the type we are looking for when exploring the minimalist paradigm. We can see that this begins to bear fruit immediately, as it might address

the particularly troublesome issue of head movement. An example from Boeckx et al. (2010) is provided in (12) showing an example of V-to-T head movement as sideward movement. The crucial steps are the copy action in (12b), which finds a target inside one syntactic object, and (12c), where we see that copied object merged to a different syntactic object than the one it was copied from.

(12) From Boeckx et al. (2010)

a. Applications of select, merge, and copy:

$$\text{VP} = [\dots \text{V} \dots]$$
$$\text{T}$$

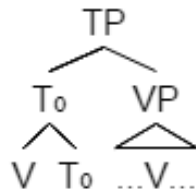
b. Copying of V:

$$\text{VP} = [\dots \text{V} \dots]$$
$$\text{T}$$
$$\text{V}$$

c. Merger of T and V (by adjunction):

$$\text{VP} = [\dots \text{V} \dots]$$
$$\text{K} = [\text{T}^\circ \text{V} [\text{T}^\circ \text{T}]]$$

d. Merger of K and VP:



Others (Nunes 1995, 2001, 2004; Hornstein & Nunes 2002, 2008; Hornstein 2001; and Boeckx et al. 2010) have argued that this means that sideward movement is an inevitable consequence of Merge as it is currently understood, since any restriction would be difficult to motivate. While I feel the claim that sideward movement is an absolutely necessary result of a minimal theory of movement is too strong, since there are clearly other ways to conceptualize what a minimal system can and cannot do, they do present a version of what is minimal that is quite reasonable while entailing sideward movement without stipulation. It is for this reason that I examine the potential (both for explanation on the one hand and for overgeneration on the other) of sideward movement, more so that its potentially fundamental position within the theory. This seems to be a fruitful path, since the consequences of sideward movement have already been invoked to help explain traditionally troublesome data such as parasitic gaps, head movement, adjunct control, and relative clause structure. It is primarily due to the potential of the further development of sideward movement theories that I pursue the consequences of including sideward movement into our syntactic theory, not its ‘clear’ inevitability.⁴

⁴ For more complete defenses of the necessity of sideward movement in minimal syntactic theories, see Boeckx et al. (2010), Nunes (1995, 2001, 2004), Hornstein & Nunes (2002, 2008), and Hornstein (2001).

2.3 Parasitic gaps as a case study

In order to better understand the consequences of sideward movement on the syntactic system, I will examine one phenomenon where sideward movement has been utilized for analysis, parasitic gaps. I will first present a summary of the data surrounding parasitic gaps, including the restrictions on extraction from parasitic gaps, the effects we find within the parasitic domain, and the relationship between parasitic gaps and islands. Second, I will discuss Nunes' (2004, 2012) account of parasitic gaps, in which he uses sideward movement to account for the relationship between the true gap and the parasitic gap. I will discuss some of the advantages of such an approach, as well as examine some of the problematic assumptions made by Nunes with regard to how adjunct islands are accounted for and assumptions made about the lack of late adjunction. Then I will discuss some refinements to Nunes' theory proposed by Haida and Repp (2012) in order to more accurately account for island effects found within the parasitic domain. Once again, I will discuss the advantages of their proposal, but also show that further refinement is needed, specifically with respect to the course of the derivation and how to properly restrict sideward movement through how the derivation proceeds from one phase to another.

2.3.1 Adjunct islands and parasitic gaps

Engdahl (1983) describes parasitic gap constructions as constructions in which there are two gaps controlled by a single filler, and one of the gaps depends on the other for grammaticality. For example, in (13) we see that both the theme of *read* and the theme of *recommended* are controlled by the filler *which book*. However, the parasitic gap (marked *pg* in this dissertation) is dependent on the presence of the main gap. We can see this in (14), where

filling in the parasitic gap in (14a) results in a grammatical sentence, but filling in the main gap while leaving the parasitic gap in (14b) is ungrammatical. We can see that the movement (14b) is across an adjunct island. We can see an example of parasitic gaps obviating subject islands by contrasting the sentence in (15) with the sentence in (16b).

- (13) [Which book]_i did Bob [_{VP} read *t_i*] after Mary recommended *pg_i*?
- (14) a. [Which book]_i did Bob [_{VP} read *t_i*] after Mary recommended the movie?
b. *[Which book]_i did Bob [_{VP} read the review] after Mary recommended *pg_i*?
- (15) Who_i did John's talking to *pg_i* bother *t_i* most?
- (16) a. *Who_i did John's talking to *pg_i* bother Suzy most?
b. Who_i did John's talking to Suzy bother *t_i* most?

One of the key discussions in the literature is the nature of the relationship between the filler, the true gap, and the parasitic gap. While it is clear that the meaning of the filler controls the meaning of both the true gap and the parasitic gap, what will be called into question is how that relationship is captured. Is there a true movement out of the parasitic domain, or is the relationship between the parasitic gap and the filler obtained some other way such as operator binding or null pronouns?

Nunes (2004) argues against many previous approaches to the nature of the syntactic object in parasitic gaps. The representations he argues against specifically are enumerated in (17) (from Nunes (2004)).

- (17) a. Traces of across-the-board (ATB) movement (William 1978, 1990)

- b. Traces of WH phrases that are not created by movement (Frampton 1990)
- c. Null resumptive pronouns (Cinque 1990)
- d. Traces resulting from the movement of null operators (Chomsky 1986)

Nunes (2004) states that additional operations such as across-the-board movement as proposed by Williams (1990) should be avoided under a minimalist framework. Additionally, he points out that theories that depend on traces that are derived in ways other than movement are problematic, since they violate the principles of inclusiveness. Chomsky (1995) suggests that if the SMT holds, a ‘perfect language’ should build structures out of objects that are already given from the lexicon, and none can be added in the course of the derivation. Arguments for the copy theory of movement state that traces always violate the condition on inclusiveness, and that the Copy Theory of Movement overcomes these problems by keeping copies of the moved object instead of inserting a trace in to the structure (which is also a violation of the Extension Condition). As for (17c-d), Nunes argues that these options deviate from what should be the null hypothesis, which is that the inventory of empty categories and operations should only be expanded when absolutely necessary. If possible, null operators or pronouns should conceptually be replaced by movement, chain formation (the process by which copies are related as chains) and then chain reduction (the deletion of certain copies).

In order to adequately replace null operators or null resumptive pronouns, it must be shown that a sideward movement account of parasitic gaps can account for a variety of restrictions and facts about parasitic gaps. Nunes specifically looks at the set of restrictions that are enumerated in (18).

- (18) a. Parasitic gaps are licensed at S-Structure
(see Chomsky (1982), for instance).
- b. Parasitic gaps cannot be c-commanded by the “real” gap
(see Taraldsen (1981) and Engdahl (1983), for instance).
- c. Parasitic gaps cannot be licensed by A-chains
(see Chomsky (1982), for instance).
- d. Parasitic gaps are “selectively sensitive to islands”; that is, although they typically occur within (Condition on Extraction Domain) islands, they cannot be separated from their licenser by more than one island
(see Kayne (1983, 1984), Contreras (1984), and Chomsky (1986), for instance).
- e. Parasitic gaps can only be NPs
(see Aoun and Clark (1985) and Chomsky (1986) for instance).
- f. Parasitic gaps cannot be licensed by nonreferential NPs
(see Cinque (1990)).

Let us examine each of these claims to better understand the restrictions that any theory of parasitic gaps must be able to account for to be considered. Engdahl (1983) provides the examples in (19) to show that WH phrases in situ cannot license parasitic gaps (with indices and bracketing added). While these WH phrases certainly must escape their original positions, that later position does not license the parasitic gap. This provides evidence for the claim in (18a)

(19) a. *John filed [which articles]_i without reading *pg_i*?

b. *I forget who filed [which articles]_i without reading *pg_i*?

In addition, it is not the case that any moved controller can license a parasitic gap. (18c) states that A-chains cannot license parasitic gaps, which we can see in (20) (from Engdahl (1983)).

(20) a. John_i was killed *t_i* by a tree falling on *pg_i*.

b. Mary_i seemed *t_i* to disapprove of John's talking to *pg_i*.

This data leads to a larger fact about parasitic gaps, which is that parasitic gaps cannot be c-commanded by their true gaps (as in (18b)). This is shown in data like (21) from Engdahl (1983), and in (22) from Nunes (2004).

(21) a. *[Which articles]_i *t_i* got filed by John without him reading *pg_i*

b. *Who_i *t_i* sent a picture of *pg_i*?

c. *Who_i *t_i* remembered talking to *pg_i*?

d. *Who_i *t_i* remembered that John talked to *pg_i*?

e. *[Which articles]_i did you say *t_i* got filed by John without him reading *pg_i*?

(22) *I wonder [which man]_i *t_i* called you before you met *pg_i*?

These examples show that the foot of the A'-movement chain of the real gap must not be in a position that c-commands the parasitic gap (even if its A-movement chain begins in a position that does not c-command the parasitic gap).

One of the key arguments for movement of within the parasitic domain (be it of operators or copied materials) is the fact that there are island effects that can occur in parasitic gap constructions (Kayne (1983, 1984), Contreras (1984), Chomsky (1986)). We can see in (23) that parasitic gap constructions do not obviate all island effects.

(23) a. Adjunct island

*[Which book]_i did Bob [_{VP} read *t_i*] after Mary [[_{VP} left the room] without recommending *pg_i*]?

b. WH island

*[Which book]_i did Bob [_{VP} read *t_i*] after [_{CP} Mary asked [_{CP} when Suzy recommended *pg_i*]])?

c. Subject island

*[Which author]_i did Bob [_{VP} criticize *t_i*] after [_{CP} [_{DP} pictures of *pg_i*] upset the fans])?

In order to account for the movement islands we see in (23), Chomsky (1986) and Frampton (1990) have posited movement within the parasitic domain. Since movement islands imply movement, these theories state the object in the parasitic domain is compelled to move from where it is first merged to a higher position, frequently to some high functional phrase such as the specifier of a CP high in the adjunct. Movement within the parasitic domain therefore resembles the structure in (24).

(24) [Which book]_i did Bob [_{VP} read *t_i*] after [_{CP} Op_{pg} Mary recommended *pg_i*]?

If the object in the parasitic domain must make it to this position to check some crucial feature, then movement islands such as those in (23) will become a source of ungrammaticality since they block such a movement.

While this theory is certainly can account for the facts in (23), we find ourselves in the position of needing two mechanisms for relating a gap to a phrase; movement (which we need on independent grounds) and null operators or pronouns that are then related to the filler. We know displacement in some form is necessary within our syntactic theory. If it is possible, reducing theories that invoke unpronounced operators to some subclass of movement would be a preferable state of affairs. Given a copy theory of movement, this provides a straight forward relationship between the meaning of the filler and the parasitic gap. Other methods that might do something similar all demand additional machinery. If the parasitic copy is an actual copy of the filler without the phonetic content, then there must be some mechanism in the lexicon, Numeration, or syntax that can take an object and strip its phonological features. While syntactic objects not being pronounced occur quite frequently in language, most theories assume this is a PF phenomenon of some sort. Another possibility is that there is some relating operation that give the parasitic object its meaning. This means there was some null object in the Numeration that was merged and its meaning is not composed of its parts, but by some relation. This relation is again additional machinery when compared to what is necessary if the project of sideward movement can adequately substitute. Lastly, if it is a true copy of the filler, but is later related to the filler's chain, but not through movement, as Frampton (1990) suggests, then there are two mechanisms that relate chains, movement and this additional relating operation. In addition to the relating operation, there is also a copy operation that is not motivated by syntactic requirements, but occurs extra-syntactically. If Frampton's analysis is that a copy of the filler

itself is made, then this in fact becomes a sideward movement theory of parasitic gaps, and since select-copy-merge are the components of movement, there is no need for the additional relating operation.

Whether we should endeavor to have all gaps that have been analyzed with null operators be related to a phrase through movement is a question that is beyond the scope of this dissertation, but if it can be done, as this would reduce the mechanisms that the syntax has access to. What does seem to be the case is that parasitic gap constructions are well suited to the type of sideward movement predicted by the minimalist conception of Merge above.

Lastly, we can see in (23) (from Postal (1993) and Cinque (1990) via Nunes (2004)) that the category of the parasitic gap is restricted to NPs, and that they cannot be licensed by nonreferential NPs. This data has been used to argue for particular representations of the parasitic gap, such as Cinque's (1990) proposal that the gap is a null resumptive pronoun.

- (23) a. *How did Deborah cook the pork t_i after cooking the chicken pg_i ?
- b. *How sick did John look t_i without actually feeling pg_i ?
- c. *This is a topic about which you should think t_i before talking pg_i .
- d. *How many weeks did he spend t_i in Berlin without wanting to spend pg_i in London?

In conclusion, any sideward movement theory of parasitic gaps should be able to provide a derivation for parasitic gaps, as well as a reason that the derivation of an adjunct island like (13b) is not allowed. In addition, that theory must be able to account for a wide variety of

restrictions on parasitic gap constructions, summarized in (17). If this can be done successfully, then the sideward movement theory of parasitic gaps has successfully leveraged an operation that is predicted by a minimal theory of Merge and the Copy Theory of Movement, while avoiding enriching the catalog of empty categories or the available operations in the syntax, whether they be special movements or versions of chain composition that are non-general. While it can be argued that these achievements are not a complete replacement for the depth of work that has gone into the theories accounting for parasitic gaps in the past, at the very least it provides a model of how these accounts can be adapted into a minimalist framework. In the next section, I will describe Nunes' (2004, 2012) version of such a sideward movement theory of parasitic gaps, which addresses all of these points. I will also examine some of the areas in which this theory falls short.

2.3.2 Nunes on Parasitic Gaps

Nunes (2004, 2012) proposes a sideward movement account of parasitic gap constructions. Instead of relating the semantic content of the parasitic gap to its filler through chain composition, he relates them through movement. By assuming that the WH phrase begins in the parasitic domain, and then moves to the position of the true gap, these two gaps are related by both being a true copy of the moved phrase. He provides a derivation for the movement from the unattached vP adjunct in (24) to the matrix-object position, and then movement in the typical fashion of WH phrases to SpecCP. We see in (24a) that the adjunct clause has been built up with *which paper* inside, and the main clause verb is introduced into the derivation. In (24b) *which paper* is copied from the adjunct clause, and then merged to the main clause verb in (24c). From here, the main clause builds up in (24d-g), merging the adjunct clause in (24e). In (24h), Chain

Reduction eliminates copies that are c-commanded by the pronounced copy, and we see the final product.

(24) [which paper]_i did you file t_i after John read *pg*_i?

a. *file* introduced into the derivation

K = [John read [which paper]_i]

L = file

b. Copy made of *which paper*

K = [John read [which paper]_i]

L = file

M = [which paper]_i

c. *which paper* merged with *file*

K = [John read [which paper]_i]

P = [file [which paper]_i]

d. Matrix vP constructed

K = [John read [which paper]_i]

Q = [you [file [which paper]_i]]

e. Parasitic domain merged with matrix vP

[vP [vP you [file [which paper]_i]] [after John read [which paper]_i]]

f. Matrix clause built up to the introduction of the matrix C

[did [TP you T [vP [vP you file [which paper]_i] [PP after John read [which paper]_i]]]]

g. WH movement of *which paper*

[[which paper]_i did [TP you T [vP [vP you file [which paper]_i] [PP after John read [which paper]_i]]]]

h. Deletion of lower copies

[[which paper]_i did [TP you T [vP [vP ~~you~~ file ~~[which paper]_i]] [PP after John read ~~[which paper]_i]]]]~~~~

While a sideward movement analysis like Nunes', if correct, does provide a mechanism for linking the true gap and the parasitic gap, this class of movement could be problematic if it were unrestricted. One crucial restriction must be made that would disallow the derivation of sentences with adjunct islands, as in (25). If sideward movement allows for movement out of islands such as the adjunct island in (24), why wouldn't a similar movement be allowed in (25)?

(25) *Which paper did you file the report after John read?

In order to prevent this possibility, Nunes points out two differences between the derivation of (24) and the potential derivation of (25). He assumes that adjuncts are in fact islands, but an object's 'adjuncthood' is a result of its position in the structure. In the derivation of (25), the vP 'adjunct' *after John read* is not attached to the vP when *which paper* moves out,

and so *after John read* is not yet an adjunct. This means that no island violation has occurred, since there is no (adjunct) island.

Second, he points out that extraction of the WH-phrase will not work for (24), since there is no position within the vP that would motivate the movement before the adjunct is attached. At the time when *which paper* is extracted for WH movement, the *after* phrase must be attached to the vP. Therefore, it is an adjunct, and thus the adjunct island is violated.

This poses a few problems. The first is that this view of adjunct islands suggests that the rule against extraction is configurational. While it is true that we can describe the adjunct island configurationally, there is a problem with understanding that this is how the actual restriction against adjunct islands is actually implemented. Under a minimalist theory of the derivation, there are only a few mechanisms that are interacting: the operations of the narrow syntax, the properties of lexical items involved, and the legibility conditions of the interfaces.

Ungrammaticality must come from the inability to produce a particular structure, either through the operations being unable to produce such a configuration, or a rejection of the representation produced by the interfaces. There is no reason to think that the operations of the narrow syntax have access to the configuration of a syntactic structure within the computational system. Merge can presumably search anywhere in the derivational workspace. While certain parts of a structure can be made unavailable through Spell-out to the interfaces, the search space seems unrestricted within the derivational workspace, and to add such restrictions seems against the minimalist intuitions that sideward movement is based on. It is unclear how this configurational constraint would be incorporated into the lexical items involved, and the legibility conditions at the interface are ill-equipped to handle this, since such arbitrary limits on configuration would be highly stipulative and hard to motivate. It is important to remember that inaccessibility through

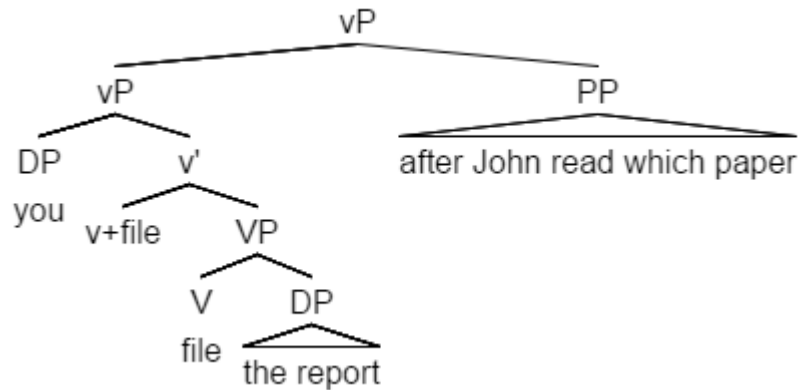
Spell-out does remain an option, as this is how Nunes attempts solves the problem, and must ultimately be where the answer lies.

Second, the adjunct is only attached at this point in the derivation if we assume that the vP adjunct is merged cyclically in the derivation. Theories of late adjunction (Lebeaux 1988, Fox 2002, Stepanov 2007, and Henderson 2007, among others), in their strongest terms, can merge any adjunct into the structure counter-cyclically. If the vP adjunct were able to delay adjunction until after the movement, then the derivation of (25) should be possible. Since (25) is ungrammatical, Nunes argues that the vP adjunct must merge cyclically. Reconciling the claim that there is no late adjunction with the broader arguments for counter-cyclic merger is tricky. This will be more thoroughly addressed in Chapter 5.

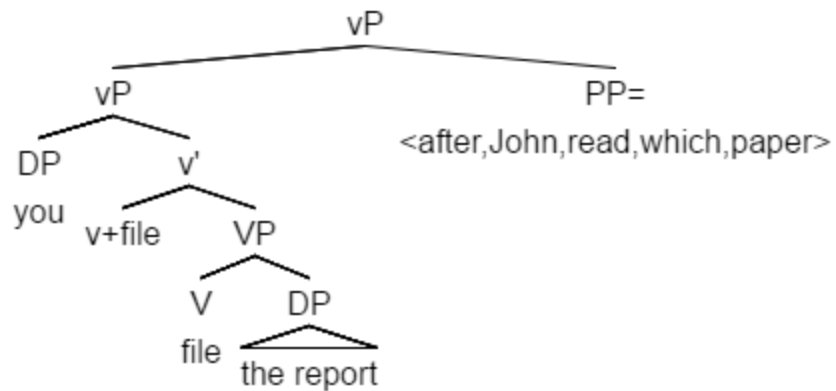
More immediately, Nunes' theory of adjunct islands introduces a problem for the timing of unavailability of adjuncts through Spell-out. While Nunes (2012) is more agnostic on the underlying source of the island-hood of adjuncts, Nunes (2004) explicitly invokes the analysis in Nunes and Uriagereka (2000) where Spell-out leads to the unavailability of certain syntactic objects. This account is derived from Uriagereka's (1999) proposal that the unavailability of objects in the derivation is the result of multiple Spell-out. As portions of the syntactic structure are spelled out, they become unavailable for further syntactic computation. Nunes and Uriagereka (2000) use this idea to account for some types of movement islands (adjunct and subject islands) as Conditions on Extraction Domains (CED), as described by Huang (1982). Specifically, they argue that, in order to linearize a structure like (26a) under the Linear Correspondence Axiom (LCA, Kayne 1994), the adjunct must be spelled out, and the adjunct's internal linear orders are ensured through that process (26b). However, a label is left behind and inserted into the structure, which is linearized with respect to the rest of the structure. Due to this

Spell-out, the members of that adjunct are unavailable for further syntactic operations. This successfully accounts for the unavailability of movement out of adjuncts.

(26) a.



b.



However, if we want to understand adjunct islands in terms of Spell-out and the resulting unavailability of syntactic objects that are spelled out, what are the targets and triggers of this Spell-out? Firstly, the trigger for the version of Spell-out that Nunes is using here is a problem. It demands lookahead to know that something will be problematic to the LCA. How does the

derivation know that something will be problematic? How is this checked, and by what operation? These questions are difficult to answer in terms of operations inside of the derivational workspace, since linearization is a process only PF has access to. This means that in order to know that linearization will be problematic for a particular syntactic object, the computational system would need to look ahead to see what PF might reject. In addition, this is only true once the adjunct has actually adjoined. There is no sense in which a structure that has not yet been adjoined is problematic for linearization. While interface conditions can reject certain structures, this should not be confused with their ability to affect the derivation. If two operations are possible, but one will be rejected, then interface conditions can explain why we only see the result of one of those operations. But if we cannot motivate an operation like Spell-out independent of the interface conditions, then it should never occur. Therefore, no operation within the computational system should react to a syntactic object's illegibility at the interface. Since PF conditions that the LCA imposes are not accessible to the computational system, then what that triggers the Spell-out that Nunes is proposing is mysterious. It cannot be the case that the trigger can be stated in terms of the PF conditions that will later reject that structure, since that would require looking ahead in the computation.

Next, Nunes' theory must be able to account for the restrictions found in (18), repeated here as (27).

- (27)
- a. Parasitic gaps are licensed at S-Structure.
 - b. Parasitic gaps cannot be c-commanded by the "real" gap.
 - c. Parasitic gaps cannot be licensed by A-chains.

- d. Parasitic gaps are “selectively sensitive to island; that is, although they typically occur within (Condition on Extraction Domain) islands, they cannot be separated from their licenser by more than one island.
- e. Parasitic gaps can only be NPs.
- f. Parasitic gaps cannot be licensed by nonreferential NPs.

For many of these, Nunes (2004) relies on the specifics of how chains are created and then reduced at PF so that only some of the elements of a chain are deleted. In many cases, this depends on whether the appropriate chains are formed. Form Chain is an operation that occurs at PF in order to later enable Reduce Chain to delete portions of that chain. While Move as an operation would enrich the narrow syntax, it is the case that PF requires a way to linearize the syntactic object sent to it from the narrow syntax. The deletion of uninterpretable features left in earlier copies is one motivation for the deletion of some copies of a syntactic object. However, according to Nunes (2004), any two non-distinct copies of the same syntactic object will produce an issue for linearization. Since they are non-distinct, PF interprets them as the same object. The output of linearization is an ordering of the syntactic objects that are given from the syntactic computation. If there are two non-distinct copies of an object that must be linearized, then it is possible for such an object, let's call it X, to be linearized both before and after another object Y. This means that the resulting representation would state that X must precede Y, and Y must precede X. This is a paradox for PF, and results in a crash.

In order to prevent this type of ordering paradox, copies of a particular syntactic object are made into a representational chain through the operation Form Chain, and then portions of the chain are deleted by Reduce Chain in order to allow linearization. Nunes (2004) proposes

several constraints on Form Chain, listed in (28), that reflect previous restrictions on Move, such as the c-command requirement (28b), Last Resort (28c), and the Minimal Link Condition (28d).

Nunes also gives definitions of sublabel and closeness, which I list here as (29) and (30).

(28) Conditions on Form Chain

Two constituents α and β can form a nontrivial chain $CH = (\alpha, \beta)$ if

- a. α is nondistinct from β ;
- b. α c-commands β ;
- c. there is at least one feature F of α such that F enters into a checking relation with a sublabel of the head of the projection with which α merges and for any such feature of α , the corresponding feature F of β is accessible to the computational system and;
- d. there is no constituent γ such that γ has a feature F' that is of the same type as the feature F of α , and γ is closer to α than β is.

(29) Sublabel

Σ is a sublabel of the head H iff σ is a feature of H or a feature of some element adjoined to H .

(30) Closeness

γ is closer to α than β is iff (a) α c-commands γ and γ c-commands β , and (b) γ is not in the same minimal domain as α or β .

The condition in (28a) ensures that chains are only made between two objects that are non-distinct copies of each other. The typical c-command requirement on movement is reflected in (28b), though here we can see that it is removed from the movement operation, and merely restricts the representations that can be formed. Nunes states that (28c) reflects Chomsky's (1995) Last Resort condition on movement, where there must be a featural motivation for movement. For example, movement might occur to check the WH feature of a syntactic object. This condition ensures that any object that forms a chain with a lower object is in a position where a feature is checked. It is notable that the only requirements this condition has on β is that it have an unchecked version of the feature that is checked by α . That is what makes it visible to the computational system. Lastly, (28d) is a reflection of Chomsky's Minimal Link Condition, which ensures that there are no intervening syntactic objects (using (30) to define what qualifies as an intervening syntactic object) between the α and β that has the features described in (28c). Only the closest object can with those features can form a chain with α .

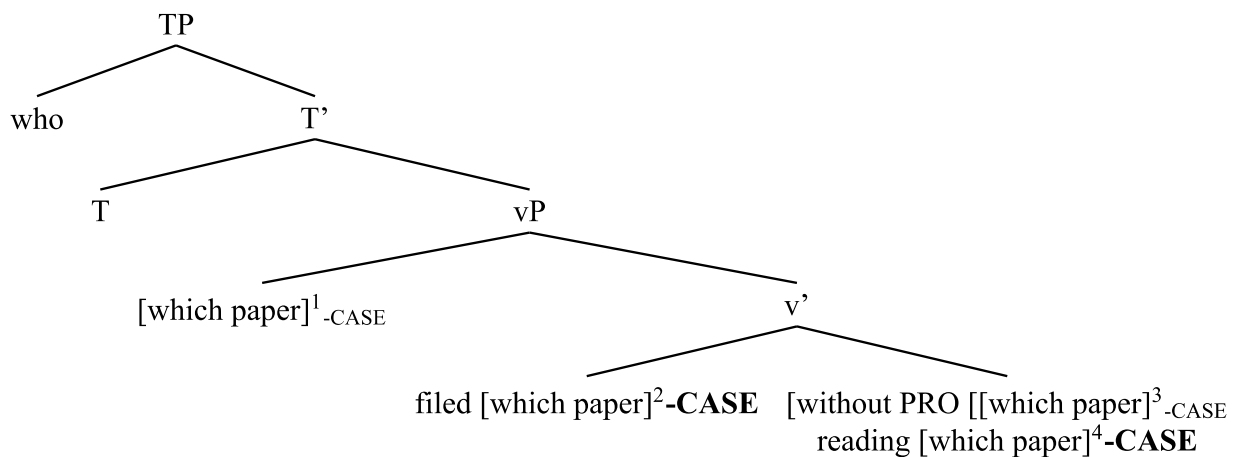
In Nunes (2004), the S-structure licensing condition is addressed in one of two ways. Given a sentence like (18a), repeated here as (31), one possibility is that there is movement that takes place between when (31) is sent to PF and its final LF representation. In that case, there is no c-command relationship between the two copies of *which articles* and no chain is able to be formed (following (28a)). This produces a violation of anti-symmetry in the linearization, and thus the derivation crashes.

(31) *John filed [which articles]_i without reading [which articles]_i?

Another possibility is that there is a movement that occurs that causes a copy of *which articles* to move to a position in which it c-commands all of the copies. Nunes provides the

structure in (32) (from Nunes (2004), p. 108, example (53)) for such a possibility. The bold case features are those that have not been checked yet, and the subscripted case features have been checked, and are therefore unavailable to the computational system. *Which articles* has moved to a higher position in both the matrix clause and the parasitic domain in order to check case. Nunes states that while copy¹ and copy² can form a chain and copy³ and copy⁴ can form a chain, no larger chain can be made with all 4 copies. This is because copy¹ cannot form a chain with copy³ according to (28c) due to the fact that the case feature in copy³ is unavailable after being checked.

(32)



Something interesting to note about this approach is that all of this assumes that there is no WH movement to a high position within the parasitic domain (such as the CP). In (32), we are seeing a movement to case checking position, such as AgrO. Additionally, the copy of *which paper* that copy² is made from must be copy⁴, since copy³ has already checked its case. Otherwise there would be no motivation for the creation of copy¹, since the case of copy³ would already be checked. This is interesting because movement to a higher position within the parasitic domain

is frequently used to account for island effects within the parasitic domain, a problem we will return to later.

Next, the restriction against the true gap c-commanding the parasitic gap (27b) and the restriction against A-chains licensing parasitic gaps (27c) can both be derived again in terms of a failure to form a chain, in this case due to intervention effects (28c). Nunes provides (33) as an example where *which man* is the subject of the clause that the parasitic domain is adjoined to.

(33) *I wonder which man called you before you met.

(34) *[I wonder [_{CP} [which man]¹_i [_{TP} [which man]²_i [_{VP} [_{VP} [which man]³_i called you] [_{PP} before you met [which man]⁴_i]]]]]]]

In (34), we can see that this subject position c-commands the copies in the parasitic domain. In this example, a chain is made from copy¹, copy², and copy³, since the feature that copy¹ has checked (WH) is legible on copy², and the feature copy² has checked (case) is visible on copy³. The trouble is in creating a chain with copy⁴. Copy¹ cannot make a chain with copy⁴ because of the intervention of copy². Condition (28d) states that since Copy¹ checked its WH feature in its position, then any objects it forms a chain with must not have an intervener with a WH feature. Copy² does, and so the chain between copy¹ and copy⁴ is not formed. Copy² cannot form a chain with copy⁴ because the feature being checked by copy² is case. This means there can be no case interveners. However, *you* does bear a case feature, and therefore the chain cannot be made between copy² and copy⁴. Lastly, copy³ cannot form a chain with copy⁴ since there is no c-command relationship between the two. Since Chain Reduction can only delete copies in the one chain (copy² and copy³ from the chain CH=(copy¹, copy², copy³)), copy⁴

remains in the derivation, which produces an anti-symmetry violation in linearization, resulting in a crash.

A similar story can be told for the restriction against A chains licensing parasitic gaps (27c), since these movements are driven by case checking of the subject, as in Chomsky and Lasnik's (1993) example in (35). Nunes gives the structure in (36).

(35) * The book was filed without my reading first.

(36) [TP [The book]¹_i was [VP [VP filed [the book]²_i] [PP without my reading [the book]³_i first.]]]

The chain of copy¹ and copy² can be made on the basis of the case feature checked by copy¹. However, no such chain can be made between copy¹ and copy³, since there is a case intervener (*my*) that prevents the chain from being made. Again, both structures ((34) and (36)) assume that there is no movement to a specifier in the CP of the adjunct clause. This becomes crucial in these examples, since such a movement would eliminate the problem of a case intervener in (33) and (35), and Nunes would not be able to account for their ungrammaticality in terms of intervention. This will prove to be a problem in dealing with island data.

One of the main focuses of Nunes (2012) was addressing the derivation of CED effects in parasitic domains (the restriction in (27d)). In (37) we see that if sideward movement is unbounded, and this derivation would allow for sentences in which there are no CED effects inside of the adjunct. The ungrammaticality of (37) (as well as 22a-c)) shows us that there are islands that must be accounted for within the parasitic domain.

(37) *[which paper]_i did you file t_i after John left the room without reading pg_i?

a. *file* introduced into the derivation

K = [John [vP [vP left the room] [PP without reading [which paper]]]]

L = file

b. Sideward movement of *which paper*

K = [John [vP [vP left the room] [PP without reading [which paper]_i]]]

M = [file [which paper]_i]

Nunes states that there is no such problem with sideward movement such as in the parasitic gap derivation in (24), as the movement in (24a-c) is not out of an adjunct, since adjunct is a relational notion, and thus it has not obtained that relationship with any other phrase at that point in the derivation. On the other hand, he states that since the derivation takes place in a “bottom-up and phase-by-phase fashion, as currently assumed (see e.g. Chomsky 2000, 2001, 2004)” (Nunes 2012, p. 123) the movement in (37) must take place when *which paper* is inside of an adjunct (not at some earlier stage when movement out of *without reading which paper* would be licit). Since the derivation is moving from more embedded phases to less embedded phases, the build-up of the adjunct *John left the room without reading which paper* must have already attached *without reading which paper* (since vP adjuncts must be merged cyclically), and so it is an adjunct, and therefore an island to movement. This predicts that the restriction on adjunct movement must not apply at the level of representation present at (24a-b), but does in fact hold in (37), since it is crucial that sideward movement is still subject to the same restrictions as upward movement (as it is composed of the same processes).

Nunes also points out that you cannot copy the moved phrase early in the derivation (while the phrase containing it is not an adjunct) and not merge it immediately, since movement generally (including sideward movement) is subject to Last Resort, which is to say that the movement must be motivated by some requirement within the syntax, such as the need to check an EPP feature or some other selectional feature (Chomsky 1995). If the Copy and Merge operations were not motivated by Last Resort, then the whole operation should be illicit. Essentially, if there is no motivation for the movement, then the copying should not take place, and if there is motivation for the movement, then the copy would need to merge into the position motivating the copy (already in the derivation). This cannot be true until much later in the derivation (when the main clause is being built).

In Nunes & Uriagereka (2000), Hornstein & Nunes (2002), and Nunes (2004) the ungrammaticality of adjunct island violations is the result of limiting the derivation to build only one phase at a time. Each phase has its own subarray created from the Numeration, which must be exhausted before moving on to building additional structure. Once a phase is built, then it can be subject to the PIC, which prevents movement out of the phase. The way in which adjunct islands are accounted for in these theories is by saying that the preposition in an adjunct is a member of the matrix subarray. Since these prepositions select the rest of the adjunct, once you begin to build the matrix subarray, it requires that the preposition be merged into it as well, which means that its sister must be available for merger and already created. Since *without reading which paper* in (37) must be spelled out as an adjunct (à la Nunes and Uriagereka (2000)), the movement from the adjunct island inside of the parasitic domain to the matrix clause violates the PIC. This does not apply to the derivation in (24), since the movement of *which*

paper is available to move since its phase has not spelled out, and the entirety of the parasitic domain is ready to be merged when the matrix subarray reaches *after*.

It also cannot be the case that the movement that derives (25) be from the matrix clause to the embedded clause as in (38).

(38) a. Introduction of *reading* into the derivation

K = [file [which paper]]

L = reading

b. Sideward movement of *which paper*

K = [file [which paper]_i]

M = [reading [which paper]_i]

Nunes claims this also is ruled out by the fact that the derivation must proceed from more embedded domains to less embedded domains. In (38) we see the matrix clause being built up first and *which paper* then being copied for movement by the building of the embedded clause.

This analysis seems to account for the structures in islands that can be accounted for with CED constraints. There is still the problem of WH islands within parasitic domains, as in (22b), repeated here as (39).

(39) *[Which book]_i did Bob [_{VP} read *t_i*] after [_{CP} Mary asked [_{CP} when Suzy recommended *pg_i*]]?

Spell-out due to LCA considerations should not affect the embedded clause in the parasitic domain. One might say that it spelled out due to regular cyclic Spell-out of phases, but this is a problem for Nunes. In his parasitic island derivation in (24), the object in the parasitic domain undergoes sideward movement. It is crucial to his account of the anti-c-command restriction on parasitic gaps and his account of A-chains not licensing parasitic gaps that there be no movement inside of the parasitic domain for the parasitic gap. This leaves that object in the Spell-out domain of the vP. In turn, this phase should be spelled out by the CP in the parasitic domain, making that object inaccessible. Therefore, Nunes cannot rely on cyclic Spell-out of embedded clauses for the unavailability of the embedded CP in the parasitic domain in (39). While WH islands can often be accounted for by using the Minimal Link Condition (Chomsky 1995), the conceptual motivation for cyclic movement is lost without the cyclic application of Spell-out. Cyclic movement is driven by the need to escape phases before they are targeted for Spell-out. This is so the object that is moved can be acted on later in the computation. So Nunes cannot appeal to inaccessibility due to Spell-out, because that would create islands for simple long distance parasitic gap constructions, and he cannot allow for phase by phase Spell-out due to the need to leave the object low in the parasitic domain to account for intervention effects from the subject in (33) and (35). This leaves the ungrammaticality of (39) mysterious under Nunes' account.

It seems clear that the island effect in (39) should be accounted for in the same way as other WH islands, which is to say that there must be some movement in the parasitic domain, which produces a violation of the Minimal Link Condition in (39). This means that another account of the anti-c-command and A-chain restrictions in (27b-c) is necessary.

While Nunes (2012) has laid the groundwork for a sideward movement theory of parasitic gaps, there are still some issues. As mentioned above, assuming that adjuncts merge cyclically, given the large amount of evidence for counter-cyclic merger, may be problematic. Also, there are the several problems regarding Nunes' account of adjunct islands. His explanation of adjunct islands seems to be based on a restriction on PF. However, operations in the syntax cannot act upon the knowledge of an upcoming violation due to restrictions on lookahead in the derivation. Also, the lack of movement within the parasitic domain in Nunes' theory of islands within the parasitic domain is problematic when accounting for WH islands. In order to introduce movement, however, new accounts of the restrictions against the true gap commanding the parasitic gap, and against the licensing of parasitic gaps using A-chains are necessary. Next I will examine some challenges to Nunes' theory made by Haida and Repp (2012) and their account for parasitic gaps when Nunes' assumptions fails to rule out certain derivations where the relative embeddedness is not as clearly defined as in (37).

2.3.3 Haida and Repp on parasitic gaps

Haida and Repp (2012) introduce some refinements to Nunes' restrictions on sideward movement. While Nunes (2012) claims that as long as the derivation is built up from more embedded to less embedded environments, one subarray at a time, then island violations (and the lack of violation in parasitic gap constructions) should be accounted for. Haida and Repp note that (40) is not so easily accounted for by what they refer to as OneSub, Nunes' restriction to completing one subarray at a time.

- (40) **[Which walls]_i did John [assume [that [the cat scratched *t_i]] before complaining to his wife without [examining *pg_i]]?***

The problematic portion of the derivation of (40) is in (41). In these examples, the members of the active lexical subarray are shown. For example, in (41a), we see that we have the numeration for the phase associated with vP1. The structure in K is the result of building up that structure. Then, the derivation should be able to move on to vP3 in (41b). We see just *scratched* taken out of the subarray. Then in (41c), *which walls* is copied out of K and merged with *scratched*. Then in (41d), the rest vP3 is built up. From here, there are no more obstacles to the derivation of (40).

- (41) * $[_{CP4} [\text{which walls}]_i \text{ did John assume } [_{CP3} \text{ that the cat } [_{vP3} \text{ scratched } t_i]] \text{ before } [_{vP2} \text{ complaining to his wife without } [_{vP1} \text{ examining } pg_i]]]$

a. vP1 is constructed

$$\text{Num}_{vP1} = \{\text{PRO, v1, examining, which, walls}\}$$

$$K = [_{vP1} \text{ PRO v1 } [_{vP1} \text{ examining } [\text{which walls}]_i]]$$

b. LA for vP3 activated, *scratched* introduced into the derivation

$$\text{Num}_{vP3} = \{\text{the, cat, v3, scratched}\}$$

$$K = [_{vP1} \text{ PRO v1 } [_{vP1} \text{ examining } [\text{which walls}]_i]]$$

$$L = \text{scratched}$$

c. Sideward movement of *which walls*

$$\text{Num}_{vP3} = \{\text{the, cat, v3, scratched}\}$$

$$K = [_{vP1} \text{ PRO v1 } [_{vP1} \text{ examining } [\text{which walls}]_i]]$$

$M = [{}_{vP3} \text{scratched } [\text{which walls}]_i]$

d. Completion of $vP3$

$\text{Num}_{vP3} = \{\text{the, cat, } v3, \text{scratched}\}$

$K = [{}_{vP1} \text{PRO } v1 [{}_{vP1} \text{examining } [\text{which walls}]_i]]$

$O = [{}_{vP3} \text{the cat } v3 [{}_{vP3} \text{scratched } [\text{which walls}]_i]]$

Haida and Repp show in (41) that (40) can be derived without violating OneSub. We see in (41a) the entire subarray for $vP1$ is exhausted, and in (41d) that the entire subarray for $vP3$ is exhausted. Nunes rules out the derivation of islands within the parasitic domain by ensuring that the subarray for the parasitic domain is completely built before the sideward movement occurs. This typically works because the matrix subarray selects the adjunct subarray, and that adjunct subarray must be completed before this is done, ensuring that the adjunct island is present, thus barring movement from the island. However, in (41), we can see that it is possible to build up the island inside of the parasitic domain without adjoining it (thus allowing movement out), and then move on to the embedded clause in (41c). This is possible because the subarray that the derivation moves to in (41c) does not contain a lexical item that selects the parasitic domain. Therefore, it does not trigger the Spell-out of the adjunct. As long as there is no direct relationship between the subarray of the parasitic domain and the domain to which the WH phrase sideward moves, then the adjunct is never spelled out, and there should be no violation of the adjunct island. So in (41), we have no violations of OneSub, but we lack a way to account for the ungrammaticality of (40). It seems clear that the issue is that we are still getting around extracting out of a position that is too embedded. Haida and Repp's solution addresses this

problem. They suggest that in addition to OneSub, we need ActivateSelector, repeated here from Haida and Repp (2012) in (42):

(42) ActivateSelector

After a subarray has been exhausted and a phase been completed **and not yet spelled out**, the next subarray to be activated must be one that contains a selectional requirement for the syntactic object just completed (if there is such a subarray).

This addresses the issues in (41) by disallowing the derivation to move from vP1 to vP3 without spelling out vP1 (which would render *which walls* incapable of moving due to the PIC). This is because there is vP1 is not selected by anything in the subarray of vP3 (it is vP2 which has its selecting lexical item, *without*). In order to not spell out vP1, vP2 must be the built up immediately. Even if this allows vP1 to continue to not be spelled out, this will not work when trying to move on from vP2. The subarray that contains the selector for vP2 is CP4. However, this subarray cannot be built without CP3, and thus the derivation cannot proceed without spelling out vP2. As such, vP2 is spelled out, which contains vP1, thus rendering *which walls* invisible to movement due to the PIC.

Something that remains constant between both of these theories is the assumption that movement must take place from more embedded environments to less embedded environments. While perhaps intuitively reasonable, in Nunes' theory it is not explained where this restriction should come from, merely that it holds. ActivateSelector begins to address this concern by

stipulating a constraint on derivation such that structures that do not adhere to this are spelled out, thus ruling out certain derivations due to inaccessibility. We should endeavor to understand the reason that the derivation proceeds in this manner. It is true that in some cases, the embedded clause must be constructed before a higher clause can continue, since the higher clause demands the embedded clause as a constituent. So in these cases we have an explanation for this behavior. However, it is not clear why the derivation should proceed this way generally.

Further, it is not always clear which is the more embedded clause. Especially when considering a sentence like (40), how does the grammar know which phase is the more embedded environment when considering vP1 and vP3 (or CP3)? One might imagine that the process of creating subarrays might receive some sort of ordering or markings of relative embeddedness, but if this is in fact what is being done, the specifics of that process remain undescribed.

Lastly, there is an empirical issue for Haida and Repp's ActivateSelector theory. This theory predicts that once a parasitic domain has been built up, the next subarray that is built must be the one that selects it, since otherwise the parasitic domain will be spelled out. If it is the case that another subarray must be accessed before the selecting subarray, this should render the parasitic domain inaccessible, and predict a lack of parasitic gaps from such an adjunct. This is a problem, though, since it predicts the ungrammaticality of the sentence in (43a).

- (43) a. Which article did Bill read after Karen left the room without Mary recommending?
- b. [Which article]_i did [_{vP} [_{vP} [_{vP} Bill read *t_i*] after Karen left the room] without Mary recommending *pg_i*]?

In (43b) we see a sentence in which there is an adjunct to the vP other than the adjunct containing the vP. If the derivation built the parasitic domain first, it could not proceed to the matrix vP, since *after* would not have its complement built. It also cannot proceed to the other adjunct clause (*Karen left the room*), since that would activate a subarray that did not select the parasitic domain, causing it to spell out, predicting (43a) to be ungrammatical. If the non-parasitic adjunct were built first, this all might be avoided, since it can be spelled out without harm, and then the derivation can move to the parasitic domain, and then to the matrix vP. This would predict (43a) is grammatical.

While the solution above might work for (43), there is still a problem in that (44a) is also predicted to be ungrammatical.

- (44) a. Which article did Bill read after leaving the room without Mary recommending?
- b. [Which article]_i did [_{vP} [_{vP} [_{vP} Bill_j read *t_i*] after PRO_j leaving the room] without Mary recommending pg_i]?

Boeckx et al. (2010) give an account of adjunct control in which the control relationship between *Bill* and PRO is derived using sideward movement in much the same way that parasitic gaps are. Since Nunes (2012) and Haida and Repp (2012) both cite this without dissent, I will assume that this is the case for the current discussion. If sideward movement is to occur out of both of the adjuncts, then once again, it is predicted that (44a) would be ungrammatical. Since *Bill* (represented by PRO in (44b)) must be able to move to the matrix vP, *after leaving the room* cannot have spelled out before activating the subarray that selects it, namely the matrix vP.

However, the same is true for the parasitic domain. *Which article* must move to the matrix VP, it

cannot be spelled out until the matrix vP subarray is active. After either of the adjuncts is built, it must activate the matrix vP subarray. Neither the adjunct control clause and the parasitic clause may allow for the activation of the other before the activation of the matrix vP, or else they will be forced to spell out, predicting that (44a) should be ungrammatical. It also cannot be the matrix vP subarray that is built first, since it does not have the components necessary, and would predict the possibility of movement into the adjuncts if building the matrix vP first were an option. Thus ActivateSelector (combined with a theory of adjunct control featuring sideward movement) predicts (44a) to be ungrammatical, which it is not.

2.4 Conclusion

In the beginning of this chapter, I laid out the assumptions of the Minimalist Program as I see them, and followed that path to the current version of Merge, which is an operation that entails typical upward movement, and predicts the possibility of sideward movement. Sideward movement from one structure to the next without restriction poses a serious problem of overgeneration. In order to discuss some of the ways in which sideward movement can be restricted, I have taken Nunes' sideward movement theory of parasitic gaps as a case study for understanding these restrictions. As we can see from Nunes' theory, many restrictions come for free, such as the need for every movement (sideward or otherwise) to be motivated. Sideward movement cannot take any phrase and merge it to another structure for no purpose. This process must be driven by some syntactic need, such as the need to fulfill a thematic role. Also, copy operations cannot occur well before the subsequent merger of that copied object. This violates Last Resort, yet another of the restrictions that carry over from upward movement to restrict sideward movement.

Nunes also provides a discussion of the computational restrictions that are in place on sideward movement, such as the general principle that movement must occur from more embedded environments to less embedded environments. While some problems with this as an explanation were discussed and will be revisited in the next chapter, this account does restrict many types of potential overgeneration that are possible.

However, Nunes' theory is not without its own problems, be it lookahead problems, issues accounting for restrictions in the parasitic domain, or in the fact that there are yet more restrictions that must be placed on the derivation with respect to how the derivation proceeds from one subarray to the next, as noted by Haida and Repp's (2012) discussion of *ActivateSelector*. In addition to these issues, Nunes has to make assumptions about the derivation, such as the cyclic merger of adjuncts into the derivation. While disposing of late adjunction might provide a more parsimonious theory of the computational system, there is a great deal of evidence that suggests that late adjunction is necessary which will be discussed in Chapter 5. We should make sure that such evidence can be adequately accounted for before disposing of such an operation.

In the following chapter, I will begin to address the assumptions about the course of the derivation that Nunes and Haida and Repp have made, providing my own assessment of the course of the derivation and the properties it must have in order to ensure efficient computation. Then I will provide a sideward movement theory of parasitic gaps that will work within this computational system and which addresses some of the issues brought up in this chapter.

3 The course of the derivation and restricting sideward movement

3.1 Introduction

One of the primary concerns of the last chapter was how to capture the restrictions on the clauses or phases that syntactic objects can move between under a sideward movement theory of parasitic gaps. The data presented in the last chapter show that sideward movement can only bring an object from a more embedded position to a less embedded position. Nunes (2004, 2012) argues that this is inherent in the order that syntactic structures are built up (more embedded structures are built before less embedded structures), so therefore the directionality of movement is fixed. He also claims that other restrictions on the phases that objects can move between can be derived from the fact that the computational system can only act on one LA at a time, and that the computational system will exhaust a LA before moving on. I argued in the last chapter that since the relative embeddedness of two objects cannot be known until they are both part of a larger structure, the computational system cannot possibly know to proceed from more embedded to less embedded environments as Nunes suggests.

Additionally, Haida and Repp (2012) present empirical evidence that such a claim is not sustainable. If sideward movement is free between any two syntactic objects (assuming other conditions for legibility and linearization are later met), then how can it be that sideward movement cannot occur between any two clauses, such as the sentences in (1). In (1a) we see that there is an adjunct island inside of the parasitic domain, and in (1b) we see the same, but the true gap can be argued to be just as embedded as the parasitic gap.

- (1) a. *Which paper did you file after John left the room without reading?
 b. *Which walls did John assume that the cat scratched before complaining to his wife without examining?

In order to account for the ungrammaticality of sentences like (1), both Nunes (2004, 2012) and Haida and Repp (2012) create restrictions on the derivation such that the source of the moved object is unavailable by having been spelled out. However, as discussed at the end of the last chapter, on Nunes' part there is an empirical problem with (1b). Specifically, the derivation in (2) (repeated from (42) in the previous chapter) is problematic, since only allowing one subarray to be built at a time still allows for the sentence in (1b) to be constructed. Additionally, I discussed the conceptual problem of describing two parallel branches of a structure in terms of embeddedness.

- (2) * $[_{CP4} [\text{which walls}]_i \text{ did John assume } [_{CP3} \text{ that the cat } [_{VP3} \text{ scratched } t_i]] \text{ before } [_{VP2} \text{ complaining to his wife without } [_{VP1} \text{ examining } pg_i]]]$

- a. vP1 is constructed

$$\text{Num}_{vP1} = \{\text{PRO}, v1, \text{examining}, \text{which}, \text{walls}\}$$

$$K = [_{vP1} \text{ PRO } v1 [_{vP1} \text{ examining } [\text{which walls}]_i]]$$

- b. LA for vP3 activated, *scratched* introduced into the derivation

$$\text{Num}_{vP3} = \{\text{the}, \text{cat}, v3, \text{scratched}\}$$

$$K = [_{vP1} \text{ PRO } v1 [_{vP1} \text{ examining } [\text{which walls}]_i]]$$

L = scratched

c. Sideward movement of *which walls*

Num_{vP3} = {the, cat, v3, scratched}

K = [_{vP1} PRO v1 [_{vP1} examining [which walls]_i]]

M = [_{vP3} scratched [which walls]_i]

d. Completion of vP3

Num_{vP3} = {the, cat, v3, scratched}

K = [_{vP1} PRO v1 [_{vP1} examining [which walls]_i]]

O = [_{vP3} the cat v3 [_{vP3} scratched [which walls]_i]]

For Haida and Repp, I showed that ActivateSelector still has trouble finding the appropriate restrictions, as we can see from the example in (3).

- (3) [Which article]_i did [_{vP1} [_{vP2} [_{vP3} Bill_j read *t_i*] [_{PP1} after PRO_j leaving the room]] [_{PP2} without Mary recommending *pg_i*]]?

Specifically, (3) is predicted to be ruled out by ActivateSelector due to spelling out one of the adjuncts before moving on to build the other adjunct. However, the sentence is grammatical. The structure in (3) would be problematic without any additional assumptions if you assume PP2 is built before PP1, and then the matrix vP is made. If that were the case, since PP1 is not the syntactic object that PP2 is adjoined to, then PP2 would need to be spelled out according to

ActivateSelector. This means that *which article* would be unavailable for movement into the matrix clause, making (3) underivable.

One could argue that there is no reason to assume this ordering in the construction of the adjuncts. If PP1 were built before PP2, then PP2 would not need to be spelled out, and *which article* would sideward move into the relative clause. First of all, this assumes that there is no ordering to the construction of adjuncts to a particular phrase. While this assumption is reasonable, in theories such as Nunes', there may be some prescribed ordering to the construction of these adjuncts. For the sake of argument though, let's assume there is no such ordering. Both Nunes (2004, 2012) and Haida and Repp (2012) cite Hornstein (2001) for its sideward movement treatment of adjunct control. Fundamentally, the process is similar to the sideward movement theories of parasitic gaps as described in the previous chapter. In this case, *Bill* is moved into the matrix clause from PP1, where it will become the subject and form a chain with the copy in PP1, much like parasitic gaps are related to the matrix WH filler. Since both Nunes and Haida and Repp cite this analysis without rejecting it, I will assume that they wish to preserve such an analysis alongside their theories of parasitic gaps. If that is the case, both PP1 and PP2 must be able to have an object sideward move into the matrix vP (*Bill* for adjunct control from PP1 and *which article* for the parasitic gap in PP2). However, ActivateSelector demands that once one of those two is built, the derivation must either proceed immediately to the matrix clause (barring sideward movement from the other PP), or must spell out one of the PPs to move on to the other, making the DP that must move out unavailable for sideward movement. Both of these options incorrectly predict that (3) should be ungrammatical.

In this chapter, I will provide an account of sideward movement that properly restricts movement in sentences like (1) and (3). This will be achieved by restricting the availability of

objects based on their availability in the search space of the relevant operations. Additionally, one of the problems in accounting for islands within the parasitic domain is how to account for the potentially parasitic element is made unavailable for movement. I will show that these islands, can be accounted for using one general principle of Spell-out that is triggered by phase heads.

3.2 Problems with previous attempts

In order to account for the ungrammaticality of the sentences in (1), it is necessary to either (i) ensure that the derivation must build the parasitic domain first or (ii) make it so that movement from the matrix vP phase to the parasitic domain is toxic to the derivation. These restrictions ensure that a movement from the matrix structure to the parasitic domain is impossible, because if such a movement is possible, then finding principled ways of ruling out CED violations within the parasitic domain becomes harder, because one of the simplest ways of accounting for movement islands is to assume that there is movement. However, if the WH feature for the parasitic element can be satisfied in the matrix domain, and objects can move from the matrix clause to the parasitic domain, the motivation for movement of the parasitic element becomes unclear. It does not need to move to check its WH features because a copy in the matrix domain will accomplish that. Any restriction on the movement operation itself from embedded to unembedded environments would also be antithetical to the arguments that predict sideward movement in the first place, namely that the domain of select and merge are unrestricted.

Nunes (2004, 2012) relies exclusively on (i), suggesting that derivations always begin in more embedded contexts, so the building of the adjunct first can be taken for granted. This is

problematic, since to know that some structure you're building is more embedded is to know something about the structure to be built. This lookahead problem is fatal. It could be the case, however, that there are principles that might result in building more embedded structure first. Take for example the structure in (4).

(4) [XP X [YP Y [ZP Z]]]

It cannot be the case that XP was built before YP, since this would mean that the root node was not the target of Merge for some of the constituents of YP. This would be a violation of the Extension Condition (Chomsky 1995), which states that structures can only be extended at the root node. It would be conceivable that the derivation either tries to merge X with some object, but none has been built, and so the derivation tries to build something else. It could also be the case that if X is selected for Merge and there are not syntactic objects for it to merge with, then the derivation crashes. This would mean only derivations that happen to start at the correct point can proceed. Both of these possibilities would do the same with YP, though since Z has no complement, it could be merged directly with Y to form YP. In any structure, there must be some syntactic object that has no selectional requirements, and the derivation can proceed from that point. Once the derivation begins with Z, it will be able to proceed in building structure upward. Essentially, this forces that derivation to build up the most embedded structures first along the spine of the tree. In the context of the spine of the tree, Nunes can reasonably claim that the derivation proceeds from more embedded to less embedded contexts, since it is an unavoidable result of the Extension Condition.

It is not obvious how this can be true for adjuncts, however. It is the case that an adjunct is embedded in a phrase of the category it is adjoined to, as in the example in (5).

(5) [XP XP₁ YP]

However, both XP₁ and YP must be built before they can be merged. The Extension Condition does not indicate which should be built first,⁵ since they both must be objects in the derivation before they are merged. While we know from the structure in (5) that YP is embedded in an XP, this cannot be known until after both structures are built. Therefore, the knowledge that YP is embedded can only come from some sort of lookahead in structure building. It would seem this means that the theory that assumes that the computational system knows to build adjuncts first cannot know this based on their adjuncthood, since that is a property of the structure, which it should have no access to. While it is trivial to say that the spine of a structure begins by building more embedded structures first due to the nature of the Extension Condition, the same cannot be said for adjuncts.

Haida and Repp's (2012) ActivateSelector is an attempt at using the second method, which is to make movement to the parasitic domain from the matrix phase toxic. This is achieved by assuming that there is a Spell-out operation that eliminates entire phases from the derivational workspace and allows for them to be merged into later structure, much like Nunes' account of CED islands. One can only proceed to a phase that does not select the adjunct (which would be the matrix vP) if this Spell-out operation has occurred. This succeeds in ruling out the sentences in (1) by making derivations that begin in the spine of the tree toxic by forcing a complete Spell-out of the spine to build the adjunct subarray (since it does not select the matrix clause). While effective in accounting for the sentences in (1), forcing only certain derivational paths results in predicting the grammaticality of (3), since moving from one adjunct to the other

⁵ If indeed the Extension Condition applies at all to adjuncts, but for the moment I will assume it does.

forces the Spell-out of one of the adjuncts, making sideward movement accounts for the parasitic gaps adjunct control impossible to do simultaneously.

3.2.1 Using availability to restrict movement

My account of parasitic gaps will use the availability of the phrase to be moved in order to properly restrict sideward movement. Since there are conceptual issues with ensuring that the derivation can only proceed from more embedded to less embedded environments, it seems necessary to rule out derivations in which movement proceeds from the matrix clause out. However, doing this in terms of the unavailability of whole phases seems to be problematic due to the problems that Haida and Repp's (2012) ActivateSelector theory has with sentence like (3). I propose that the process of copying makes the object that the (original) copy is made from unavailable for further copying. This always leaves a copy in the derivation (the higher copy in upward movement, or the new copy more generally). This would account for the lack of movement into the parasitic domain. If an object in the matrix clause is copied into the parasitic domain, then the copy in the matrix domain would be made unavailable. Since the copy in the parasitic domain will be inside of an adjunct island, it cannot be moved back into the matrix clause. Therefore, copying from the matrix clause into any island would render further movements through the matrix clause impossible, since there would be no copies to move that did not violate island constraints somehow.

Let us examine an example of this in understanding why this forces a normal parasitic gap construction as in (6) to have the WH phrase move from the parasitic domain to the matrix clause. The derivation in (7) shows how this might proceed normally, with unavailable material struck out.

(6) Which paper did you file after John read?

(7) a. Adjunct vP built

$K = [_{VP} \text{John read } [\text{which paper}]_i]$

b. WH movement of *which paper*

$L = [_{VP} [\text{which paper}]_i [_{VP} \text{John read } \{\text{which paper}\}_i]]$

c. Completion of the adjunct

$M = \text{after } [\text{which paper}]_i \text{John } [_{VP} \{\text{which paper}\}_i [_{VP} \text{John read } \{\text{which paper}\}_i]]$

d. Introduction of *file*

$M = \text{after } [_{CP} [\text{which paper}]_i [_{TP} \text{John } [_{VP} \{\text{which paper}\}_i [_{VP} \text{John read } \{\text{which paper}\}_i]]]]$

$N = \text{file}$

e. Sideward movement of *which paper*

$M = \text{after } [_{CP} \{\text{which paper}\}_i [_{TP} \text{John } [_{VP} \{\text{which paper}\}_i [_{VP} \text{John read } \{\text{which paper}\}_i]]]]$

$O = [_{VP} \text{file } [\text{which paper}]_i]$

f. Completion of matrix vP and WH movement of *which paper*

M = after [CP ~~[which paper]_i] [TP John [vP ~~[which paper]_i] [vP John read
[~~which paper]_i]]]]]~~~~~~

P = [vP [which paper]_i] [vP you file ~~[which paper]_i]]]~~

g. Adjunction of the parasitic PP to the matrix vP

Q = [vP [vP [which paper]_i] [vP you file ~~[which paper]_i]]] after [CP ~~[which
paper]_i] [TP John [vP ~~[which paper]_i] [vP John read [~~which paper]_i]]]]]]]~~~~~~~~

In (7a), we can see the parasitic vP has been built up. In (7b), *which paper* moves to the edge of the vP to enable later WH movement. This process leaves the lower copy unavailable for further movement. Then in (7c) the rest of the adjunct is created, include the last movement of *which paper* to SpecCP and the movement of the subject to SpecTP. Both of these movements result in the unavailability of the lower copies. Next, the matrix V is introduced in (7d). In (7e), *which paper* sideward moves to the matrix clause, which once again causes the object is was copied from to be unavailable to later computation. The rest of the matrix vP is built up in (7f), including moving *which paper* to the vP for later WH movement, once again making the lower copy unavailable. In (7g) the matrix clause and the adjunct clause are merged together. In Nunes and Haida and Repp's analyses, this might have caused Spell-out. I will not be assuming this part of their theories, but whatever the theory used, it must be the case that further WH movement out the adjunct is illicit. For the moment, in (7), this is accomplished by there being no copies available in the adjunct clause. This will not be sufficient later, but for now this is fine.

In (8) we can see the derivation that would result if the WH phrase were to move into the parasitic domain from the matrix clause.

(8) a. Build matrix vP

$$K = [{}_{VP} [{}_{which\ paper}]_i [{}_{VP}\ you\ file\ {}_{which\ paper}]_i]$$

b. Introduce adjunct verb *read*

$$K = [{}_{VP} {}_{which\ paper}]_i [{}_{VP}\ you\ file\ {}_{which\ paper}]_i]$$

$$L = read$$

c. Sideward movement of *which paper*

$$K = [{}_{VP} {}_{which\ paper}]_i [{}_{VP}\ you\ file\ {}_{which\ paper}]_i]$$

$$M = [{}_{VP}\ read\ [{}_{which\ paper}]_i]$$

d. Build up the adjunct PP

$$K = [{}_{VP} {}_{which\ paper}]_i [{}_{VP}\ you\ file\ {}_{which\ paper}]_i]$$

$$N = after\ [{}_{CP}\ [{}_{which\ paper}]_i\ [{}_{TP}\ John\ [{}_{VP}\ {}_{which\ paper}]_i\ [{}_{VP}\ John\ read\ [{}_{which\ paper}]_i]]]$$

e. Adjunction of PP to matrix vP

$$O = [{}_{VP}\ [{}_{VP}\ {}_{which\ paper}]_i\ [{}_{VP}\ you\ file\ {}_{which\ paper}]_i] after\ [{}_{CP}\ [{}_{which\ paper}]_i\ [{}_{TP}\ John\ [{}_{VP}\ {}_{which\ paper}]_i\ [{}_{VP}\ John\ read\ [{}_{which\ paper}]_i]]]$$

In (8a) we see the matrix vP is built up, and the WH phrase has moved to the edge of the matrix vP, making the lowest copy unavailable. We will later see that this movement is not responsible for the problematic derivation, but I will assume this movement happens for consistency. In (8b) the adjunct verb is introduced. Since there is no restriction on sideward

movement between these two positions (given that the computational system has no information about which one belongs where in the structure), sideward movement occurs in (8c), but the copy in K is now unavailable for further movement. Then the adjunct clause is built up in (8d) much as it was in (7), to produce the object N. In (8e), the adjunct is merged with the matrix vP. From (8e) we can see that the only copy of *which paper* that is unavailable is the one in the adjunct clause. However, given the ungrammaticality of (9), it cannot be the case that this copy can move to SpecCP in the matrix clause. Accounting for the inability adjunct islands will be one of the goals of the next section, but for now let us assume that the same mechanism that makes the adjunct in (9) an island would do so for (6), thus barring the derivation in (8) from converging.

(9) *Which paper did you file the report after John read?

I would now like to take a moment to discuss the nature of the unavailability that I have so far been assuming in this section. There have been many arguments that previous copies are unavailable for movement. An interesting example is the case of ‘freezing’, where movement out of derived positions is not possible. This phenomenon has been accounted for in many ways (Wexler and Culicover 1980, Kitahara 1994, Takahashi 1994, Boeckx 2003, Rizzi 2006, Stepanov 2007, Müller 2009, Müller 2011, Hofmeister et al. 2015), but one aspect that is interesting is that both movement out of the derived position as well as the original position must be barred in order to fully account for the unavailability. This is easy to see in an example like (10). This is derived in (11), with the resulting structure represented in (11d).

(10) *What did John take a picture yesterday of?

(11) a. vP built

K = [_{vP} John take a picture of what]

b. *Yesterday* adjoined to vP

L = [_{vP} [_{vP} John take a picture of what¹] yesterday]

c. Extraposition of *of what*

M = [_{vP} [_{vP} [_{vP} take a picture [of what¹]_i] yesterday] [of what²]_i]?

d. Completion of CP

N = [_{CP} What³ did John [_{vP} [_{vP} [_{vP} take a picture [of what¹]_i] yesterday] [of what²]_i]]?

In (11a), the vP is built up. Then in (11b) *yesterday* is adjoined. Next, in (11c), *of what* is extraposed to the edge of that vP. Lastly, we might expect that *what* can move to the specifier of CP. The ungrammaticality of (10) shows us that something in this derivation is incorrect. We know that complements can be extraposed as in (11c). So why can't *what* be moved to the specifier of CP? We can see that (11d) contains two copies of *what* that might be able to be copied. Accounts of freezing can explain why *what*² cannot be moved under the umbrella of restrictions on movement from inside of moved objects. However, there also must be a restriction also against movement from *what*¹ to the *what*³ position. Many of these theories were developed within a theory of movement that utilizes traces to account for this. However, in a copy theory of movement, *what*¹ still must be restricted from being able to move. This restriction seems similar to the one that we are trying to derive in the parasitic gap domain.

One possibility is to say that much like other forms of unavailability, these moved objects are spelled out. This view would certainly accomplish the goal of unavailability, but adds a

Spell-out operation to the process of movement. On the other hand, Spell-out is already an operation we have independent justification for, much like the other operations involved in movement, (Select, Copy, Merge). Another possibility is that there is some other sort of freezing that takes place that makes the target unavailable for movement, either at the feature level or invokes some memory of moved objects that must later be avoided. Why the derivation might undergo these new operations is not clear. However, if we imagine Spell-out to be an operation meant to reduce the size of the syntactic objects in the derivational workspace, then spelling out the initial copy of a moved object is quite reasonable, since the derivation will retain a copy for later use. Therefore, I will assume that the unavailability that results from movement is the caused by the computation spelling out previous copies of a moved object in order to reduce the size of the derivation workspace. Even though the addition of another condition that can triggers Spell-out may be less than ideal, we can motivate the removal of previous copies under an appeal to reducing the amount of extraneous material in the derivational workspace. This new trigger also can act generically on all moved objects in the derivational workspace, and requires no advance knowledge of the interface conditions to apply. This cannot be said for Nunes' (2004, 2012) or Haida and Repps' (2012) account of the directionality of the derivation.

Lastly, the rule that makes objects that have been copied unavailable must work in conjunction with a modified version of ActivateSelector. While it is necessary for the copies to become unavailable, a crucial component of the ActivateSelector theory was ensuring that the derivation would proceed from building the adjunct to building up either the phase that it would attach to or (in the case of sentences like (3)) some other phase that would need to attach to it. This is necessary to rule out the derivation of the sentence in (1a) given in (2), since it would still be possible to derive (1a) if ActivateSelector did not ensure that the derivation moved from an

adjunct to the phase that selected it instead of to any other phase in the derivation. This modified version of ActivateSelector is given below in (12).

(12) ActivateSelector (modified)

After a phase has been completed, the next phase to be built must be one that selects for the syntactic object just completed or one of its subparts.

The wording about subarrays has been removed in order to be agnostic about the use of subarrays specifically, but this restriction on the derivation ensures that the phase that is built next is the one that selects it or another object that is contained within that phase. What is crucially omitted is any wording entailing a Spell-out for the entire phase that would be used especially in these circumstances, since I will later account for CED islands with a generic, cyclic rule for Spell-out.

The combination of this modified version of ActivateSelector and the Spell-out of old copies gives us the means to restrict the derivation in such a way that (1b) cannot be derived, but (3) still can. While ActivateSelector eliminates the derivation in (2), it should still allow the matrix vP to be built up first. This derivation is shown in (13)

(13) *Which walls did John assume that the cat scratched before complaining to his wife without examining?

a. Matrix vP constructed

$K = [_{vP} [\text{which walls}]_i [_{vP} \text{John assume } [_{CP} \text{which walls}]_i \text{ that the cat scratched } [\text{which walls}]_i]]]$

b. Introduction of the embedded adjunct island verb *examining*

K = [_{VP} [which walls]_i [_{VP} John assume [_{CP} ~~[which walls]_i~~ that the cat scratched [~~which walls~~_i]]]]

L = examining

c. Sideward movement of *which walls* into the adjunct island

K = [_{VP} ~~[which walls]_i~~ [_{VP} John assume [_{CP} ~~[which walls]_i~~ that the cat scratched [~~which walls~~_i]]]]

M = [_{VP} examining [which walls]_i]

d. Build up of the adjunct island PP

K = [_{VP} ~~[which walls]_i~~ [_{VP} John assume [_{CP} ~~[which walls]_i~~ that the cat scratched [~~which walls~~_i]]]]

N = [_{PP} without [_{TP} PRO examining [which walls]_i]]

e. Construction of the adjunct PP

K = [_{VP} ~~[which walls]_i~~ [_{VP} John assume [_{CP} ~~[which walls]_i~~ that the cat scratched [~~which walls~~_i]]]]

O = [_{PP} without [_{TP} PRO examining [which walls]_i]]

P = [_{VP} PRO complaining to his wife]

f. Adjunction of the adjunct island PP to the adjunct vP

K = [_{VP} ~~[which walls]_i~~ [_{VP} John assume [_{CP} ~~[which walls]_i~~ that the cat scratched [~~which walls~~_i]]]]

Q = [_{vP} [_{vP} PRO complaining to his wife] [_{PP} without [_{TP} PRO examining
[which walls]_i]]]

In (13a) we see the embedded clause and the matrix clause built up, with *which walls* moved up to the edge of the matrix vP. Notably, each of the lower copies is spelled out and unavailable for further computation. This is not a problem because *which walls* is still present in the highest position. In (13b), the verb of the most embedded adjunct is introduced into the derivation. This is not a problem for ActivateSelector, since the matrix vP has not been finished yet, and the adjunct being built will become a subpart of the phase being created. Then *which walls* is copied and merged to examining in (13c), which causes the Spell-out of the copy in the matrix clause. This will be crucial later. In (13d) the rest of the adjunct is built up. Then in (13e), the vP that O will eventually be merged to is built up. O and P are merged to produce Q in (13f), and we can now see why this derivation will fail to converge. At this point in the derivation, we are left with no available copies of *which walls* in the matrix clause. In addition, the only copy that is available is inside of an adjunct within the parasitic domain Q. Whatever theory of adjunct islands that is presumed, *which walls* should not be able to move out of this configuration. I will discuss the details of how adjunct islands are made unavailable in the derivational workspace in section 3.3, but for now, if we assume that the copy in Q is unavailable, and the spelled out copies in K are unavailable, then there are no available copies to move into the specifier of the matrix CP, and the WH feature on *which walls* will never be checked. Therefore, the derivation crashes, and (1b) is correctly predicted to be ungrammatical.

On the other hand, each of the adjuncts in (3) can be built up one at a time since ActivateSelector allows another part of the matrix phase to be built after the first adjunct is built. This derivation is given below in (14).

(14) [Which article]_i did [_{vP1} [_{vP2} [_{vP3} Bill_j read *t_i*] [_{PP1} after PRO_j leaving the room]] [_{PP2} without Mary recommending pg_i]]?

a. Construction of PP2

K = [without Mary recommending [which article]_i]

b. Construction of PP1

K = [_{PP} without Mary recommending [which article]_i]

L = [_{PP} after Bill_j leaving the room]

c. Introduction of the matrix verb *read*

K = [_{PP} without Mary recommending [which article]_i]

L = [_{PP} after Bill_j leaving the room]

M = read

d. Sideward movement of *which article*

K = [_{PP} without Mary recommending [~~which article~~]_i]

L = [_{PP} after Bill_j leaving the room]

N = [_{VP} read [which article]_i]

e. Matrix *v* merged

K = [_{PP} without Mary recommending [~~which article~~]_i]

L = [_{PP} after Bill_j leaving the room]

O = [_{v'} read [~~read~~ [which article]_i]

f. Sideward movement of *Bill*

K = [_{PP} without Mary recommending [~~which article~~_i]

L = [_{PP} after ~~Bill~~_j leaving the room]

P = [_{VP} Bill_j read [~~read~~ [which article]_i]

g. WH movement of *which article*

K = [_{PP} without Mary recommending [~~which article~~_i]

L = [_{PP} after ~~Bill~~_j leaving the room]

Q = [_{VP} [which article]_i] [_{VP} Bill_j read [~~read~~ [~~which article~~_i]]]

h. PP1 adjoins to matrix vP

K = [_{PP} without Mary recommending [~~which article~~_i]

R = [_{VP} [_{VP} [which article]_i] [_{VP} Bill_j read [~~read~~ [~~which article~~_i]]] [_{PP} after ~~Bill~~_j leaving the room]]]

i. PP2 merges to matrix vP

S = [_{VP} [_{VP} [_{VP} [which article]_i] [_{VP} Bill_j read [~~read~~ [~~which article~~_i]]] [_{PP} after ~~Bill~~_j leaving the room]]] [_{PP} without Mary recommending [~~which article~~_i]]]

j. Matrix TP is built, *Bill* moved to specifier of TP

T = [TP Bill_j did [_{VP} [_{VP} [_{VP} [which article]_i] [_{VP} Bill_j read [read [which article]_i]]] [_{PP} after Bill_j leaving the room]]] [_{PP} without Mary recommending [which article]_i]]

k. CP is built and *which article* is moved to specifier of CP

U = [_{CP} [which article]_i did [TP Bill_j ~~did~~ [_{VP} [_{VP} [_{VP} ~~[which article]_i] [_{VP} Bill_j read [read [which article]_i]]] [_{PP} after Bill_j leaving the room]]] [_{PP} without Mary recommending ~~[which article]_i]]]~~~~

In (14a), PP1 is built up, and in (14b), PP2 is built up. Since PP2 will become part of the eventual matrix vP, this is not a problem for ActivateSelector. In (14c), the matrix verb is introduced. We see the sideward movement of *which article* in (14d), which spells out the copy within PP2. Then the matrix *v* is introduced in (14e), and *Bill* is moved into the specifier of vP in (14f), which spells out the copy within PP1. In (14g), *which article* is moved to the edge of the vP for later WH movement (spelling out the lower copy). Then PP1 is adjoined to the matrix vP in (14h) and PP2 is adjoined to the matrix vP in (14i). After this, the matrix TP is built up, and *Bill* is moved to the specifier of TP in (14j). This causes the Spell-out of the copy within the matrix vP. Then in (14j), *which article* moves to the specifier of CP, finally checking its WH feature.

As we can see in the derivation of (13) and (14), the combination of this modified version of ActivateSelector and the Spell-out of old copies can properly restrict the directionality of the movement of parasitic elements in (1b), while still allowing for multiple adjuncts to be formed in (3). With the directionality of movement in parasitic gap constructions accounted for, I will

address the problem of island effects, both within and outside of parasitic gaps, in the next two sections.

3.2.2 Restricting movement out of adjunct islands

In this section I will provide an account for adjunct islands. In Nunes (2004, 2012), Nunes and Uriagereka (2000), and Haida and Repp (2012), a special Spell-out operation is applied to adjuncts that applies to the entire adjunct, but still allows that object to be merged as a lexical item into the derivation. This section will discuss the problems with such an account, as well as discuss the necessary phase architecture and Spell-out mechanisms in order to account for adjunct islands through a single, general purpose Spell-out mechanism that is applied to all phases.

3.2.2.1 Agree and A' movement

While the previous section discussed an important aspect of the movement mechanism in the Spell-out of copied object, there is a problematic derivation caused by the freedom of sideward movement. Since a WH phrase in the object position moves through the vP edge in order to later be moved to the CP edge, it must be the case there is a position here more generally to move to. This poses a problem in the derivation of (15). (15a) skips to the relevant portion of the derivation, where the adjunct has been built up in its entirety, and the matrix vP has been constructed as well. Since the vP can select and copy a WH phrase to adjoin to it, what stops this from happening in (15b)?

(15) *Which paper did you file the report after John read?

a. Matrix vP and adjunct PP constructed

K = [_{VP} you file the report]

L = after [_{CP} [which paper]_i [_{TP} John [_{VP} ~~[which paper]_i~~ [_{VP} John read [~~which paper]_i~~]]]]]

b. Sideward movement of *which paper* to vP edge

M = [_{VP} [which paper]_i [_{VP} you file the report]]

L = after [_{CP} ~~[which paper]_i~~ [_{TP} John [_{VP} ~~[which paper]_i~~ [_{VP} John read [~~which paper]_i~~]]]]]

If WH phrases are capable of sideward movement into this position, it would be impossible to restrict sideward movement out of adjunct islands. If sideward movement is completely free, then this movement should be allowed. How can movement to this position be restricted to objects inside of the vP? While Merge cannot be restricted in this way, Agree is constrained to searching within the structure it is a member of. It is clear that the search domain for Agree cannot be the same as Merge, since Merge can find targets in other structures and in the Numeration to find syntactic object. We also never see agreement into adjuncts. If it is the case that movement to A' positions is driven by agreement, then this may provide the answer. Since Agree can only look inside of the structure given, there should be no way of sideward moving to an A' position.

This is also true after the adjunct is merged. We can see an example in (16a). Agreement always follows c-command relationships. Since *v* does not c-command into the adjunct, *which paper* cannot move as it does in (16b).

(16) a. PP adjoined to vP

K = [_{VP} [_{VP} you file the report] after [_{CP} [which paper]_i [_{TP} John [_{VP} ~~which paper~~_i [_{VP} John read ~~which paper~~_i]]]]]

b. Movement of *which paper* to the vP edge

L = [_{VP} [which paper]_i [_{VP} [_{VP} you file the report] after [_{CP} ~~which paper~~_i [_{TP} John [_{VP} ~~which paper~~_i [_{VP} John read ~~which paper~~_i]]]]]]]

3.3 Triggers and targets of Spell-out

In order to build up a theory of parasitic gaps that will account for parasitic gaps, adjunct islands, island effects within the parasitic domain, as well as rule out the sentences that are problems for Nunes and Haida and Repp, I am going to need several crucial parts to come together.

- (17) a. WH phrases begin in the parasitic domain
- b. WH phrases move to the edge of the parasitic CP in order to ‘escape’ the parasitic domain.
- c. A single, general purpose Spell-out rule will account for the unavailability of adjunct islands as well as WH islands.

(17a) was the subject of the beginning of this chapter. The Spell-out of the object that is copied ensures that objects that move into islands will not leave syntactic objects that can move. (17b) is a necessity in order to ensure that the movement to the edge of the parasitic domain is motivated. Much like WH phrases move up through the edges of the phases along the spine of the structure in order to eventually check their WH feature in the final CP they land in, parasitic

WH phrases move up to the edge of the parasitic domain in order to be extracted through sideward movement. If not for this motivation, there would be no reason that these WH phrases moved within the parasitic domain, and accounting for the island effects would become much more difficult. It also cannot be the case that WH phrases in the parasitic domain would move in order to check their WH feature at the edge of the parasitic domain, since if that were possible, we would expect WH questions within adjunct, which we do not see.

One of the primary goals of this theory is (17c). This comes out of the conceptual problem with the Spell-out processes used by Nunes (2004, 2012) and Haida and Repp (2012). Recall that these theories rely on the unavailability of objects inside adjunct islands to be a result of the adjunct's status as an adjunct. This is a problem since most phases spell out automatically based on some trigger (be it the completion of the phase itself or the creation of a higher phase). Also, the target in this cyclic Spell-out is typically the complement of the head of a phase. These theories suggest that adjuncts behave differently in order to satisfy linearization requirements. However, these requirements aren't cannot be within the narrow syntax, since they refer directly to linearization, which is outside of the scope of the narrow syntax. They rely on a type of looking ahead to the requirements of the interface, and this is problematic. Additionally, the target for Spell-out cannot be the adjunct as a whole, but some sub-portion that contains the objects that cannot move out, or else there would be nothing left to adjoin to the matrix vP. In order to solve this issue, we will need to consider both the targets and the triggers of Spell-out.

There are two aspects of the target question that must be considered. Chomsky (2000) gives a definition of the PIC as in (18). If we think the source of this inaccessibility is Spell-out, then this definition makes the complement of the phase-head the target of Spell-out.

- (18) In phase α with head H, the domain of H is not accessible to operations outside α ; only H and its edge are accessible to such operations.

In Nunes (2004, 2012), we see that Spell-out is in fact applying to a whole CP when adjuncts spell out for linearization reasons. The target of Spell-out is the whole phase before it has been attached, not just the complement of the phase-head. This is justified because Nunes (2012) and Nunes and Uriagereka (2000) say it leaves a label that can be merged. Unifying islands under the umbrella of the PIC would be preferable to the two types of Spell-out operations accounting for it in other theories, especially since one is targeted at islands specifically and relies on lookahead to be triggered.

The second aspect of the target of Spell-out to consider is which phase is the target of cyclic Spell-out. Chomsky (2001) describes higher phases as causing the inaccessibility of parts within a lower phase. In terms of Spell-out, this would mean that the higher phase causes the constituents of the lower phase to spell out. However, the Spell-out of adjuncts in Nunes and Uriagereka (2000) shows that perhaps a phase can target its own constituents for Spell-out. Conceivably a phase could spell out its own constituents as long as there is the possibility for cyclic computation.

In addition to the targets of Spell-out, we need to consider the possible triggers of Spell-out. Is it the completion of a phase that causes Spell-out or the introduction of a phase head?

Lastly, we must consider what heads and phrases constitute phases. While CP and vP are widely considered phases, are there other phases that are relevant? We will see that it is necessary to show that some PPs are in fact phases.

These four questions, restated in (19) leave us with a four-dimensional possibility space that must be explored, which I visualize in Table 1.

- (19)
- a. Do phases spell out their constituents or constituents of lower phases?
 - b. Is it the complement of the phase-head or the maximal projection of the phase-head that spells out?
 - c. Is it the merger of the phase-head that triggers Spell-out or the completion of the phase-head's phrase?
 - d. Is P a phase head?

Table 1: The four-dimensional possibility space for phase architectures

Target: Same or lower phase	Target: Complement or entire phrase	Trigger: Completion of phase or merger of phase-head	P a phase-head?
Phases target themselves	Entire phrase	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No
	Complement of phase-head	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No
Phases target lower phases	Entire phrase	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No
	Complement of phase-head	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No

I will explore this possibility space explicitly, going through three types of architecture failure. In the end, we will see that only one of the possibilities left will work for a theory of parasitic gaps that doesn't predict movement out of adjunct islands. Specifically, our theory should be able to predict the ungrammaticality of (20a), while still being able to derive (20b).

- (20) a. *[Which book]_i did Bob read a review after Mary recommended *t_i*?
- b. *[Which book]_i did Bob read *t_i* after Mary recommended *pg_i*?

3.3.1 Type 1 failure: No cyclic computation

Some settings for the parameters in (19) will produce a grammar that doesn't allow for cyclic computation. If a phases target their own maximal projection for Spell-out, then nothing will remain in the derivation to build up the structure. Similarly, if a phase spells out its own complement, it cannot do it as early as the introduction of its own phase-head, or else there will be no opportunity for movement out of that phase. This is less severe than having nothing left, but there cannot be any meaningful cyclic computation and movement in a system with those parameters.

One more system of settings that fails to allow for cyclic computation is when higher phases spell out the entire maximal projection of a lower phase, upon introduction of the higher phase head. This will prevent any meaningful cyclic computation between phases, since so much of that cyclic computation occurs by phrases moving from the 'edge' of one phase, namely its specifier or WH phrases moved to adjoin to vP, to another. Let's look at an example in (21).

(21) Which book did Mary recommend?

a. vP built up

K= [_{vP} Mary recommended [which book]_i]

b. *which book* moved to the edge of the vP

L= [_{vP} [which book]_i [_{vP} Mary recommended [~~which book~~]_i]]

c. TP constructed

M= [_{TP} Mary T [_{vP} [which book]_i [_{vP} Mary recommended [~~which book~~]_i]]]

d. C merged into structure

N= [C' C [TP Mary T [_{vP} [which book]_i] [_{vP} Mary recommended [~~which~~
~~book~~_i]]]]]

e. Spell-out of vP

O= [C' C [TP Mary T [_{vP} [~~which book~~]_i] [_{vP} Mary recommended [~~which~~
~~book~~_i]]]]]

In (21a) the vP is built up as normal. In (21b) we need to move *which book* higher in the phrase for later movement (though we will see not even this will help). (21c-d) build up the rest of the clause until the introduction of another phase-head. This triggers the Spell-out of the lower phase. We can see in the crucial step, (21e), *which book* has been spelled out, and is no longer available for WH movement. This system will not allow cyclic WH movement from low in the clause, not to mention cyclic movement from an embedded clause. These parameter settings cannot be right either.

In Table 2, we can see the settings that have been eliminated here are greyed out. I will continue to will the tables in as we go through the other two types of architecture failure.

Table 2: Phase architectures eliminated due to a lack of cyclic computation

Target: Same or lower phase	Target: Complement or entire phrase	Trigger: Completion of phase or merger of phase-head	P a phase-head?
Phases target themselves	Entire phrase	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No
	Complement of phase-head	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No
Phases target lower phases	Entire phrase	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No
	Complement of phase-head	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No

3.3.2 Type 2 failure: Movement out of adjuncts

Since one of the goals of this theory of parasitic gaps is to capture adjunct islands with cyclic Spell-out rules, it must be the case that our phase architecture can rule out sentences like (20a).

This type of overgeneration is predicted under a few types of phase architectures. This class of structures centers around when SpecCP in the adjunct clause (where we expect that

which book move to escape from the adjunct clause) is still available when WH movement to the matrix SpecCP occurs. If island effects are to be accounted for with Spell-out, we should avoid the introduction of other mechanisms that are adjunct specific in order to account for adjunct islands. Therefore, if there is an adjunct island effect, it should be the case that the movement is illicit because the moved phrase has spelled out before it could have moved. There are five parameter settings that will result in this type of overgeneration, all of which will look similar in this respect. First let us look at the four that occur if we do not assume any phase-heads other than C and *v*.

It cannot be the case that a phase spells out its own complement once the entire phase is complete. The relevant derivation is shown below in (22).

(22) a. Adjunct vP constructed

$$K = [{}_{VP} \text{ Mary recommended } [{}_{VP} \text{ recommended } [{}_{\text{which book}}]_i]]$$

b. *Which book* moved to the vP edge

$$L = [{}_{VP} [{}_{\text{which book}}]_i [{}_{VP} \text{ Mary recommended } [{}_{VP} \text{ recommended } [{}_{\text{which book}}]_i]]]$$

c. Spell-out of adjunct VP

$$L = [{}_{VP} [{}_{\text{which book}}]_i [{}_{VP} \text{ Mary recommended } [{}_{VP} \text{ recommended } [{}_{\text{which book}}]_i]]]$$

d. Adjunct built until the introduction of C

M = [_{C'} C [_{TP} Mary [_{VP} [~~which book~~]_i [_{VP} ~~Mary~~ recommended [_{VP} recommended [~~which book~~]_i]]]]]

e. WH movement of *which book*

N = [[~~which book~~]_i C [_{TP} Mary [_{VP} [~~which book~~]_i [_{VP} ~~Mary~~ recommended [_{VP} recommended [~~which book~~]_i]]]]]

f. Spell-out of TP

N = [[~~which book~~]_i C [_{TP} ~~Mary~~ [_{VP} [~~which book~~]_i [_{VP} ~~Mary~~ recommended [_{VP} recommended [~~which book~~]_i]]]]]

g. *After* merged into structure

O = [_{PP} *after* [[~~which book~~]_i C [_{TP} ~~Mary~~ [_{VP} [~~which book~~]_i [_{VP} ~~Mary~~ recommended [_{VP} recommended [~~which book~~]_i]]]]]

h. Construction of matrix vP

O = [_{PP} *after* [[~~which book~~]_i C [_{TP} ~~Mary~~ [_{VP} [~~which book~~]_i [_{VP} ~~Mary~~ recommended [_{VP} recommended [~~which book~~]_i]]]]]

P = [_{VP} Bob read [_{VP} ~~read~~ a review]]

i. Adjunct merged to vP

Q = [_{VP} [_{VP} Bob read [_{VP} ~~read~~ a review] [_{PP} *after* [[~~which book~~]_i C [_{TP} ~~Mary~~ [_{VP} [~~which book~~]_i [_{VP} ~~Mary~~ recommended [_{VP} recommended [~~which book~~]_i]]]]]]]

j. Spell-out of VP

R = [_{VP} [_{VP} Bob read [~~_{VP} read a review~~] [_{PP} after [[which book]_i C [~~_{TP} Mary~~
~~[_{VP} [which book]_i [_{VP} Mary recommended [~~_{VP} recommended [which~~
~~book]_i]]]]]]]]]]~~~~

k. Matrix clause constructed until the introduction of C

S = [C' did [_{TP} Bob ~~did~~ [_{VP} [~~Bob~~ read [~~_{VP} read a review~~] [_{PP} after [[which
~~book]_i C [~~_{TP} Mary~~ [_{VP} [which book]_i [_{VP} Mary recommended [~~_{VP}~~
~~recommended [which book]_i]]]]]]]]]]]]~~~~

I will describe this derivation in greater detail than some of the later derivations to make clear the mechanisms in place. In (22a) the adjunct vP is built up. The WH phrase then moves to the edge of the vP in (22b). Then in (22c), since the vP phase has been completed, the complement of the phase head should spell out. In this case, this is only the VP complement of *v* is spelled out. Since the moved verb and the moved WH phrase are all that this contains, the change here is vacuous. Then in (22d), the adjunct TP is built and C is merged to the structure. Next, *which book* is moved from the edge of the vP to SpecCP in (22e). The completion of the CP phase triggers the Spell-out of the complement of the C head in (22f), which in this case is the TP. Then the last part of the adjunct is attached, which is the preposition *after* in (22g). In (22h), the entire matrix vP is created, except for the merger of the adjunct, which occurs in (22i). Now that the vP is complete, the complement to *v*, in this case the matrix VP is spelled out in (22j). Then the rest of the structure is built up to the introduction of the matrix C in (22k).

It is at this stage in the derivation that the problem becomes clear. Since phases only spell out their own complements, the copy in the specifier of the embedded CP is still available

in the structure. If adjunct islands are to be accounted for using cyclic Spell-out, then this architecture predicts that *which book* should be able to move out of the adjunct clause. This rules out this particular architecture. Similar problems will occur in other architecture failures of this type.

Similarly, it cannot be that the higher phase will spell out the entirety of a lower phase when it has completed. Once again there is a similar problem to that in (21), shown below in (23). In this case I have skipped to the relevant part of the structure.

- (23) [C' did [TP Bob ~~did~~ [vP [vP ~~Bob~~ read [VP ~~read~~ a review] [PP after [CP [which book]_i C [TP Mary [vP [~~which book~~]_i [vP ~~Mary~~ recommended [VP ~~recommended~~ [~~which book~~]_i]]]]]]]]]]

In (23) there are two complete phases that are embedded in the structure (the matrix vP and the adjunct CP). They have not been spelled out since neither of them is c-commanded by a higher phase head. The completion of the adjunct CP phase has triggered the Spell-out of the vP inside the adjunct, but there is still a copy of *which book* that has moved up to the adjunct CP. Neither the matrix vP or the adjunct CP will spell out until a phase above it is complete. This will be the matrix CP, but not until it has pulled up a WH phrase. This architecture also fails because it predicts that *which book* should again be able to move to the matrix CP, incorrectly predicting upward movement out of adjuncts.

The next two architecture failures are those in which the complement of a lower phase-head is spelled out by either the completion of the upper phase or the merger of the phase head. In (24) we can see the same point in the derivation we saw above in (23) and (22).

- (24) a. [C' did [TP Bob ~~did~~ [vP [vP Bob read [vP ~~read a review~~] [PP after [CP [which book]; C [TP-Mary [vP-[which book];i-[vP-Mary recommended-[vP recommended-[which book];i]]]]]]]]]]
- b. [C' did [TP Bob ~~did~~ [vP [vP Bob read [vP read a review] [PP after [CP [which book]; C [TP Mary [vP [which book];i [vP Mary recommended-[vP recommended-[which book];i]]]]]]]]]]

In (24a), the introduction of phase heads triggers the Spell-out of lower phase complements. In (24b), only the completion of phases causes the Spell-out of lower phase complements. First, we see in (24a) that the introduction of the matrix C has caused the Spell-out of the complement of two different lower phases, the matrix vP and the adjunct CP. This still leaves *which book* in the specifier of the adjunct CP. Much like (23), this structure permits movement out of adjuncts, and so this architecture is ruled out. In (24b), even less has been spelled out, since the matrix CP has not completed, and so it has not triggered any Spell-out. The only cyclic Spell-out is the complements of the adjunct vP, which was triggered by the completion of the adjunct CP previously in the structure. Once again, we see that *which book* is available for movement into the matrix CP, which rules out this phase architecture.

This leaves us only one possibility without assuming prepositions are phase-heads, the one in which the completion of the higher phase triggers the Spell-out the entire lower phase. However, we see in (25) that this also predicts (20a) to be grammatical.

- (25) [C' did [TP Bob ~~did~~ [vP [vP Bob read [vP read a review] [PP after [CP [which book]; C [TP Mary [vP-[which book];i-[vP-Mary recommended-[vP recommended-[which book];i]]]]]]]]]]

Much like we saw above in (23) and (24b), much of the structure is remains since the matrix CP is not complete, which means the Spell-out of the lower phases (matrix vP and adjunct CP) have not been spelled out. As a result, *which book* has again been left in the derivation and is predicted to be eligible for WH movement to the specifier of CP. Thus this architecture is ruled out as well. This leaves us with only the possibilities in Table 3.

Table 3: All phase architectures eliminated so far, including those eliminated from predicting movement out of adjunct islands (without P phase-heads)

Target: Same or lower phase	Target: Complement or entire phrase	Trigger: Completion of phase or merger of phase-head	P a phase-head?
Phases target themselves	Entire phrase	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No
	Complement of phase-head	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No
Phases target lower phases	Entire phrase	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No
	Complement of phase-head	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No

Given the evidence provided, no combination of architectural features will successfully rule out movement out of adjunct islands without assuming an additional phase head. I will

leave the justification for prepositional phase-hood until section 3.4. For now, I will assume that prepositional heads must be phase-heads, in addition to CP or vP.

Once allowing for prepositional phase-hood, there is one more architecture that is ruled out on the basis of predicting movement out of adjuncts. This is the case if we assume that higher phases trigger the Spell-out of the complement of lower phase heads when the higher phase has been completed. The relevant derivation is shown in (26).

- (26) [C' did [TP Bob ~~did~~ [vP [vP Bob read [VP ~~read~~ a review] [PP after [CP [which book]_i C
[TP ~~Mary~~ [vP [which book]_j [vP ~~Mary~~ recommended [VP ~~recommended~~ [which
book]_i]]]]]]]]]]

In this architecture, we must remember that the number of phases that are relevant has increased. While the matrix CP has not spelled anything out since it is not yet completed, the complement of the adjunct CP has been spelled out due to the completion of the prepositional phase. Given the structure in (26), this architecture again predicts that *which book* should be able to move out of the adjunct CP into the matrix CP, and therefore must be ruled out, leaving the possible architectures in Table 4.

Table 4: All phase architectures eliminated so far, including those eliminated from predicting movement out of adjunct islands (with P phase-heads)

Target: Same or lower phase	Target: Complement or entire phrase	Trigger: Completion of phase or merger of phase-head	P a phase-head?
Phases target themselves	Entire phrase	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No
	Complement of phase-head	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No
Phases target lower phases	Entire phrase	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No
	Complement of phase-head	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No

3.3.3 Type 3 failure: Movement out of adjuncts predicted

Table 4 shows only three remaining possible phasal architectures. Crucially, the goal of these architectures is to create a theory of the derivation in which we can derive a parasitic gap structure with sideward movement. This means that any system that would spell out a WH phrase in the adjunct of a parasitic gap structure before it moves to the matrix clause would be

impossible. The architectural configurations in which this is a problem are ones in which the introduction of the preposition would spell out the CP containing the WH to be moved. To be specific, in (20b), we cannot spell out *which book* before the matrix verb *read* is introduced.

First let us examine what happens if phases spell out their own complement after it is complete. This is the case in (27).

(27) a. *Which book* moves to the specifier of CP, completing CP

$$K = [[\text{which book}]_i \text{ C } [\text{TP Mary } [\text{VP } \text{which book}]_i [\text{VP Mary recommended } [\text{VP recommended } [\text{which book}]_i]]]]]$$

b. Spell-out of TP

$$K = [[\text{which book}]_i \text{ C } [\text{TP Mary T } [\text{VP } \text{which book}]_i [\text{VP Mary recommended } [\text{VP recommended } [\text{which book}]_i]]]]]$$

c. Completion of PP

$$L = [\text{PP after } [[\text{which book}]_i \text{ C } [\text{TP Mary T } [\text{VP } \text{which book}]_i [\text{VP Mary recommended } [\text{VP recommended } [\text{which book}]_i]]]]]$$

d. Spell-out of CP

$$L = [\text{PP after } [[\text{which book}]_i \text{ C } [\text{TP Mary T } [\text{VP } \text{which book}]_i [\text{VP Mary recommended } [\text{VP recommended } [\text{which book}]_i]]]]]$$

In (27a), we can see the adjunct clause has been built up to the parasitic CP. Since the CP phase is complete in (27a), we see the Spell-out of the complement to that phase head (the TP) being spelled out in (27b). In (27c), the PP phase is built up on top of this. Once again, since

the PP phase is now complete, its complement must be spelled out as it is in (27d). This is a problem for deriving parasitic gap constructions. Since the WH phrase cannot have moved out before this (due to the modified version of ActivateSelector forcing the derivation to continue up the spine of the adjunct clause), there is no longer a WH phrase available to be copied due to the Spell-out of the CP complement of *after*.

This also happens if the completion of a higher phase causes the Spell-out of an entire lower phase. Let us look at this in (28). In this case we have skipped to the point of the derivation in (28a), where the adjunct CP has been built up, which has caused the Spell-out of the lower phase (the adjunct vP). In (28b), the PP phase is built up. This results in the Spell-out of the entire lower phase (the adjunct CP) in (28c). Again, we have eliminated all of the copies of *which book* from the derivation before it is able to sideward move to the matrix clause. This architecture does not allow parasitic gap constructions to be built with sideward movement, and so it is ruled out.

(28) a. Completion of CP triggers Spell-out of vP

K = [[which book]_i C [TP Mary T ~~{vP [which book]_i {vP Mary recommended
{vP recommended [which book]_i}}}]~~]]

b. PP completed

L = [PP after [[which book]_i C [TP Mary T ~~{vP [which book]_i {vP Mary
recommended {vP recommended [which book]_i}}}]~~]]

c. Spell-out of CP

$L = [_{PP} \text{ after } [_{CP} \text{ after } [_{TP} \text{ Mary } T [_{VP} \text{ recommended } [_{VP} \text{ recommended } [_{CP} \text{ after } [_{TP} \text{ Mary } T [_{VP} \text{ recommended } [_{VP} \text{ recommended } [_{CP} \text{ after } \dots]]]]]]]]]]]$

One might note that Nunes' derivation would avoid this issue, since the preposition would not merge to the adjunct until much later in the derivation, namely after sideward movement has taken place and right before adjunction. This will not work when prepositions are phase-heads, since my modified version of ActivateSelector forces the derivation to proceed from the adjunct CP to the adjunct prepositional phase. While this would not be true if prepositions were not phase-heads, in the previous two sections I showed that those architectures either fail to produce meaningful cyclic computation or predict the availability of sideward movement.

3.4 Justifying prepositions as a phase-heads

We can see that many of the logical possibilities for the parameter settings will either (i) prevent cyclic computation within the syntax, (ii) predict that movement out of adjunct islands is possible, or (iii) fail to allow for sideward movement derivations of parasitic gaps. This leaves us with only one set of parameter settings in Table 5. These can be summarized in (29).

Table 5: All phase architectures eliminated so far, including those eliminated from predicting movement out of adjunct islands (with P phase-heads)

Target: Same or lower phase	Target: Complement or entire phrase	Trigger: Completion of phase or merger of phase-head	P a phase-head?
Phases target themselves	Entire phrase	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No
	Complement of phase-head	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No
Phases target lower phases	Entire phrase	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No
	Complement of phase-head	Completion of phase	Yes
			No
		Merger of phase-head	Yes
			No

- (29) a. Phases spell out the constituents of a lower phase.
- b. The complement of phase-heads spell out.
- c. The introduction of the phase-head into the derivation is the trigger for the Spell-out of lower phases.
- d. P is a phase-head.

While most of these settings are not terribly controversial, the choice for P to be a phase-head is not standard. I will provide what is taken to be evidence for prepositional phase-hood from other domains. Then I will provide an account of why some prepositions appear not to be phases.

Van Riemsdijk (1978) provides a theory of preposition stranding in which certain types of objects are allowed to move to the edge of the prepositional phrase, which allows extraction of the WH while leaving the prepositions behind. What is interesting about the data from Dutch in (30) is that not all objects are allowed to move to the edge of the prepositional phrase, and so those objects are barred from movement to the edge of the prepositional phrase, and therefore their extraction from the prepositional phrase is impossible.

(30) a. Waar_i heb je [PP *t_i* op *t_i*] gerekend?

Where have you on counted

“What did you count on?”

b. *Wie_i heb je [PP *t_i* op *t_i*] gerekend

Who have you on counted

“Who did you count on?”

Van Riemsdijk’s account is that R-pronouns⁶ such as *waar* in (30a) are selected for this type of movement. The fact that there is movement to the edge of the PP contributes to the

⁶ In Van Riemsdijk’s theory, R pronouns are lexically marked as +R since they contain an /r/ phoneme that is visible to the syntax.

argument that PPs are phases (though this cannot be the whole reason, since TPs move subjects to their edge but are not phases). The important part of this evidence for my theory is the fact that the WH phrase is unavailable when it is unable to move to the edge of the prepositional phrase, as in (30b). The pattern of being an island is the result of Spell-out and avoiding it through movement is one of the hallmarks of phase-hood. Abels (2003) accounts for the data in (30) as well as for data from languages that disallow preposition stranding more broadly by stating that in these languages, PPs are phases. Additionally, he argues that the complement of a phase-head cannot move to the edge of that phase. He argues that the pattern in (30) reflects the fact that R-pronouns are in fact in the specifier of the complement of P. This allows R-pronouns to move to the edge of the prepositional phrase for later extraction. Other objects are simply the complement of the preposition and therefore cannot be extracted.

In addition to the arguments for phasehood from movement phenomena, we can see that prepositional phases are also potentially the cause of a lack of agreement in certain constructions. Bobaljik (2008), Holmberg and Hróarsdóttir (2003), Sigurðsson (1993, 1996) all describe the lack of agreement that dative nominals in relation to verbs. Two examples from Icelandic are given in (31) and (32).

- (31) Morgum studentum líki/*líka verkið
 many student.pl.Dat like.3sg/*3pl job.the.Nom
 “Many students like the job.”

(Harley 1995)

(32) Strákunum leiddist/*leiddust

Boy.the.pl.Dat were.bored.3sg/*3pl

“The boys were bored.”

(Sigurðsson 1996)

In (31) we can see that the dative argument is skipped over for agreement, and the nominative argument is agreed with. The sentence in (32) shows that this cannot be accounted for with a preference for nominative targets, since there is no target other than the dative one, but the predicate still fails to agree with it. If this lack of agreement is the result of unavailability due to the cyclic Spell-out of the DPs inside of prepositional phrases, then the lack of agreement would be unsurprising.

In order to entertain a theory in which prepositions are phase-heads, we need an account of why some prepositions seems to allow movement through their edge while others do not. For example, we know that adjuncts can be islands to movement as they are in (20a), but movement is fine in a sentence like (33).

(33) What did John leave the party after?

Munn (2001) gives an account of the category restrictions for parasitic gap constructions. In his theory, temporal adjuncts would move a silent temporal operator to a high position within the relative clause. He gives the structure in (34), though states that it is not obvious which of the two operators (the temporal operator and the parasitic gap operator) ends up in which high position.

(34) $[_{PP} Op_{pg} \text{ before } [_{CP} Op_{temp} [_{IP} \dots t_{pg} \dots t_{temp}]]]]$

If it is the case that prepositions move an operator into a local position, this movement could be leveraged as an escape hatch in examples like (33). If we think that *after* requires a temporal operator that describes the time that the leaving was after (Geis 1970, Larson 1990), in the case of (33), that would be the object of the preposition, in this case, *what*. This structure is given in (35). Once at the PP edge, it would be able to move out.

(35) What_i did John leave the party [what_i after what_i]?

However, in (20a) (repeated here as (36a), the requirement for the preposition cannot be satisfied by *what book*.

(36) a. *[Which book]_i did Bob read a review after Mary recommended *t_i*?

b. *[Which book]_i did Bob read *t_i* after Mary recommended *pg_i*?

Instead, the temporal operator in the parasitic domain is moved into that position, blocking movement out. As a result, the highest *what book* can move in the adjunct is into the edge of the adjunct CP phase. This is not high enough to escape Spell-out without sideward movement.

3.5 Applying the phase architecture

Given the architectural necessities from the last section and a justification for prepositional phase-hood, it must be shown that this theory can account for the data in (36). Generally speaking, the remaining architecture must account for adjunct islands, as well as allow for the derivation of parasitic gaps. It must also be able to account for the islands found within parasitic gap constructions.

3.5.1 Deriving adjunct islands and parasitic gaps

Below in (37), we can see that the parameter settings in (30) correctly predict the ungrammaticality of (36a).

(37) *[Which book]_i did Bob [_{VP} read a review] after Mary recommended *t_i*?

a. Construction of PP

K = [_{PP} after [_{CP} [which book]_i] C [~~TP~~ ~~Mary~~ [~~VP~~ [~~which book~~]_i] [~~VP~~ ~~Mary~~ recommended [~~VP~~ recommended [which book]_i]]]]]]

b. Matrix vP built up

K = [_{PP} after [_{CP} [which book]_i] C [~~TP~~ ~~Mary~~ [~~VP~~ [~~which book~~]_i] [~~VP~~ ~~Mary~~ recommended [~~VP~~ recommended [which book]_i]]]]]]

L = [_{VP} Bob read [_{VP} read a review]]

c. PP adjoined to matrix vP

M = [_{VP} [_{VP} Bob read [_{VP} read a review]] [_{PP} after [_{CP} [which book]_i] C [~~TP~~ ~~Mary~~ [~~VP~~ [~~which book~~]_i] [~~VP~~ ~~Mary~~ recommended [~~VP~~ recommended [which book]_i]]]]]]]]

d. TP built

N = [_{TP} Bob [_{VP} [_{VP} ~~Bob~~ read [_{VP} read a review]] [_{PP} after [_{CP} [which book]_i] C [~~TP~~ ~~Mary~~ [~~VP~~ [~~which book~~]_i] [~~VP~~ ~~Mary~~ recommended [~~VP~~ recommended [which book]_i]]]]]]]]

e. Matrix C merged

$$M = [C' C [_{TP} \text{Bob} [_{VP} [_{VP} \text{Bob} \text{read} [_{VP} \text{read a review}]]] [_{PP} \text{after} [_{CP} [\text{which book}]_i C [_{TP} \text{Mary} [_{VP} [\text{which book}]_i [_{VP} \text{Mary recommended} [_{VP} \text{recommended} [\text{which book}]_i]]]]]]]]]$$

f. Spell-out of adjunct CP and matrix VP

$$M = [C' C [_{TP} \text{Bob} [_{VP} [_{VP} \text{Bob} \text{read} [_{VP} \text{read a review}]]] [_{PP} \text{after} [_{CP} [\text{which book}]_i C [_{TP} \text{Mary} [_{VP} [\text{which book}]_i [_{VP} \text{Mary recommended} [_{VP} \text{recommended} [\text{which book}]_i]]]]]]]]]$$

In (37a) we can see that the adjunct clause has been built up. The introduction of the preposition has previously caused the Spell-out of the complement of the adjunct C (the adjunct TP). In (37b), the matrix vP is built up. Since all of its arguments are filled, there can be no sideward movement into a thematic position, and movement into the vP edge is barred since such a movement depends on Agree, and Agree cannot see into the adjunct clause since it does not c-command it. In (37c) the adjunct is merged to the matrix vP, and in (37d) the rest of the matrix TP is constructed. In (37e), the matrix C is merged into the derivation. This triggers the Spell-out of all the complements of lower phase heads in (37f). This include the complement of the matrix vP (the matrix VP), and the complement of the PP (the adjunct CP). At this point in the derivation, we know we have accounted for the adjunct island, since all available copies of *which book* are removed from the derivation. This will always be the case since it cannot move to the edge of the PP (since it is not a temporal operator for *after*), and it cannot move into the matrix clause before this point.

Next, the parasitic gap constructions like (36b) must be able to be derived using the same assumptions about the phase architecture. This derivation is provided in (38).

(38) [Which book]_i did Bob [_{VP} read *t_i*] after Mary recommended *pg_i*?

a. Adjunct PP built up

$K = [PP \text{ after } [CP [which \text{ book}]_i C [TP \text{ Mary } [_{VP} [which \text{ book}]_i [_{VP} \text{ Mary recommended } [_{VP} \text{ recommended } [which \text{ book}]_i]]]]]]$

b. Introduction of the matrix verb *read*

$K = [PP \text{ after } [CP [which \text{ book}]_i C [TP \text{ Mary } [_{VP} [which \text{ book}]_i [_{VP} \text{ Mary recommended } [_{VP} \text{ recommended } [which \text{ book}]_i]]]]]]$

$L = \text{read}$

c. Sideward movement of *which book*

$K = [PP \text{ after } [CP [which \text{ book}]_i C [TP \text{ Mary } [_{VP} [which \text{ book}]_i [_{VP} \text{ Mary recommended } [_{VP} \text{ recommended } [which \text{ book}]_i]]]]]]$

$M = [_{VP} \text{ read } [which \text{ book}]_i]$

d. Matrix vP built up

$K = [PP \text{ after } [CP [which \text{ book}]_i C [TP \text{ Mary } [_{VP} [which \text{ book}]_i [_{VP} \text{ Mary recommended } [_{VP} \text{ recommended } [which \text{ book}]_i]]]]]]$

$N = [_{VP} \text{ Bob read } [_{VP} \text{ read } [which \text{ book}]_i]$

e. WH movement of *which book* to the edge of the vP

K = [_{PP} after [_{CP} ~~{which book}_i~~] C [_{TP} ~~Mary~~ [_{VP} ~~{which book}_i~~] [_{VP} ~~Mary recommended~~ [_{VP} ~~recommended~~ [_{VP} ~~{which book}_i~~]]]]]

O = [_{VP} [which book]_i] [_{VP} Bob read [_{VP} read [which book]_i]]]

f. PP adjoined to vP

P = [_{VP} [_{VP} [which book]_i] [_{VP} Bob read [_{VP} read [which book]_i]]] [_{PP} after [_{CP} ~~{which book}_i~~] C [_{TP} ~~Mary~~ [_{VP} ~~{which book}_i~~] [_{VP} ~~Mary recommended~~ [_{VP} ~~recommended~~ [_{VP} ~~{which book}_i~~]]]]]]]

g. TP built up

Q = [_{TP} Bob did [_{VP} [_{VP} [which book]_i] [_{VP} ~~Bob~~ read [_{VP} read [which book]_i]]] [_{PP} after [_{CP} ~~{which book}_i~~] C [_{TP} ~~Mary~~ [_{VP} ~~{which book}_i~~] [_{VP} ~~Mary recommended~~ [_{VP} ~~recommended~~ [_{VP} ~~{which book}_i~~]]]]]]]

h. C merged into structure

R = [_{C'} did [_{TP} Bob ~~did~~ [_{VP} [_{VP} [which book]_i] [_{VP} ~~Bob~~ read [_{VP} read [which book]_i]]] [_{PP} after [_{CP} ~~{which book}_i~~] C [_{TP} ~~Mary~~ [_{VP} ~~{which book}_i~~] [_{VP} ~~Mary recommended~~ [_{VP} ~~recommended~~ [_{VP} ~~{which book}_i~~]]]]]]]

i. Spell-out of matrix VP and adjunct CP

S = [_{C'} did [_{TP} Bob ~~did~~ [_{VP} [_{VP} [which book]_i] [_{VP} ~~Bob~~ read [_{VP} read [which book]_i]]] [_{PP} after [_{CP} ~~{which book}_i~~] C [_{TP} ~~Mary~~ [_{VP} ~~{which book}_i~~] [_{VP} ~~Mary recommended~~ [_{VP} ~~recommended~~ [_{VP} ~~{which book}_i~~]]]]]]]

j. Movement of *which book* to specifier of CP

T = [CP [which book]_i did [TP Bob did [VP [VP [which book]_i [VP Bob read [VP read [which book]_i]]] [PP after [CP [which book]_i C [TP Mary [VP [which book]_i [VP Mary recommended [VP recommended [which book]_i]]]]]]]]]]]

In (38a) we see a build up of the adjunct clause. Since the *after* has been merged, that triggers the Spell-out of the complements of the phase below, namely the complement of the adjunct C. Then we are able to introduce the matrix verb in (38b). Since *which book* has not been spelled out yet, it is available for sideward movement out of the adjunct, into the complement of the matrix verb in (38c). Next, the rest of the matrix vP is built up in (38d). While the introduction of the v-head might normally induce the Spell-out of lower phases, there are not phases embedded in the structure that includes v, and so nothing is spelled out. In (38e), *which book* is moved to the edge of the vP phase. It is copied from inside of the matrix vP since A' movement is contingent upon Agree, and Agree can only look inside of the structure that the v-head is a part of. In (38f) the adjunct is adjoined to the vP and the rest of the TP is build up in (38g). Next, in (38h) the matrix C is introduced into the derivation. Since C is a phase head, its merger triggers the Spell-out of lower phases, in this case the adjunct PP and the matrix vP. Both of their complements are spelled out in (38i), which leaves the copy of *which book* that is adjoined to the matrix vP. Lastly, *which book* is moved from that position into SpecCP in (38j), finishing the derivation.

3.5.2 Accounting for islands within the parasitic domain

In section 3.2, I provided a partial account of island effects within parasitic gaps. Crucially, I tied movement to a rule that spelled out the object that was copied so that only the most recent copy of an object was available for further computation and reducing the amount of

structure in the derivational workspace. This works in concert with a modified version of Haida and Repp's (2012) ActivateSelector to make derivations in which copies move into an adjunct clause fail to converge if adjunct islands are properly accounted for (as done in section 3). In order to show that these islands are accounted for, I will restate the three types of islands within the parasitic domain that I presented in Chapter 2 below in (39).

(39) a. Adjunct island

*[Which book]_i did Bob [_{vP} read *t_i*] after Mary [[_{vP} left the room] without recommending *pg_i*]?

b. WH island

*[Which book]_i did Bob [_{vP} read *t_i*] after [_{CP} Mary asked [_{CP} when Suzy recommended *pg_i*]]?

c. Subject island

*[Which author]_i did Bob [_{vP} criticize *t_i*] after [_{CP} [_{DP} pictures of *pg_i*] upset the fans]?

First, let us see how the ungrammaticality of (39a) is accounted for by examining the derivation in (40).

(40) * [Which book]_i did Bob [_{vP} read *t_i*] after Mary [[_{vP} left the room] without recommending *pg_i*]?

a. Adjunct island PP constructed

K = [PP without [TP Mary [vP [which book]_i [vP ~~Mary~~ recommending [vP recommending [which book]_i]]]]]

b. Adjunct vP constructed

K = [PP without [TP Mary [vP [which book]_i [vP ~~Mary~~ recommending [vP recommending [which book]_i]]]]]

L = [vP Mary left [vP ~~left~~ the room]]

c. Adjunct island PP merged to adjunct vP

M = [vP [vP Mary left [vP ~~left~~ the room]] [PP without [TP Mary [vP [which book]_i [vP ~~Mary~~ recommending [vP ~~recommending~~ [which book]_i]]]]]]]

d. Adjunct C merged

N = [C' C [TP Mary [vP [vP ~~Mary~~ left [vP ~~left~~ the room]] [PP without [TP Mary [vP [which book]_i [vP ~~Mary~~ recommending [vP ~~recommending~~ [which book]_i]]]]]]]]]

e. Spell-out of adjunct island TP

N = [C' C [TP Mary [vP [vP ~~Mary~~ left [vP left the room]] [PP without [TP ~~Mary~~ [vP [which book]_i [vP ~~Mary~~ recommending [vP ~~recommending~~ [which book]_i]]]]]]]]]

In (40a), we see the derivation has built up the entire adjunct containing the parasitic gap. Since this clause is a control clause, I will follow Boeckx, Hornstein and Nunes' (2010) theory of

control using sideward movement in which there is no complementizer in this adjunct. The introduction of the P head has spelled out the complement of the vP phase. In (40b), we see phase that selects the adjunct K being built up, since ActivateSelector forces the derivation to move on to this phase. The thematic positions in the vP are filled up by *Mary* moving to the matrix clause and *the room* merging into the object position from the numeration. In (40c) the adjunct merged to the vP in L to make the syntactic object M. In (40d), the rest of the adjunct is built up until the introduction of C. This triggers the Spell-out of the complements of lower phases in (40e).⁷ This step is crucial, since the last copy of *which book* has been spelled out, making, the derivation will fail to check its WH feature, and the derivation will crash. WH islands suffer a similar fate. This is shown in the derivation in (41).

(41) * $[\text{Which book}]_i$ did Bob $[\text{vP read } t_i]$ after $[\text{CP Mary asked } [\text{CP when Suzy recommended } pg_i]]?$

a. Adjunct embedded vP constructed

$$K = [\text{vP } [\text{vP } [\text{which book}]_i [\text{vP Suzy recommended } [\text{vP } \text{recommended } [\text{which book}]_i]] [\text{PP when}]_j]]$$

b. Adjunct embedded C merged

$$L = [\text{C'} C [\text{TP Suzy } [\text{vP } [\text{vP } [\text{which book}]_i [\text{vP } \text{Suzy recommended } [\text{vP } \text{recommended } [\text{which book}]_i]] [\text{PP when}]_j]]]]]$$

⁷ Building this structure made an assumption that should be addressed. There was no CP in the adjunct for *which book* to move to. This is done in order to accommodate Boeckx, Hornstein and Nunes' (2010) movement theory of control. However, this would not have saved the derivation, since a copy in the CP would still have been spelled out in (40e) as the complement of the phase head P.

c. WH movement of *when*

$$M = [_{CP} [_{PP} \text{when}_j] C [_{TP} \text{Suzy } [_{VP} [_{VP} [\text{which book}]_i [_{VP} \text{Suzy recommended} \\ [_{VP} \text{recommended } [\text{which book}]_i]] [_{PP} \text{when}_j]]]]]$$

d. Adjunct built up to the merger of *v*

$$N = [_{v'} \text{asked } [_{VP} \text{asked } [_{CP} [_{PP} \text{when}_j] C [_{TP} \text{Suzy } [_{VP} [_{VP} [\text{which book}]_i [_{VP} \\ \text{Suzy recommended } [_{VP} \text{recommended } [\text{which book}]_i]] [_{PP} \text{when}_j]]]]]]]$$

e. Spell-out of embedded TP

$$N = [_{v'} \text{asked } [_{VP} \text{asked } [_{CP} [_{PP} \text{when}_j] C [_{TP} \text{Suzy } [_{VP} [_{VP} [\text{which book}]_i [_{VP} \\ \text{Suzy recommended } [_{VP} \text{recommended } [\text{which book}]_i]] [_{PP} \text{when}_j]]]]]]]$$

In (41a), the embedded vP of the adjunct is built up, with *which book* moving to the edge of the vP and the *when* merged to the vP above that. In (41b) the embedded CP of the adjunct is built and C is introduced spelling out the VP below (in this case vacuously). In (41c) *when* is selected for movement to the edge of the CP. Then in (41d), the higher clause is built up until the introduction of *v*. This triggers the Spell-out of the complement of the CP, which is the embedded TP containing the only remaining copy of *which book*.

3.5.2.1 Subject islands and Spell-out

Lastly, subject islands need to be accounted for. These structures are more complicated, since it is not clear that they can be the result of the same cyclic Spell-out that accounts for WH islands and adjunct islands. Additionally, subject islands can be thought of as the same phenomenon as the ‘freezing’ phenomenon discussed in 3.2.1., since the restriction on extraction

is only true for the moved copy. We can see in (42) and (43) that the WH phrases embedded within the DPs in (42) can be extracted, they cannot be extracted from the same structures in (43) (from Narita 2011).

- (42) a. [Which article]_i did everyone know [that John wrote *t_i*]?
 b. [Which person]_i did Bob sell [pictures of *t_i*]
- (43) a. *[Which article]_i was [that John wrote *t_i*] known to everyone?
 b. *[Which person]_i were [pictures of *t_i*] on sale?

The pattern we see in (42) and (43) is the same pattern we see in the restriction against extracting from extraposed objects. This predicts that it is not subject-hood, but movement that is the cause of freezing. This predicts that unmoved subjects should not be islands to extraction. The data in (44) from Stephanov (2007) shows that this is the case in languages such as Japanese, in which subjects have been argued to not move outside of the vP (Kuroda 1988, Fukui 1986,1988, 1995, Fukui and Speas 1986, Lasnik and Saito 1992).

- (44) a. Clefting

[*Op_j* [John-ga *t_j* okane-o karita koto] -ga Bill-o

John-Nom money-acc borrowed NOMINAL -Nom Bill-Acc

kizutuketa no] -wa *Mary-kara_j* datta.

hurt NOMINAL -Top Mary-from CPL.past

“It was *from Mary_j* [*Op_j* that [that John borrowed money *t_j*] hurt Bill]”

- b. Scrambling

sono hon-o_i John-ga [[*Mary-ga t_i* katta koto] -ga Bill-o kizutuketa

that book-Acc John-Nom Mary-Nom bought NOMINAL -Nom Bill-Acc hurt
to] omotteru

that think

“*That book_i*, John thinks [that [that Mary bought *t_i*] hurt Bill]”

In (44a), we can see that *Mary* has moved out of the subject of the embedded clause into a position for clefts. The same is true in (44b), where *that book* has moved to a leftward scrambling position, despite being extracted out of the subject of the embedded clause. This data shows two crucial points. The first is that if subjects are not moved, syntactic objects can be extracted from them. In addition, Narita (2011) points out that if we assume that these subjects are in the specifier of vP, then this is problematic for Nunes and Uriagereka’s (2000) account of these subject islands, since it crucially relies on them being in specifiers.

I will discuss three ways that these islands that result from movement might be analyzed. First, Narita (2011) proposes that portions of the moved object must spell out in order to be moved, and this results in their unavailability to later computation. Therefore, his theory would account for all freezing effects through a Spell-out operation that be triggered by movement. This theory is appealing in its use of a motivated operation (Spell-out), and its use of a trigger that can be motivated without a knowledge of the resulting structure. Narita’s theory suggests that the entire moved structure is spelled out and reduced to a single syntactic atom that can then be merged into the structure. This is problematic in the larger theory being presented in this dissertation, since this Spell-out operation would have a new target for Spell-out (all but the head that remains).

Another alternative is a separate ‘freezing’ mechanism that prevents extraction without Spell-out. This introduces an entirely new mechanism into the computational system, and doing so should be avoided, especially since such an operation has no apparent conceptual motivation.

Another possibility is that movement triggers a normal Spell-out within the moved object. This can happen one of two ways. The first way is to assume there are already phase-heads within the moved object that can trigger for Spell-out much like the introduction of phase-heads triggers the Spell-out of lower phases. This cannot be correct, since it predicts the wrong pattern for (42) and (43). If there can be movement out of an unmoved object, as in (42), then there must be an escape hatch for whatever phase is within the moved object. This would result in a structure like (45), where XP is the phase within the DP that provides an escape hatch.

(45) [DP \emptyset [XP[which person]_i [pictures [of [~~which person~~]_i]]]]

As we can see in (45), *which person* is now outside of the phase that would be spelled out by movement. However, such an escape hatch would also allow extraction after movement, predicting the grammaticality of the sentences in (43). If there is no escape hatch, then we should expect that WH extraction should be impossible in (42). We can see this in the derivational step in (46).

(46) [_v v [VP sell [DP \emptyset [XP X [~~pictures~~ [~~of~~ [~~which person~~]_i]]]]]]

Upon the introduction of *v* in (46), the phase within the DP should spell-out its complement. If *which person* has not moved to the edge of that DP, it will not be able to move to adjoin to *vP*, and later to the specifier of CP to check its WH feature.

A second way that movement could trigger Spell-out within the moved object is to treat the target of movement as though it were marked as a phase. If a DP is targeted by movement, then the complement of that DP is targeted by Spell-out, much like if it were a phase. While this would capture the data at hand in much the same way that Narita's theory would, it expands the possible targets of Spell-out.

For the purposes of this dissertation, I will assume that movement will treat the target of movement much like a phase, and trigger the Spell-out of the complement of the head of the moved phrase. Many of the options above can account for the data at hand, but this version best preserves the target of the Spell-out operation.

Now that I have presented a theory of the islandhood of moved objects, including subjects, I will show how subject islands can be accounted for within the parasitic domain. The relevant portion of the derivation is below in (47).

(47) * $[\text{Which author}]_i$ did Bob $[\text{VP criticize } t_i]$ after $[\text{CP } [\text{DP pictures of } pg_i]]$ upset the fans]?

a. Adjunct vP built up

$K = [\text{VP } [\text{DP } D \text{ pictures of } [\text{which author}]_i]] \text{ upset the fans}$

b. Adjunct T merged

$L = [\text{T}' \text{ T } [\text{VP } [\text{DP } D \text{ pictures of } [\text{which author}]_i]] \text{ upset the fans}]$

c. Movement of *pictures of which author* and movement Spell-out

M = [_{TP} [_{DP} D ~~pictures of [which author]_i]~~ T [_{VP} [~~DP~~ D ~~pictures of [which author]_i]~~-upset the fans]]

d. Merger of adjunct C

N = [C' C [_{TP} [_{DP} D ~~pictures of [which author]_i]~~ T [_{VP} [~~DP~~ D ~~pictures of [which author]_i]~~-upset the fans]]]

In (47a), the parasitic vP is constructed. Since there are no empty thematic positions, there is no sideward movement available here. In (47b), the adjunct T is introduced into the structure, which requires the movement of the subject. In (47c), *pictures of which author* moves to the specifier of TP. This causes the Spell-out of the whole lower copy. In addition, this causes the complement of D to spell out, which is the NP *pictures of which author*. This eliminates all of the copies of *which author* from the derivation. As a result, in (47d), when *which author* needs to move to the edge of the embedded CP, it is unavailable, and therefore will be unable to move out at all to the matrix clause. This allows us to account for subject islands within the parasitic domain.

3.6 Conclusion

This chapter has introduced a sideward movement theory for parasitic gaps that accounts for adjunct islands and islands within parasitic gap constructions without appeal to a special rule that only target islands for linearization purposes. Such special rules are problematic in their specialized nature and also their reliance upon knowing the linearization requirements of PF. I also introduced a modified version of Haida and Repp's (2012) ActivateSelector that forces the derivation to proceed to a phase that selects for the phase just produced. This captures an

intuition expressed by many previously (Chomsky 2000, 2001, 2004; Nunes 2012; Haida and Repp 2012).

Also, since A' movement to crucially relies on Agree, we can restrict such positions from recruiting sideward movement, since it involves search outside of the structure. While it seems it has been argued that this must be possible for Merge (specifically Internal Merge), this cannot be true of Agree. Next, I have shown that the architecture of the derivation can provide many of the limitations we need on sideward movement, since the crucial derivations are ruled out by Spell-out, while still allowing for sideward movement where we see it. In addition to these restrictions, this theory allows us to describe adjunct island effects without proposing rules that specifically target adjunct for inaccessibility. This is done with simple PIC effects caused by the same process that spells out other phases, making them unavailable for further computation. This is far preferable to stipulating special processes that apply to adjuncts.

In the next chapter, I will address the other restrictions on parasitic gaps that are no longer accounted for straightforwardly in a theory of parasitic gaps where the parasitic copy is moved to the edge of the adjunct. These include the first four constraints of (48), repeated from Chapter 2. In addition, I will address the restrictions on the type of object that can be moved out of parasitic domains.

- (48)
- a. Parasitic gaps are licensed at S-Structure.
 - b. Parasitic gaps cannot be c-commanded by the “real” gap.
 - c. Parasitic gaps cannot be licensed by A-chains.

d. Parasitic gaps are “selectively sensitive to island; that is, although they typically occur within (Condition on Extraction Domain) islands, they cannot be separated from their licensor by more than one island.

e. Parasitic gaps can only be NPs.

f. Parasitic gaps cannot be licensed by nonreferential NPs.

4 Accounting for parasitic gap restrictions

4.1 Introduction

In the last chapter, I laid out a theory of phases and sideward movement that accounts for parasitic gaps, as well as CED islands within and outside of them. This theory fills some empirical gaps in Nunes' (2004, 2012) and Haida and Repp's (2012) theories of parasitic gaps and sideward movement. Parasitic gap constructions are complex, and come with many descriptive restrictions that must be accounted for. Nunes (2004) provides an account of many of these restrictions within his sideward movement theory. However, the deviations from Nunes' theory that I have made mean that some of his accounts of the restrictions on parasitic gaps must be altered. This chapter is designed to lay out the assumptions from Nunes and Haida and Repp that continue to be used in this account, as well as describe the changes that must be made in light of the theory developed in Chapter 3. I will show that given these changes, we can indeed still account for the data Nunes (2004) describes. In addition, I will show that this account can be extended to ATB constructions as well while accounting for the differences in restrictions between ATB and parasitic gap constructions in a way that follows naturally from the general syntactic operations that govern both. Due to how the restrictions are built in to the theory of parasitic gap constructions in Chapter 3 in conjunction with the phase architecture that was developed, the lack of restrictions in ATB constructions follows quite naturally.

4.2 Structural restrictions on parasitic gaps

Nunes (2004) accounts for several restrictions on parasitic gap constructions, which I will repeat here in (1).

- (1) a. Parasitic gaps are licensed at S-Structure.
- b. Parasitic gaps cannot be c-commanded by the “real” gap.
- c. Parasitic gaps cannot be licensed by A-chains.
- d. Parasitic gaps are “selectively sensitive to islands.” That is, although they typically occur within CED islands, they cannot be separated from their licenser by more than one island.
- e. Parasitic gaps can only be NPs.
- f. Parasitic gaps cannot be licensed by nonreferential NPs.

Property (1d) has been accounted for by my theory of parasitic gaps in the previous chapter. This leaves five other restrictions that have yet to be directly discussed. This section of the chapter will be focused on the (1a-c). In Nunes’ (2004) sideward movement theory of parasitic gaps, these restrictions are accounted for through a series of conditions on the formation of chains. In Nunes’ theory, chains are not formed from a movement history, since the narrow syntax has no such record. Instead, identical syntactic objects are made into chains at PF through an operation Nunes calls Form Chain. I will be assuming that Nunes’ theory of chain formation is correct. This operation and its conditions are repeated in (2-4).

(2) Conditions on Form Chain

Two constituents α and β can form a nontrivial chain $CH = (\alpha, \beta)$ if

a. α is nondistinct from β ;

b. α c-commands β ;

c. there is at least one feature F of α such that F enters into a checking relation with a sublabel of the head of the projection with which α merges and for any such feature of α , the corresponding feature F of β is accessible to the computational system and;

d. there is no constituent γ such that γ has a feature F' that is of the same type as the feature F of α , and γ is closer to α than β is.

(3) Sublabel

Σ is a sublabel of the head H iff σ is a feature of H or a feature of some element adjoined to H .

(4) Closeness

γ is closer to α than β is iff (a) α c-commands γ and γ c-commands β , and (b) γ is not in the same minimal domain as α or β

Nunes uses these conditions on Form Chain to restrict the possible chains that can be made in parasitic gap constructions. In doing so, he captures the restrictions in (1a-c). However, many of his assumptions about the structure of parasitic gap constructions depended on a lack of

movement within the parasitic domain. In the last chapter, I demonstrated that this leaves empirical and conceptual gaps. In order to address these gaps, my theory necessitates movement to the edge of the parasitic domain in order to properly derive parasitic gaps. This movement is motivated by the WH features on the moved object (though they cannot be checked in the parasitic domain). This goes against some of the specifics of Nunes' explanations for the restrictions in (1a-c), and so this section will address these restrictions each in turn. First, I will discuss Nunes' account of the restriction. Then I will show whether this account holds in a theory in which there is movement in the parasitic domain, such as the one this dissertation proposes. Lastly, if necessary, I will show how the current theory still accounts for the same set of restrictions.

4.2.1 S-Structure licensing of parasitic gaps

Parasitic gaps are described in the literature as being licensed at S-structure (Chomsky 1982, Engdahl 1983). The ungrammaticality of (5a) below shows that parasitic gaps cannot be licensed by some higher position that *which paper* will covertly move to that c-commands the parasitic gap, such as in (5b).

- (5) a. *Who filed [which paper]_i after John read *pg*_i?
 b. Who [[which paper]_i [filed [which paper]_i] after John read [which paper]_i]?

Nunes (2004) provides two accounts for a sentence like (5), one for each of two different structures. Nunes' first account relates to the structure in (6) (adapted from Nunes 2004), where *which paper* doesn't move high enough to c-command the parasitic gap.

- (6) [who [[filed [which paper]¹][after John read [which paper]²]]]

This structure is accounted for quite easily by Nunes in a way that does not need to be altered. In this case, the two copies of *which paper* cannot form a chain, since there is no c-command relationship between them. This means that there are no chains to be reduced, and both copies are left in the PF output. This crashes the derivation, since both copies of *which paper* are equivalent, which is a violation of the irreflexivity requirement on linearization. In this case, Nunes' account will work for understanding the S-Structure licensing restriction on parasitic gap constructions.

However, Nunes (2004) also addresses a potential structure that would be problematic for his theory, where *which paper* has moved in the matrix clause to a higher position to check case (such as the specifier of AgrO) that does in fact c-command the parasitic gap. This structure is shown in (7).

- (7) [who [AgrOP [which paper]¹ [[filed [which paper]²][after John [AgrOP [which paper]³ read [which paper]⁴]]]]]

In order to rule this out, Nunes relies on the condition on Form Chain used in (2c). In this case, since *which paper* has moved to a higher position in the matrix clause for case, then it must form chains with objects that have unchecked case features. Copy 3 of *which paper*, having moved to the same position for the same reasons must also have its case features checked. This means that it cannot form a chain with copy 1, as the checked status of its case feature renders that case feature inaccessible to the computational system by Nunes' definition in (2c). As a result, no chain can be made between copies 1 and 3, and so chain reduction cannot delete copy 3, leaving both copy 1 and 3 pronounced at PF, which results in a crash.

The crucial device in Nunes' theory is that copy 3 can never be targeted by Form Chain when the c-commanding element is in a case position, since its case feature is inaccessible. This still holds if the assumption of covert movement for case is extended to my proposed theory where the WH phrase moves within the parasitic domain. I do not believe that this is the case, but for the sake of discussion, I will show that the analysis Nunes give will extend to the theory presented here. This is because the highest copy in the parasitic domain will also be a case intervener, and thus it will have the same properties. The structure for this is given in (8).

- (8) [who [_{AgroP} [which paper]¹ [[filed [which paper]²][after [which paper]³ John [_{AgroP} [which paper]⁴ read [which paper]⁵]]]]]

In this case, copy 3 cannot form a chain with copy 1, which is the only copy that c-commands copy 3. This means that both copy 1 and copy 3 will be pronounced (copy 4 and 5 are in a chain with copy 3 because 3 forms a chain with 4 via WH features, and 4 forms a chain with 5 via case features). Both copies being pronounced at PF is once again problematic.

Under such an account, another version of the restriction against S-structure licensing becomes relevant. If quantifier raising can move an object high enough in the structure, why can sentences like (9a) license parasitic gaps, given a LF structure like (9b)?

- (9) a. *John reviewed every paper after Steve read?
 b. [[John [reviewed [every paper]¹ after Steve read [every paper]²] [every paper]³]?]

In (9b), we can see that *every paper* has moved to scope over the entire matrix TP. In doing so, copy³ c-commands both the matrix copy and the parasitic copy. However, this

structure is not derivable. *Every paper* can not escape the adjunct clause because it does not move high enough to escape the Spell-out of the adjunct CP. Therefore, it cannot move into the matrix clause to create copy², and thus cannot move up to the position copy³ occupies.

Therefore, in the case of the S-Structure licensing requirement of parasitic gap constructions, Nunes' account extends to the present theory because the case marking that renders the highest copy invisible to the computational system should remain on copies made from that copy, ensuring the same problem propagates up the parasitic chain.

4.2.2 C-command condition on parasitic gaps

The next restriction I will consider is from (1b), states that the parasitic gap cannot be c-commanded by the true gap (Taraldsen 1981, Engdahl 1983). This is shown in (10), where the WH phrase is the subject of the matrix clause, and the subject position c-commands into the parasitic domain. Under Nunes' (2004) account, this would have the structure shown in (11).

(10) *Which man t_i filed the paper after John met pg_i ?

(11) [_{CP} [which man]¹ C [_{TP} [which man]² [_{VP} [_{VP} [which man]³ [_{VP} filed the paper]]][_{PP} after John met [which man]⁴]]]]

In this structure, copy 1 can form a chain with 2 because copy 2 has unchecked WH features, and copy 1 is in a WH position. Copy 2, in turn, can form a chain with copy 3, since copy 2 is in a case position and copy 3 has its unchecked case features. However, copy 4 cannot form a chain with any of these. It cannot form a chain with copy 3, since there is no c-command relationship between them to allow a chain to form. Copy 4 also cannot form a chain with copy 2. Because copy 2 is in a case position, it only attempts to form a chain with a copy that has not

yet had its case feature checked. While copy 4 does not have its case feature checked, there is an intervener with a case feature (in this structure, *John*). Therefore, the condition on Form Chain in (2d) disallows the chain between copy 2 and copy 4 to form. Lastly, copy 1 cannot form a chain with copy 4. Since copy 1 it is in a position that checks its WH feature, and copy 2 serves as an intervener for such a chain due to the unchecked WH feature on copy 2.

In my theory of parasitic gaps, the same principles of Form Chain apply, but the final result is achieved through another path. The structure I assume for (10) is given in (12).

- (12) [CP [which man]¹ C [TP [which man]² [vP [vP [which man]³ [vP filed the paper]]][PP after [which man]⁴ John [vP [which man]⁵ [vP met [which man]⁶]]]]]]

In (12), copies 1, 2, and 3 form a chain (1 and 2 through WH features, 2 and 3 through case features). Copies 4, 5, and 6 also form a chain through WH features. The question is why copy 4 cannot form a chain with any higher elements for later chain reduction. Copy 3 does not c-command copy 4, and copy 1 does not form a chain with copy 4 due to copy 2 being a WH intervener, much as in Nunes' theory. I will argue that copy 2 cannot form a chain with copy 4 because it has already checked its case feature, making it inaccessible to Form Chain. Since copy 2 is in a case checking position, any copy it forms a chain with must have a case feature that is accessible to the computational system (unchecked) following (2c). If copy 4 (and in fact copy 5 and 6) has had its case checked, this would render it invisible to the Form Chain operation.

We can see the effects of case marking in the parasitic domain in the sensitivity of some languages to the case of both the true and the parasitic gap. É. Kiss (2001) provides the data in

(13) from Hungarian, in which (13a) and (13c) are acceptable because they matching in case, but (13b) is unacceptable due to a mismatch in case between the true gap and the parasitic gap.

(13) a. [_{FP} Milyen iratokat tettél el *t* [_{CP} meilőtt elolvastál volna *pg*]]

what papers-ACC you-put away before you-had-read

“What papers did you put away before you had read?”

b. * [_{FP} Milyen iratokat veszték el *t* [_{CP} meilőtt elolvastál volna *pg*]]

what papers-ACC got-lost away before you-had-read

“What papers were lost before you had read?”

c. [_{FP} Milyen iratokat gondoltál [_{CP} meilőtt elolvastál volna *pg*][_{CP} hogy nem

what papers-ACC you-thought before you-had-read that not

szeretnél [_{CP} ha elvesznének *t*]]

you-would-like if got-lost

“Which papers did you think before reading that you would not like if were lost?”

Nunes argues citing Franks (1993), that the requirement for case matching between the matrix clause and the parasitic domain is not universal. The Russian data in (14) from Franks (1993) shows that the morphological forms of the case from the parasitic domain and the case from the matrix clause only need to be syncretic to satisfy this requirement.

- (14) a. *devuška, katoroji Ivan daval den'gi ti do togo, kak (on) stal*
 girl who-(Dat/Gen) Ivan-Nom gave money until (he) started
izbegat' pgi,
 to-avoid
 “The girl who Ivan gave money to until he started to avoid her.”
- b. *mal'čik, *kotoromu_i/*kotorogo_i Maša davala den'gi ti do togo, kak (ona) stala*
 boy who-Dat/ who-Gen Masha gave money until (she) started
izbegat' pgi
 to-avoid
 “The boy who Masha gave money to until she started to avoid him”

In (14a), we can see that the case necessary in the matrix domain and the parasitic domain do not need to match, as long as the form of ‘who’ is syncretic between the form the matrix clause needs (dative) and the form the parasitic domain requires (genitive). When that is not the case, such as in (14b), either form is ungrammatical.

However, the important aspect of the data in (13) is to show that there is a sensitivity to the case of the parasitic gap, suggesting that such a position is indeed sensitive to case. This is indeed confirmed in (14b), where the form of the object does not match the case necessary in the parasitic domain. So if we presume that case checking occurs in the parasitic domain, then it stands to reason that copy 4 in (12) should be case marked. The question then becomes why such case markings do not result in an ungrammaticality when the true gap and the parasitic gap do not require the same case, as in (15) (from Barss 1986), where the parasitic gap is accusative, but the matrix gap is nominative.

- (15) I wonder which papers John said were unavailable before reading.

The data in (15), as well as languages in which a mismatch in case between the parasitic and the matrix domain does not result in ungrammaticality, demands there be some way to reconcile the competing case requirements for the matrix and parasitic domains. I will assume that the phenomenon that forces morphological case matching in these languages is separated from the theory of featural case checking (McFadden 2004). This could come in the form of an extra constraint on recoverability or the constraints on the morphological form inserted after the structure has been built up. The potential independence of the case in the matrix domain and the parasitic domain can be accomplished if the case is reset for the matrix domain, and be checked again after having been checked in the parasitic domain. Boeckx et al. (2010) suggest that there is a strong relationship between case and thematic role features, and that when a new thematic role is acquired, case features can be erased. They show this in sentences like (16), where there are multiple case markings associated to the moved object are visible on floating quantifiers in Icelandic.

(16) Jón vonast til að leiðast ekki einum

Jon.Nom hopes to to be-bored not alone.Dat

“Jon hopes not to be bored alone.”

In (16), we see that John has nominative case as a result of its position in the subject of the matrix clause. However, we also see that *alone* has a dative marking, a quirky case assigned by the embedded verb. Under Boeckx et al.’s (2010) Movement Theory of Control, *Jon* has moved out of the embedded clause and into the external argument position of the matrix clause. Moving into this thematic position eliminates the case features assigned to *Jon*, and so it can be reassigned nominative case in the matrix clause. If a similar principle (that of removing the case

marking of an object that has moved into a thematic position and received a new thematic role) were applied to the parasitic gap constructions, then it would be quite reasonable the highest copy in the parasitic domain in (15) (and also copy 4 in (12)) to have case marking that is inaccessible to the computational system, So we can explain the grammaticality of (15) with the clearing of case when moving into the thematic position in the matrix clause, as well as the ungrammaticality of (12), since the process of moving into the matrix gap forces that copy to be assigned case again, which can be a different case than the one given in the parasitic domain. Therefore, in (15), the case feature of copy 4 is inaccessible to the computational system, since it has been checked earlier in the derivation. This makes a chain between copy 2 and copy 4 unable to be formed.

This analysis can be extended to the sentences in (17) (adapted from Nunes (2004)) as well. In (17), the parasitic gap is a subject in the parasitic domain. The structure for this is given in (18).

(17) **[Which man]_i t_i called you before pg_i met you?*

(18) $[_{CP} [\text{which man}]^1 [_{TP} [\text{which man}]^2 [_{VP} [_{VP} [\text{which man}]^3 \text{ called you}] [_{PP} \text{ before } [_{CP} [\text{which man}]^4 [_{TP} [\text{which man}]^5 [_{VP} [\text{which man}]^6 \text{ met you}]]]]]]]]]]?$

In the structure we see in (18), the formation of the chains {copy 1, copy 2, copy 3} and {copy 4, copy 5, copy 6} is done straightforwardly. This leaves copy 1 and copy 4 pronounced unless a chain can be formed between copy 4 and some member of the matrix chain. Once again, copy 3 cannot form a chain with copy 4 due to a lack of c-command, and copy 1 cannot form a chain with copy 4 due to copy 2's intervention (due to its WH feature). This leaves copy 2 to form a chain with copy 4, but this cannot work because copy 2 is in a case position, and

copy 4 does not have a visible case feature (having been checked in a lower position). This means both copy 1 and copy 4 are pronounced, which produces a crash at PF.

In this section, the key to correctly ruling out sentences in which the true gap c-commands the parasitic gap has been (i) the case marking of the parasitic object within the parasitic domain, and (ii) the fact that the position which might c-command into the parasitic domain, namely the specifier of TP, is a case position. This makes it impossible for the copy in the case position to form a chain with a case marked object high within the parasitic domain. So while the specific intervener proposed by Nunes' (2004) cannot prevent the formation of chains under the necessity for movement within the parasitic domain, the details of Form Chain still prevent the type of overgeneration that Nunes is trying to prevent in the case of the true gap c-commanding the parasitic gap.

4.2.3 Restriction against A-chains licensing parasitic gaps

The restriction against A-chains licensing parasitic gap constructions, as noted by Chomsky (1982) and Engdahl (1983), is meant to capture the ungrammaticality of sentences like (19). The structure assumes is given in (20).

(19) * $[\text{The book}]_i$ was filed t_i without my reading pg_i first

(20) $[\text{TP} [\text{the book}]^1 \text{ was } [\text{VP} [\text{VP} \text{ filed } [\text{the book}]^2] [\text{PP} \text{ without my reading } [\text{the book}]^3 \text{ first}]]]$

Sentences like (19) are accounted for much like the sentences that are ruled out by the restriction against true gaps c-commanding parasitic gaps. In both of these cases, there is a problematic copy that is in the specifier of the TP that intervenes between copy in a WH position

(the specifier of CP) and the parasitic domain. Broadly speaking, parasitic gap constructions are able to be made because there is a WH in the matrix domain that is able to form a chain into the parasitic domain without the intervention of a copy in a case position. Since the WH features of copies in the parasitic gap is not checked, the copy in the matrix CP is able to form both the chain in the matrix clause through WH movement, and form a chain into the parasitic domain, as long as there is no copy in an intervening case position in the matrix clause (the specifier of TP).

Given that both (19) and (17) have a copy in the specifier of TP, we can expect that the accounts for their ungrammaticality are similar. In (20), copy 1 and copy 2 form a chain (through a case relationship). However, copy 3 cannot enter into a chain with either of them. First of all, copy 2 does not c-command copy 3. Copy 1 does, but it is in a case position, which means that it must find a copy that has a visible case feature. Copy 3 has its case feature checked, and so that case feature is not visible to the computational system, and no chain can be formed. Therefore, copy 1 and copy 3 are both pronounced, which results in a crash at PF. Also, *my* would be a case intervener, which would also block a chain formed through a case relationship.

4.2.4 Summary of structural restrictions

The structural aspect of the restrictions can be summarized as the result of only a few key points. First, parasitic gap constructions are licensed when a WH phrase is in a position where it is in a checking relationship with a WH head (such as the specifier of CP). In Culicover (2001), this reflects the fact that the antecedent must be in an A' position, which is implicitly the restriction against A-chains licensing parasitic gaps. However, there is nothing in the theory presented here that can narrow the path to such 'split chain' structures to WH phrases

specifically. We can see an extension of this into the case domain in Boeckx et al.'s (2010) treatment of adjunct control. It is conceivable that any feature that triggers movement into a position might find similar effects. The high copy being in a WH position ensures that chains can be formed between this high copy and any c-commanded copy with its WH feature unchecked.

Second, it must c-command two (or more) copies of itself in two (or more) different branches of the structure (which is possible through sideward movement) and there must be no intervening WH phrases, including copies of the same WH phrase, intervening between this high WH position and the two copies that it will form a chain with. Such copies, especially in the specifier of TP, will block the crucial chain formation which allows for parasitic gap constructions, due to the TP position's status as a case checking position. This point reflects the restriction against the true gap c-commanding the real gap, since the two low copies must be in different branches without an intervener that c-commands both.

This account of parasitic gaps, based on a theory of Merge that entails sideward movement and the Form Chain operation at PF, naturally accounts for many of the restrictions on parasitic gaps. They fall naturally out of these two operations, which are crucially separated. The interesting aspect of parasitic gaps is the nature of the forked chain we find, which is problematic for a theory of movement that only predicts movement up the tree. Parasitic gap constructions are also a problem for any theory of the construal of copies that depends on a movement history. Such a theory either doesn't predict parasitic gaps at all (given an upward-only theory of movement) or overgenerates, since the sideward movement operation alone can presumably produce many arrangements of elements that cannot be resolved at PF with Form Chain. I think this justifies Nunes' separation of movement and chain formation, given the data

we encounter in language. While it is conceivable that they could go hand in hand, and that the computational system could keep such a record, it seems clear that this is not the case, given the restricted possibilities of sideward movement.

4.3 The NP restriction on parasitic gaps

The previous section focused on some of the structural restrictions on parasitic gaps and how these restrictions can be captured through the conditions on Form Chain. However, there is one more pair of restrictions to be accounted for. Specifically, parasitic gaps are restricted to occurring only when the parasitic gap is filled with an NP, and specifically a referential NP. We can see this in the data in (21) from Postal (1993).

- (21) a. *How_i did Deborah cook the pork *t_i* after cooking the chicken *pg_i*?
- b. *[How sick]_i did John look *t_i* without actually feeling *pg_i*?
- c. *This is a topic [about which]_i you should think *t_i* before talking *pg_i*.
- d. *[How many weeks]_i did he spend *t_i* in Berlin without wanting to spend *pg_i* in London?
- e. *Did_i John *t_i* call Mary after Paul *pg_i* call Sue?

These sentences show that parasitic gap constructions only allow for a specific type of phrase is able to escape from the parasitic domain. Nunes (2004), following Hornstein and Nunes (2002), proposes that Last Resort allows for sideward movement in order to satisfy the thematic requirements of the matrix clause in typical parasitic gap constructions such as (22).

- (22) [Which book]_i did Bob [_{VP} read *t_i*] after Mary recommended *pg_i*?

Under Hornstein and Nunes' account, in the case of (22), *which book* can move from the parasitic domain to the matrix domain because Last Resort licenses the movement to fulfill the thematic requirements of *read*. On the other hand, in the sentences in (21), none of the moved objects properly fulfill the thematic roles of the verb in the matrix domain. Specifically, in reference to (21b-d), Nunes (2004) argues that these types elements do not satisfy the requirement for last resort and behave like adjuncts. He points to the ungrammaticality of the sentences in (23) as evidence for this, as the same types of phrases cannot pass through weak islands. He takes this to be evidence that the matrix verbs in (19b-d) do not assign thematic roles to the moved object, at least not in the way that is relevant to Last Resort.

- (23) a. *How_i do you wonder whether Jane cooked the chicken *t_i*?
b. [How sick]_i do you wonder whether John felt *t_i*?
c. This is a topic [about which]_i I wonder whether you want to talk *t_i*?
d. [How many weeks]_i do you wonder whether he spent *t_i* in Berlin?

If true, this would certainly account for the disparity between referential NPs and these other types of phrases. However, this depends heavily on the thematic roles not being assigned to non-referential arguments in (23b-d), and therefore changing how they are inserted into the structure as a result. It also relies on the discharge of thematic roles being feature driven, as well as the certainty that the sentences in (23) do not do so in the same way.

Munn (2001) presents another account of the NP restriction in parasitic domains which instead relies on Szabolcsi and Zwarts' (1997) account of weak island effects (such as those we see in (23)) and the logical structure of non-individual denoting phrases. Since these non-

individual denoting phrases do not form Boolean algebras, they must be represented with higher semantic types. Szabolcsi and Zwarts suggest that since individual denoting NPs do form Boolean algebras, they are able to talk wide scope over other scopal elements, since all of the same Boolean operations can be performed on them. This is not true for non-individual denoting phrases, and therefore they cannot scope over these crucial scopal elements. In the case of the parasitic gaps in (19), Munn (2001) suggests that this involves the temporal operator, which moves up in temporal adjuncts as discussed in the previous chapter, such as in (21a), (21c), and (21e). The structure for (21a) is given in (24).

- (24) a. [$*\text{How}_i$ did Deborah [[[cook the pork] how_i] [Op_{temp} after [how_i [[cooking the chicken how_i] Op_{temp}]]]]?

While the WH phrase in (24) does not escape the operator within the parasitic domain, the c-commanding element that is found in the matrix clause does. This all is to say that the restriction against non-referential NP parasitic gap fillers is based on the ability of a particular phrases to semantically scope over elements that are obligatorily found in the adjunct phrase. In non-temporal adjuncts, such as *without* in (21b) and (21d), it is the negation contained in the meaning of *without* that provides the scopal restriction.

The advantages of Munn's account of parasitic gap restrictions over Hornstein and Nunes' is that we do not need to lean so heavily on the featural discharge of thematic roles (or lack thereof), and it does not need to be the case that this is the only relevant trigger for Last Resort in the sentences that must be ruled out. Instead, the semantic restrictions on which elements are allowed to scope over which other elements can take care of all of this work for us. This should be especially attractive since Szabolcsi and Zwarts' argument focuses on exactly the

types of weak islands that Hornstein and Nunes cite in (23). While Hornstein and Nunes rely on the status of these phrases as being sensitive to weak islands to justify the lack of thematic discharge that makes the sentences in (21) ungrammatical, Munn's account of parasitic gaps states that the referential NP restriction on parasitic gaps is derived from the same facts that the weak islands are. In fact, the adjuncts we find parasitic gaps in are weak islands in a sideward movement theory of parasitic gaps for semantic reasons.⁸ We will also see that these assumptions will be meaningful when accounting for the asymmetries between parasitic gap constructions and ATB constructions. This is because Munn's account of NP restrictions carries less baggage than Hornstein and Nunes' theory of thematic role discharge and its relationship to Last Resort when trying to account for those asymmetries.

4.4 ATB constructions and sideward movement

Nunes (2004) frames his account of the NP restrictions on parasitic gap constructions in contrast to the lack of these restrictions on ATB constructions. Nunes follows Munn (1992, 1993) in assuming that the same process that underlies ATB movement also underlies parasitic gap constructions. The restriction on parasitic gap construction can be seen when comparing the ungrammatical constructions in (21) to the grammatical ATB constructions in (25).

- (25) a. How_i did Deborah cook the pork *t_i* and Jane cook the chicken *t_i*?
- b. [How sick]_i did John look *t_i* and Betty say he actually felt *t_i*?
- c. This is a topic [about which]_i you should think *t_i* and I should talk *t_i*.

⁸ And in fact, strong islands due to the phase architecture described in the last chapter.

- d. [How many weeks]_i did you spend t_i in Berlin but want to spend t_i in London?
- e. Did_i John t_i call Mary and Paul t_i call Sue?

These sentences show that restrictions we find against the movement of objects that are not referential NPs does not hold in the case of ATB constructions. Nunes (2004), I believe rightly, wants to be able to account for ATB constructions with sideward movement in a manner similar to parasitic gaps. However, this means that his restrictions that allow only referential NPs in parasitic gaps must not hold in ATB constructions. Since Nunes relies on Last Resort to justify any sideward movement, it must be the case that Last Resort justifies the movement between conjuncts in (25). However, Nunes specifically states that thematic role discharge cannot be the explanation when the moved object is not a referential NP. This forces Nunes to say that Last Resort must somehow justify the sideward movement between conjuncts in (25) in another way. Following Hornstein and Nunes (2002), he states that instead of Last Resort being triggered by the discharge of thematic roles, coordinate structures are subject to a Parallelism Requirement, citing Chomsky and Lasnik (1993), Chomsky (1995), and Fox (1995). He goes on to state that movements apply to all conjuncts if it applies to any, comparing this to the Coordinate Structure Constraint.

This view of Last Resort is highly problematic due to its dependence on lookahead. Let us take under consideration the derivation of (25b). In (26), I show the relevant portion of the derivation.

(26) K = [and Betty [_{VP} [how sick]_i [_{VP} say [_{CP} [how sick]_i he actually felt [how sick]_i]]]]]

L = look

The question that is most problematic for Nunes' theory is how the syntax, given the structure in (26), can determine that if there is no movement, then there will be a violation on Last Resort. The violation of the Parallelism Requirement is presumably one that is encountered at the interfaces, since introducing such an ability to crash in the middle of the derivation would involve adding additional machinery to the narrow syntax. At the very least this entails a level of representation that must be accessible to be examined. Additionally, any such checking would require quite a bit more structure, such as both conjuncts being attached and the resulting movement out in order to successfully detect a violation of the Parallelism Requirement in coordinate structures. If this is true, then at the state of the derivation in (26), the narrow syntax cannot know about the impending crash that will justify the Last Resort movement. This isn't to say that Last Resort is a problem in all cases. Take the snapshot of the derivation of (22) presented in (27).

(27) K = [after [which book]_i Mary [which book]_i recommended [which book]_i]

L = read

In this case, if *which book* is not moved, it is clear that *read* will not have discharged its thematic role and satisfied its selectional requirements. This can be understood from the uninterpretable features on *read* that will not be checked if the derivation were to move on. In this case, Last Resort has immediate access to the negative consequences of a lack of action (in this case, the sideward movement of *which book*). The ability to see the impending failure to satisfy the Parallelism Requirement is not available to the narrow syntax in the derivation in (24). Therefore, Nunes' account of the lack of NP restrictions in ATB constructions seems to be problematic.

Notice that Munn's (2001) account of the NP restrictions on parasitic gaps does not predict the same problem on ATB constructions. Since the NP restrictions are characterized by weak island effects from the types of adjuncts we find in parasitic gap constructions, this predicts that these restrictions should only be relevant when those weak islands exist in one of the conjuncts, as in (28). This has the advantages mentioned in the previous section of deriving both weak islands and parasitic gap restrictions from the same scope effect instead believing they are derived from separate (though related) causes.

(28) ??[How sick]_j did John look t_i and Betty wonder whether he felt t_i ?

Adopting Munn's account of NP restrictions does begin to call into question what the motivations of such sideward movements are. Under Nunes' (2004) theory, Last Resort was the entire motivation of sideward movement. Under his theory, the discharge of thematic roles was the primary motivation for such movement. I am proposing that such requirements do not exist. It is still the case that Last Resort forces moved object to merge immediately as Nunes does. The computational system will not allow a copy to be made and then for that copy to hang in the computational space without immediate purpose in structure building. In this sense, Last Resort (in the loosest sense) is still governing sideward movement. However, I propose that if an object is available in the derivational workspace to be copied, it can be. In the case of parasitic gaps, only NPs of a particular sort are available when the derivation reaches the point shown in (27). Below in (29), I have removed the material that would be spelled out by cyclic, phase-by-phase Spell-out, as described in the previous chapter.

(29) K = [PP after [CP [which book]_j C $\bar{\mathbb{P}}$]

L = read

As we can see in (29), the entire TP in the parasitic domain has been removed from the derivational workspace. This is because the introduction of the head of the PP, *after*, has triggered the spell out of the complement of the next phase down, therefore the TP is spelled out as the complement of CP. This greatly restricts the possible targets for movement to only those which can make it to this ‘escape hatch’. This domain is slightly larger in parasitic gap constructions where there is no CP, such as in untensed adjuncts, but even then is limited to the space between the prepositional head and the next highest phase. This means that only a small portion of phrases should be able to make it to such a position. It is important to keep in mind that the scopal account of the NP restrictions still holds when larger domains are present, since parasitic gap constructions still demand that the WH phrase be able to scope over the parasitic domain, and the parasitic domain itself is a weak island.

ATB constructions still function under the same set of Spell-out expectations. However, there is no weak island built into their structure. There are, however, restrictions on extraction out of only one of the conjuncts. This can be seen in (30-32).

- (30) a. [Which book]_i did Bob read *t_i* and Mary recommend *t_i*?
 b. *[Which book]_i did Bob read an article and Mary recommend *t_i*?
 c. *[Which book]_i did Bob read *t_i* and Mary recommend an article?
- (31) a. [How sick]_i did John look *t_i* and Betty say he actually felt *t_i*?
 b. *[How sick]_i did John look terrible and Betty say he actually felt *t_i*?
 c. *[How sick]_i did John look *t_i* and Betty say he actually felt terrible?
- (32) a. Did_i John *t_i* call Mary and Paul *t_i* call Sue?

b. *Did_i John *t_i* call Mary and Paul called Sue?

c. *Did_i John called Mary and Paul *t_i* call Sue?

These sentences show that ATB constructions allow there to be gaps in both conjuncts, but either of the conjuncts individually bars extraction. This is where the Parallelism Requirement becomes relevant. While it requires look-ahead to motivate an operation that would avoid violating a restriction at the interfaces, the Parallelism Requirement can act as a filter at the LF interface, which only allows the (a) versions of the sentences in (30-32) to converge. The versions which feature extraction out of only one conjunct will be rejected due to a violation of the Parallelism Requirement, but those in which both contain a gap will pass such a requirement. In terms of how ATB constructions are created, I will assume that Nunes is largely correct in his implementation. The derivation for (30a) is given in (33).

(33) [Which book]_i did Bob read *t_i* and Mary recommend *t_i*?

a. Second conjunct's vP constructed

K = [_{VP} Mary recommended [_{VP} ~~recommended~~ [which book]_i]]

b. Movement of *which book*

L = [_{VP} [which book]_i [_{VP} Mary recommended [_{VP} ~~recommended~~ [~~which book~~_i]]]]

c. Construction of second conjunct's TP

M = [_{TP} Mary did [_{VP} [which book]_i [_{VP} ~~Mary~~ recommended [_{VP} ~~recommended~~ [~~which book~~_i]]]]]

d. *And* merged

N = [_{BP} and [_{TP} Mary did [_{VP} [which book]_i [_{VP} ~~Mary~~ recommended [_{VP} ~~recommended~~ [_{VP} ~~[which book]_i]]]]]]]~~

e. Introduction of first conjunct's verb *read*

N = [_{BP} and [_{TP} Mary did [_{VP} [which book]_i [_{VP} ~~Mary~~ recommended [_{VP} ~~recommended~~ [_{VP} ~~[which book]_i]]]]]]]~~

O = read

f. Sideward movement of *which book*

N = [_{BP} and [_{TP} Mary did [_{VP} [~~which book]_i [_{VP} ~~Mary~~ recommended [_{VP} ~~recommended~~ [_{VP} ~~[which book]_i]]]]]]]~~~~

P = [_{VP} read [which book]_i]

g. Construction of first conjunct's vP

N = [_{BP} and [_{TP} Mary did [_{VP} [~~which book]_i [_{VP} ~~Mary~~ recommended [_{VP} ~~recommended~~ [_{VP} ~~[which book]_i]]]]]]]~~~~

Q = [_{VP} Bob read [_{VP} ~~read~~ [which book]_i]]

h. Movement of *which book*

N = [_{BP} and [_{TP} Mary did [_{VP} [~~which book]_i [_{VP} ~~Mary~~ recommended [_{VP} ~~recommended~~ [_{VP} ~~[which book]_i]]]]]]]~~~~

R = [_{VP} [which book]_i [_{VP} Bob read [_{VP} ~~read~~ [~~which book]_i]]]]]~~

i. Construction of first conjunct's TP

N = [_{BP} and [_{TP} Mary ~~did~~ [_{VP} ~~[which book]_i~~ [_{VP} ~~Mary~~ recommended [_{VP} recommended [~~which book~~_i]]]]]]

S = [_{TP} Bob did [_{VP} [which book]_i] [_{VP} ~~Bob~~ read [_{VP} read [~~which book~~_i]]]]]

j. Second conjunct adjoined to first conjunct

T = [_{TP} [_{TP} Bob did [_{VP} [which book]_i] [_{VP} ~~Bob~~ read [_{VP} read [~~which book~~_i]]]] [_{BP} and [_{TP} Mary ~~did~~ [_{VP} ~~[which book]_i~~ [_{VP} ~~Mary~~ recommended [_{VP} recommended [~~which book~~_i]]]]]]]

k. C merged, VPs spelled out

U = [_{C'} did [_{TP} [_{TP} Bob ~~did~~ [_{VP} [which book]_i] [_{VP} ~~Bob~~ read [_{VP} read [~~which book~~_i]]]] [_{BP} and [_{TP} Mary ~~did~~ [_{VP} ~~[which book]_i~~ [_{VP} ~~Mary~~ recommended [_{VP} recommended [~~which book~~_i]]]]]]]]]

l. Movement of *which book* to specifier of CP

[_{CP} [which book]_i] did [_{TP} [_{TP} Bob ~~did~~ [_{VP} ~~[which book]_i~~ [_{VP} ~~Bob~~ read [_{VP} read [~~which book~~_i]]]] [_{BP} and [_{TP} Mary ~~did~~ [_{VP} ~~[which book]_i~~ [_{VP} ~~Mary~~ recommended [_{VP} recommended [~~which book~~_i]]]]]]]]]

In (33a), the vP of the second conjunct is built up, and due to the spell out associated with the copy operation, the Spell-out of the verb in the VP as it moves to the vP. Then in (33b), *which book* moves to the edge of the vP, and spells out the lower copy. Next, the rest of the second conjunct's TP is built up in (33c), including the movement of the subject to the specifier

of TP and the subsequent spell out of the lower copy. The second conjunct is then taken as the complement of *and* in (33d). I will be assuming that conjunctions are structurally hierarchical, and that the Boolean phrase is an adjunct to the phrase it conjoins to (Munn 1993), though this bit of analysis is not crucial to the theory under discussion. In (33e), the construction of the first conjunct begins with the introduction of the verb *read*. The sideward movement of *which book* occurs in (33f), which causes its spell out in the second conjunct. Then in (33g), the rest of the first conjuncts vP is constructed, and *which book* is moved to the edge of the vP in (33h). In (33i), the first conjuncts TP is constructed, which involves the sideward movement of the T from the second conjunct to the first. Then BP is adjoined to the first conjunct in (33j). C is introduced into the derivation in (33k), which entails the movement of *did* to C as well as the spell out of the complement of the vPs in both conjuncts due to the cyclic Spell-out of phases. Lastly, *which book* is moved up from the first conjunct into the specifier of CP, thus deriving the sentence in (30a). At PF, *which book* forms a chain with its copies in both conjuncts through the unchecked WH features they possess, resulting in Chain Reduction deleting them for linearization.

This section has shown how ATB structures can be derived through sideward movement. The derivation is similar to what Nunes (2004) would suggest, but the motivations and restrictions on such a process are distinct. In Nunes (2004), sideward movement in ATB constructions is driven by Last Resort attempting to satisfy a Parallelism Requirement. However, I argued that using Last Resort to motivate a movement to avoid a violation requires lookahead to a point in the structure at which such a violation would be detectable. This form of lookahead is barred from the computational system, and so Last Resort cannot be the motivation for such a movement. Instead I propose that there are far less restrictions on the motivations for

movement as long as all movements are toward the goal of immediate structure building. This allows ATB structures to be accounted for without depending on looking ahead in the derivation to motivate Last Resort. I also demonstrated that such a process can successfully derive ATB structure. In addition, the lack of NP restrictions on ATB constructions in comparison to parasitic gap constructions comes not from a different treatment from Last Resort, but instead from the lack of weak island effects caused by scopal restrictions on non-individual denoting syntactic objects.

4.5 Conclusion

This chapter shows that many of the restrictions on parasitic gap constructions can be accounted for successfully within a sideward movement theory of parasitic gaps. In fact, I argue that the coverage that comes from this system provides a more principled account of (i) the construal of parasitic gaps in terms of the copy theory of movement, (ii) how chains are formed in these constructions through Nunes' (2004) Form Chain, and (iii) how these facts can be accounted for within a system that assumes both movement within the parasitic domain and without reliance on lookahead in order to motivate operations. Both the copy theory of movement and Form Chain are general purpose mechanisms in the syntax that interact to allow for parasitic gap constructions. None of the restrictions come from rules that are designed to target the problem specifically, but are used to account for a variety of data. These advantages make this proposal powerful and principled, and avoids an over-fitted theory that cannot be generalized. This can be seen in the way that the theory extends to ATB constructions in a very straightforward manner. The only principle that is not a straightforward extension from the account of parasitic gap constructions is the Parallelism Requirement, which seems to be an intrinsic and crucial semantic requirement of coordinated structures.

In the next chapter, I will expand the current sideward movement theory to account for yet another phenomenon which has been argued to involve sideward movement, relative clauses. Much like was done with ATB constructions in this chapter, the current premises of the theory capture the details of sideward movement in relative clauses. However, we will see that some of the underlying assumptions made in the sideward movement literature about the directionality of movement and the course of the derivation prove not to hold in deriving relative clause structures.

5 Relative clauses and sideward movement

5.1 Introduction

In the previous chapters, I have made an argument for the explanatory power of sideward movement by showing how it can be used to analyze parasitic gap and ATB constructions. It should be the case that we can use the same tool throughout the syntax. This means that we should find other types of structures constructed using sideward movement. Relative clauses have also been analyzed using sideward movement by Nunes (2004) and Henderson (2007). Each of these two presents a different analysis of relative clauses, and there is no guarantee that either will continue to hold under the phase architecture and restrictions proposed in Chapter 3.

In addition to the restrictions on the derivation and sideward movement that might affect a sideward movement account of relative clauses, many of the facts surrounding relative clause constructions depend on the ability for adjuncts such as relative clauses to be able to merge into the structure counter-cyclically. While Nunes (2004) explicitly states that this must not be possible in order to account for the patterns we see in parasitic gap constructions, there is a great deal of data that depends on late adjunction which must be accounted for. Until recently, late adjunction was the only account of certain patterns of reconstruction and anti-reconstruction effects, as well as the facts we find in extraposed adjuncts and the Antecedent-Contained Deletion (ACD) constructions. Sportiche (2016) presents a theory of Neglect, or deletion at the interfaces which provides an alternate account of such effects, which I will ultimately adopt.

Lastly, while Sportiche (2016) accounts for a great deal of the effects typically attributed to late adjunction that surround relative clause constructions, I will show that there is still data unaccounted for that shows a sideward movement theory of relative clause construction is still necessary in order to properly account for all of the facts presented.

This data will also call into question Nunes' claim that sideward movement must proceed from more embedded to less embedded structures. The claim that the derivation proceeds from more embedded to less embedded environments is intended to account for the directionality of movement in his account of parasitic gaps. This claim has been made before (Chomsky 2000, 2001, 2004), but this seems to largely serve as a description of the course of the derivation rather than following from something deep about the computational system. In Chapter 2, I argued that although in some cases this can be derived (such as how the lower part of a structure must be built before the higher parts of the structure), in the case of adjunction it is clear that the structure can be conceivably built in any order. I also argue that it would require a knowledge of the structure being built (that is to say, lookahead) in order to reasonably assume that adjuncts must be built before the phrase that they will attach to. If we assume that the order that phrases are built in is not determined from their 'embeddedness' specifically, then we should expect to find cases in which derivations begin with the phrase that will be adjoined to before building the adjunct. If we predict the possibility that adjuncts can be constructed after the object they will later adjoin to, it becomes necessary demonstrate the result of this possibility. It also predicts that there might be circumstances in which sideward movement could have an object move into an adjunct, which goes more directly against Nunes' (2004) claim about the directionality restrictions on sideward movement. In this chapter, I will demonstrate that given all of the facts

surrounding relative clause construction, it must be the case that they are built through sideward movement into the relative clause from the head noun.

5.2 Reconstruction and the Copy Theory of Movement

Much of the evidence for late adjunction comes from a lack of reconstruction effects. Reconstruction effects are effects of the Binding Theory which only occur if we assume that a moved object is acting as though it is in a position that it has moved from. Binding effects are a commonly used method with which to test the syntactic relationships between two phrases. When binding and movement interact, however, we find reconstruction effects that are surprising based on the surface word orders. The sentences in (1) show the effects that movement has on two types of noun phrases - NPs that contain an anaphor (1a,c) and NPs that contain an R-expression (1b,d).

- (1) a. [Which picture of himself_i] did Steve_i prefer t_i ?
b. [Which picture of Steve_i] did he_{j/*i} prefer t_i ?
c. [Which picture of himself_i] did Steve_i prefer *which picture of himself_i*?
d. [Which picture of Steve_i] did he_{j/*i} prefer *which picture of Steve_i*?

In (1a), we see that even though *Steve* does not c-command *himself* on the surface, Principle A is not violated. Also, we see in (1b) that reconstruction effects can cause a violation of Principle C, even though on the surface, there is no c-command relationship between *he* and *Steve*. This is a problem if we assume that the words can only be in their surface positions since these moved objects have binding facts that behave as if they haven't moved. However, if we assume a Copy Theory of Movement in which the DPs inside of the WH-phrase are in the

surface position as well as the position from which they moved, we see that the pattern becomes clearer. This is a natural consequence of the Copy Theory of Movement. We can see in (1c) and (1d) that when we have the lower copy of the WH-phrase in the structure, the binding facts are fairly straight-forward since *himself* is able to be bound by *Steve* in (1c) and *Steve* is bound by *he* in (1d). When looking at the data in this manner, notice it becomes necessary to assume only one copy within a member of a chain needs to be able to be bound to license it throughout the chain, otherwise the high copy of *himself* in (1c) would still be unbound and result in a Principle A violation. The ability to account for reconstruction effects straightforwardly is one of the primary advantages of the Copy Theory of Movement.

5.2.1 Anti-reconstruction and late adjunction

On the other hand, we see that reconstruction is not always obligatory (Freidin 1986) in (2).

- (2) a. Which picture that Steve_i took did he_{i/j} prefer?
 b. Which man near the hose_i did it_i spray?

If we assume that the entire WH-phrase in the sentences in (2) is reconstructed, we would expect a Principle C violation. However, we see anti-reconstruction effects here in which reconstruction of at least part of the phrase does not take place. We should certainly expect that if movement is handled with copies in one instance, this should be the case with all movement. What is the difference between (1b) and the sentences in (2)? Van Riemsdijk and Williams (1981) point out that we find anti-reconstruction effects in situations where the R-expression that would be violating Principle C is contained within an adjunct (in this case *that Steve took*). They

propose that in cases where R-expressions violate Principle C due to reconstruction effects (1b), the R-expression itself is either in an argument position or contained in a phrase in an argument position, and arguments must be merged as soon as possible in the derivation. If we assume a Copy Theory of Movement, this means that complements must be merged at the position in which the phrase is base generated into before it can undergo movement (via a copy). Thus, in the example of (1b), everything in the phrase is c-commanded by the pronoun *he* including the R-expression *Steve*. On the other hand, R-expressions that escape Principle C violations are inside adjuncts, such as the sentences in (2), where the R-expressions are in a phrase that is adjoined to the NP picture.

Lebeaux (1988, 1991) suggests that adjuncts have the option of merging counter-cyclically. In a Copy Theory of Movement, the lowest copy does not need to contain a copy of the adjunct and thus it can escape the Principle C violation. For an example of this derivation, see (3).

- (3) a. Did he prefer [which picture]
b. [Which picture] did he prefer which picture
c. [Which picture that Steve took] did he prefer which picture

In (3a), we see what we expect to be the question before the WH-phrase undergoes movement. The important aspect of this stage of the derivation is the fact that the WH-phrase does not contain the adjunct. In (3b), we see the copy of *which picture* being merged into Spec CP, which is still identical to the base generated version. Then in (3c), the adjunct is merged into the WH-phrase. We see that this is done by adding it in the middle of the existing structure.

5.3 Theories of relative clauses

Late adjunction and reconstruction play an important role in theories of relative clauses. In the literature there are two widely used structures in current theories of relative clauses: a raising structure and a matching structure. In a raising structure (Bhatt 2002, Bianchi 1995, Kayne 1994), the head noun is base generated within the relative clause and then raises up to its surface position in Spec CP such as is demonstrated in (4).

(4) [DP The [CP[NP picture_i][C' that Bill took t_i]]]

One of the most compelling arguments for a raising analysis is the fact that idioms show a close relationship between the head noun and the gap position inside the relative clause. We see in (5a) and (5b) that idioms are not easily broken up (Brame 1968 via Bhatt 2002).

- (5) a. We made headway
b. *The headway was satisfactory
c. What headway was Bill hoping to make?
d. The headway that we made was satisfactory

In (5a), we see the idiom *make headway* in its standard configuration. In (5b) we see that without the rest of the idiom, *headway* is not licensed. However, (5c) shows us that *headway* can be licensed in non-standard configurations. The low copy of *what headway* is the complement of the verb *make*, which licenses the idiom in this instance. From this example, we see that idioms can be licensed through reconstruction. We can create a similar story for (5d). If we assume a raising theory of relative clauses, we know that the low copy of *headway* is base generated

within the relative clause, which licenses its use in (5d). This is strong evidence that there is a copy of the head noun within the relative clause.

Another argument in favor of the raising structure the reconstruction effects of noun phrases containing R-expression and anaphors from the head noun position. Traditionally reconstruction effects are the result of movement, and in these cases, it is compelling to draw the same conclusion. In (6), we see that the complement of *of* is subject to binding effects from reconstruction within the relative clause (from Schachter (1973)).

- (6) a. The [opinion of himself_i] that John_i has is favorable
b.* The [opinion of John_i] that he_i has is favorable

These examples point to an analysis where the head noun moves out of the relative clause. Such a conclusion is consistent with a raising theory.

Late adjunction is a problem for a raising analysis of relative clauses though. The derivation in (7) shows how we might derive a relative clause under a raising analysis.

- (7) Which picture that Steve took did he prefer?
a. Steve took picture
b. [_{CP} picture_i that Steve took t_i]
c. [Which picture_i [t_i that Steve took t_i]]
d. Did he prefer [which picture_i [t_i that Steve took t_i]]

e. [Which picture_i [t_i that Steve took t_i]] did he prefer [which picture_i [t_i that Steve took t_i]]

f. [Which picture_i [t_i that Steve took t_i]] did he prefer [which picture_i [t_i that Steve took t_i]]

We can see that it is necessary to have the relative clause attached in one of the earliest steps of the derivation (in this case (7a)). This is the case because we know that in a Copy Theory of Movement, *picture* must be in the complement of V in (7d). However, because we include the relative clause this early in the derivation, we are left with a copy of the relative clause low in the tree. This is a problem because looking at (7f), we see that *Steve* in the low copy of the relative clause is c-commanded by *he*, which should be a violation of Principle C. This will be important to keep in mind when examining potential explanations of relative clauses and their status with relation to noun phrases.

Others (Quine 1960, Montague 1974, Partee 1975, Chomsky 1977, and Jackendoff 1977) have argued for a matching structure, in which the head noun is base generated outside of the relative clause. Most frequently, an operator is used to reconcile the relationship between the head noun and the argument of the verb in the relative clause such as the structure in (8).

(8) [DP The [NP [NP picture][CP Op_i that Bill took t_i]]]

The one of the strongest arguments for this structure is its ability to accommodate late adjunction. We saw in (7) how late adjunction is an issue for any raising analysis of the form discussed above. A matching analysis allows for the head noun to be present in the derivation independently of the relative clause. On the other hand, we lose the tight connection between the head noun and the element moved out of the relative clause that got us the reconstruction and

idiom facts. Munn (1994) discusses the difficulties with a matching analysis in more detail, pointing out that for a matching analysis to satisfy the reconstruction and idiom facts, there are a number of unattractive stipulations that must be made on the operator in the relative clause that make it behave almost precisely like a moved object. Nunes' (2004) criticism of operator analyses of parasitic gap constructions makes similar arguments.

There have been many arguments for either or both of these structures as the explanation for the structure of relative clauses. The most notable of recent theories is that of Hulsey and Sauerland (2006) who argue that the structure for the relative clause is potentially ambiguous between raising and matching, though some sentences will force a particular structure. They give evidence for this in the form of extraposition data below in (9) and (10).

- (9) a. Mary praised the headway that John made
b. *Mary praised the headway last year that John made
c. Mary praised the pot roast yesterday that John made
- (10) a. I saw the picture of himself that John liked
b. *I saw the picture of himself yesterday that John liked
c. I saw the picture of Clinton yesterday that John liked

We see in (9a) that the idiom *make headway* is fine in a typical relative clause position, but when the relative clause is extraposed such as in (9b), we see that the sentence is unacceptable. It is important to contrast this with (9c) to show that extraposition alone is not the culprit and it seems that the extraposition of this relative clause is the problem. Hulsey and Sauerland argue that this is due to the idiom and its forcing of a raising structure in the relative

clause. (10) shows the same pattern as (9), but this time with a simple reconstruction of the anaphor causing the raising structure. Hulsey and Sauerland argue that the distinction between (9b) and (9c) is that while (9c) is fine because it has a matching structure, and the relative clause can therefore late adjoin, (9b) is bad because it has a raising structure, and the relative clause phrase isn't a constituent eligible for movement. It will become crucial that the theory relies heavily on a late adjunction story for extraposition. If the extraposition of these adjuncts were not a result of late adjunction, then it would be necessary to account for how some relative clauses are licit when extraposed (9c, 10c), but others are not (9b, 10b).

A matching theory as presented above would be problematic as we need to be able to explain the raising effects. While a raising theory of relative clauses has trouble dealing with anti-reconstruction effects, Hulsey and Sauerland solve this by assuming that when there are raising effects, we have a raising structure; when there are anti-reconstruction effects, the relative clause has a matching structure; and the structure is ambiguous in other cases.

5.3.1 The raising paradox and sideward movement

Hulsey and Sauerland's (2006) theory of relative clauses predicts that since raising effects must be accounted for using a raising structure, and matching effects must be accounted for using a matching structure, there should be no relative clause that has raising effects and anti-reconstruction effects. Henderson (2007) points out that this is not the case with sentences such as (11).

- (11) What headway that John made did he later regret?

In (11), one must assume that *headway* is reconstructed into the relative clause in order to be licensed. On the other hand, we must also assume that the relative clause is late adjoined in order to obviate the Principle C effects we would expect if *John* were reconstructed. This leaves us between a rock and a hard place. If *headway* is raised out of the relative clause, then the clause must have merged low and not late adjoined. However, if Principle C is to be obviated by *John*, then the relative clause must have been late adjoined. Henderson calls this *the raising paradox*.

(12) The *raising paradox* (From Henderson 2007)

- a. The heads of relative clauses must be merged cyclically.
- b. The heads of relative clauses originate within the relative clause that modifies them.
- c. Relative clauses must be late merged (acyclically).

We are left needing to create a structure in which the relative clause is late adjoined, but we are also able to have *headway* reconstructed.

Recall that in Nunes' (2004) theory of parasitic gaps, he had to assume that there could be no counter-cyclic merger. This is not the case for my theory of parasitic gaps since the crucial restriction against sideward movement to A' positions was accomplished through the use of Agree. However, Nunes' claim that there is no counter-cyclic merger motivates him to provide a way to build up relative clauses without the use of late adjunction, recreated below in (13).

(13) a. Relative clause constructed, matrix verb *discuss* introduced

$$K = [_{CP1}[\text{which claim}]_k [_{CP1} \text{ that John made } [\text{which claim}]_k]]$$

L = [discuss]

b. Matrix CP built up

K = [_{CP1} [which claim]_k [_{CP1} that John made [which claim]_k]]

M = [_{CP2} was + Q [he willing to discuss [which claim]_k]]

c. Relative clause CP merged into specifier of the matrix CP

N = [_{CP2}[_{CP1}[which claim]_k [_{CP1} that John made [which claim]_k]] [_{c'} was +Q
[he willing to discuss [which claim]_k]]]

We see in (13a) that we have already built up the relative clause and the head noun has raised out into the Spec CP position. In (13b), we see the constituent that moved in (13a) moves again, but instead of moving higher within K, we see it moves to merge with *discuss* in L and then extended with the rest of the matrix clause to create M. We then see that the entire relative clause in K is merged into the Spec CP position in M to create the structure we find in (13c).

Nunes' goal is to give a clear example of how sideward movement functions, but this particular analysis is a bit problematic. The primary issue is that it suggests that WH phrases that contain relative clauses undergo a special and specific method of movement that is unnecessary in typical WH movement. In an example of WH movement without a relative clause involved, we would want to be able to say that an object moved directly from within the matrix clause up to Spec CP. In addition to providing a simpler mechanism, this process also preserves the relationship between the thematic role of the position the WH phrase was moved from and the WH phrase itself. There is no chain created between *which claim* within the matrix VP and the CP that is merged into the specifier of CP. In (13c) we are assuming that in this special case the

WH phrase in fact moves into the low position from another derivational object. While a theory that takes advantage of sideward movement might allow this in principle, it is strange that all other WH movement works one way (the WH moving from its canonical position up to Spec CP), while once a relative clause is introduced, it works another way (the WH moves from outside the phrase into the low position), which seems unavailable to normal WH movement. This is an unattractive state of affairs.

Henderson (2007) gives his own version of how relative clauses might be derived while avoiding this particular pitfall. I show his derivation in (14).

- (14) a. Relative clause TP built up

$$K = [_{TP} \text{John } [_{VP} \text{read } [_{NP} \text{book}]_i]]$$

- b. *Book* moves to specifier of relative CP

$$L = [_{CP} [_{NP} \text{book}]_i \text{ that } [_{TP} \text{John } [_{VP} \text{read } [_{NP} \text{book}]_i]]]$$

- c. Sideward movement of *book* into DP

$$L = [_{CP} [_{NP} \text{book}]_i \text{ that } [_{TP} \text{John } [_{VP} \text{read } [_{NP} \text{book}]_i]]]$$

$$M = [_{DP} \text{the } [_{NP} \text{book}]_i]$$

- d. Late adjunction of relative clause to *book*

$$N = [_{DP} \text{the } [[_{NP} \text{book}]_i [_{CP} [_{NP} \text{book}]_i \text{ that } [_{TP} \text{John } [_{VP} \text{read } [_{NP} \text{book}]_i]]]]]$$

In (14a), we see the relative clause is built up to the TP. In (14b), the head noun *book* is moved to the Spec CP position. *Book* is then moved out of L to create DP that will contain the

relative clause in M. In (14d), we see the relative clause is late adjoined to the head noun in N. This gives us a moveable object, as is normal for WH movement. Also, the relative clause is a separate object that is available to merge at the appropriate stage in the derivation which is helpful when accounting for anti-reconstruction effects.

The Henderson derivation of relative clauses allows us to deal with the *raising paradox* by providing us with a way to “raise” the head noun out of the relative clause without committing the relative clause to being merged early. This provides the flexibility needed to deal with anti-reconstruction effects. Thus, we show that the derivation can deal with the sentence that is problematic for Hulsev and Sauerland repeated here as (15).

(15) What headway that John_i made did he_i later regret?

a. Relative clause TP constructed

$$K = [_{TP} \text{John}_i \text{ made } [_{NP} \text{headway}]^k]$$

b. *Headway* moved to specifier of relative CP

$$L = [_{CP} [_{NP} \text{headway}]^k [_{TP} \text{John}_i \text{ made } [_{NP} \text{headway}]^k]]$$

c. Sideward movement of *headway* into DP

$$L = [_{CP} [_{NP} \text{headway}]^k [_{TP} \text{John}_i \text{ made } [_{NP} \text{headway}]^k]]$$

$$M = [_{DP} \text{what } [_{NP} \text{headway}]^k]$$

d. Matrix clause built up, C merged

$$L = [_{CP} [_{NP} \text{headway}]^k [_{TP} \text{John}_i \text{ made } [_{NP} \text{headway}]^k]]$$

N= [C' did he_j later regret [DP what [NP headway]^k]]

e. Movement of *what headway* to specifier of CP

L = [CP [NP headway]^k [TP John_i made [NP headway]^k]]

O = [CP [DP what [NP headway]^k]^m did he_j later regret [DP what [NP headway]^k]^m]

f. Late adjunction of the relative clause

P = [CP [DP what [[NP headway]^k [CP [NP headway]^k [TP John_i made [NP headway]^k]]]]]^m did he_j later regret [DP what [NP headway]^k]^m]

Once again, we see the relative clause built up in (15a). Then, the head noun is moved up to Spec CP of the relative clause in (15b). Then, the head noun, which must be licensed as part of an idiom, is moved out to form the DP that contains the relative clause in (15c). (15d) shows the matrix clause being built up, notably without any Principle C violations with *John*, since *John* and *he* are not in the same object of the derivation yet. (15e) shows the WH movement from the object position to Spec CP of the relative clause. Then in (15f), the relative clause is late adjoined to the head noun, so that *John* is not bound by *he*. We also see that *headway* is licensed within the relative clause, which makes the whole chain for *k* licensed.

This model of relative clauses has three key aspects that will be relevant to the forthcoming discussion of late adjunction. The first two aspects work in concert and are key to Henderson's analysis: the use of sideward movement and late adjunction. Late adjunction is necessary in order to obtain the matching effects we see, and sideward movement is necessary in order for late adjoined relative clauses to be able to have actual copies of the head noun within

them in order to account for raising effects. If late adjunction is to be replaced in the theory, it must be replaced in a way that also accounts for sentences with a combination of raising effects and anti-reconstruction effects. This will be the standard by which Sportiche's (2016) Neglect account of these effects will be measured in the next section. The third aspect to keep in mind is that this is truly a raising story, in that the head noun is base generated within the relative clause and then raises into the head noun position. I will look into the problems with this aspect of the account in 5.4.1.

5.4 Neglect as an alternate account

Sportiche (2016) presents the operation of Neglect as an operation that we must assume is applied to LF representations ubiquitously in order to ensure legible representations for semantic computation. Neglect is an operation which selectively ignores material in order to eliminate redundant syntactic material. This process resembles the LF analog of Nunes' (2004) Reduce Chain. While Sportiche makes no comparison directly to Reduce Chain, he does suggest that Neglect is a general interface operation that operates in the same way at PF as it does at LF for determining when copies are pronounced. Sportiche points out that such an operation is independently motivated by the need to eliminate uninterpretable features from structures. He also points out that such an operation is implicit in Fox's (2002) trace conversion operation, which eliminates semantic material in lower copies of moved objects. Sportiche's definition of the operation Neglect is shown in (16).

(16) **Neglect:** Any material at any interface can be ignored up to crash

In Sportiche's Neglect theory, this operation is restricted from unbounded application by Full Interpretation, as seen in (17).

(17) **Full Interpretation:** Interpret every syntactic object

Sportiche states that the requirement that every syntactic object be interpreted refers to all the members of a chain collectively, not the individual components of a chain. This means that portions of a chain can be pronounced in one position and portions of a chain can be pronounced in another position, as long as all of them are pronounced somewhere. This leads to the operation neglecting uninterpretable features as long as there is an interpretable version that is left in the structure that remains to be interpreted.

Among the other consequences of such a theory, Sportiche provides a new account anti-reconstruction effects and other effects that have been previously accounted for with late adjunction. He proposes that these effects can instead be accounted for in a derivation where all the phrases are merged cyclically and Neglect can account for the lack of interpretation of certain structures in lower positions. In order to illustrate the idea, I will show Sportiche's treatment of the reconstruction and anti-reconstruction effects in (1b) and (2a). Since both structures are built up cyclically, I will give the entire structure of each sentence in (18).⁹

(18) a. *[Which picture of Steve_i] did he_i prefer [which picture of Steve_i]?

b. [Which picture that Steve_i took] did he_i prefer [which picture that Steve_i took]?

Given these structures in (18), the question must be what causes the structure of (18a) to be ungrammatical, but still allows for (18b) to be constructed and remain grammatical.

Sportiche argues, following Kayne (2010), that *of Steve* is not actually a complement of the noun, but in fact represents a relative clause that is attached. That relative clause is headed by *of*,

⁹ For simplicity, intermediate movements that are irrelevant have been omitted.

and contains within it a small clause that consists of *Steve* and the predicate *picture*. In adopting this analysis, Sportiche rejects the argument/adjunct distinction typically assigned to the contrast in (18) and reframes it so that it is the status of *picture* as a predicate in (18a) that drives the ungrammaticality, whereas in (18b) *picture* is an argument of *took*. In his theory of Neglect, arguments are capable of being partially reconstructed, but predicates must be totally reconstructed in order for their arguments to saturate the predicate (Huang 1982). This means that Neglect cannot act on the lowest copy of the predicate *picture* at LF in (18a), since it is the predicate of the small clause *Steve picture*. I give a more complete structure below in (19) that shows the small clauses and the copies these entail.

- (19) *[[Which picture of [Steve_i picture] did he_i prefer [which picture of [Steve_i picture]]]]?

Since the lowest copy of *picture* must be interpreted, it must be the case that *Steve* satisfies that predicate in the same position. Therefore, *Steve* must be interpreted in the lowest position in (19), resulting in a Principle C violation.

The structure of (18b) is different, however. Since *picture* is only an argument of the verb *took*, it can partially reconstruct. It is also possible for the whole adjunct to be neglected in the low position, since it has no predicate moved out (like (18a)) and also is not itself an argument, which would need to saturate its predicate locally. As a result, Neglect can apply to the lower copy of the relative clause, which obviates the Principle C effect.

If we assume that cyclic merger is driven by restricting merger to the root node of a structure, then late adjunction poses a serious conceptual problem. Sportiche's (2016) Neglect account of late adjunction effects is a creative and intriguing solution to the conceptual problem

of late adjunction. I will be adopting Sportiche's notion of this operation. However, in the following section I will show that a raising analysis of relative clauses where the head noun must be base generated in the relative clause cannot be correct.

5.4.1 Pseudo-complements and reconstruction

In previous theories, the reconstruction facts presented in the sentences in (20) have been construed as evidence for an argument/adjunct distinction between the grammatical and ungrammatical sentences.

- (20) a. *Which student of linguistics_i did it_i drive crazy?
b. *Which picture of Steve_i did he_i hate?
c. Which child near the teacher_i did she_i reprimand?
d. Which book that Steve_i took to the library did he_i hate?

We see in (20a-b) that *of linguistics* and *of Steve* must reconstruct causing a Principle C violation. In (20c-d), there is no reconstruction due to late adjunction and thus no Principle C violation. Based on this, we might assume that (20a-b) are complements and (20c-d) are adjuncts.

However, this distinction does not hold up when other tests are applied to the same sentences. Another diagnostic we can use is *one* substitution (Jackendoff 1977 and Baker 1978). NPs can be replaced by *one* while leaving the rest of the DP as shown in (21).

- (21) a. *The student of linguistics was taller than the one of physics
b. The picture of Steve was bigger than the one of John

c. The man from France is taller than the one from England

d. The man that saw Bill was taller than the one that saw Mary

For example, in (21a) we see that *student* is not able to be replaced by *one* in this case because *of linguistics* is an obligatory part of that NP. We say, then, that *of linguistics* is a complement of *student*. However, we see in (21b) that we can replace *picture* with *one* which indicates that *of Steve* must be an adjunct to the NP *picture*. We see similar effects in (21b-d). The picture phrases are behaving in a strange way, sometimes patterning with complements as in (20) while other times patterning with adjuncts as in (21).

In addition to *one* substitution, we would expect that complements must be in particular syntactic positions due to their status as complements. We see this inflexibility in (22).

(22) a. The student of linguistics who was tall aced the class

b. *The student who was tall of linguistics aced the class

When *of linguistics* appears after the relative clause, the sentence becomes ungrammatical. This is expected since complements must be to the left of adjuncts due to the latter's higher attachment site. On the other hand, we see in (23) that adjuncts are able to be interchanged with the relative clause.

(23) a. The man from France who was tall aced the class

b. The man who was tall from France aced the class

While it could be the case that *from France* is in fact inside the relative clause instead of adjoined above it, it is still clear that this behavior is not possible with complements. This

follows from the selectional requirements of the noun. If a phrase such as *of linguistics* as in (22) is the complement of the head noun, then it must be immediately merged into the derivation. This precludes the possibility that *from France* was absent inside of the relative clause. We could imagine a scenario in which you could have the complement in both copies, but the head noun is in one position, but the complement is in another, but we can see from the data in (22) that this is not an option in this case.¹⁰ On the other hand, in (23), the adjunct is either adjoined to the NP that includes the relative clause, or to the NP within the relative clause. Regardless of which analysis is correct, this arrangement is possible with adjuncts as in (23), but not with complements as in (22).

Since the adjuncts can attach in any order, this result further reinforces the status of *from France* as an adjunct. Once again we see *picture* phrases behaving strangely if they are to be considered complements. They seem to be able to change order with the relative clause in (24).

- (24) a. The picture of Steve that Bill liked was large
b. The picture that Bill liked of Steve was large

This evidence, in conjunction with the evidence in (22), leads to the conclusion that these phrases must indeed be adjuncts. These adjuncts do behave strangely with respect to reconstruction (see (20)), however. Because of this, I will refer to them as “pseudo-complements.”

We saw in (20) that pseudo-complements have peculiar properties with regards to reconstruction, but we see in (25) they once again pattern with other adjuncts.

¹⁰ Though this is not always true: See Hulsey and Sauerland (2006) and their discussion of extraposition.

- (25) a. *Which student of linguistics_i that it_i was hard for dropped the class?
- b. Which picture of Steve_i that he_i preferred was used in the yearbook?
- c. Which man with a pit bull_i who it_i bit was sent to the hospital?
- d. Which book that Steve_i took to the library that he_i didn't like was very heavy?
- e. *Which student of linguistics_i did it_i drive crazy?
- f. *Which picture of Steve_i did he_i hate?

In (25a) we see the complement *of linguistics* reconstructing into the relative clause, causing a Principle C violation. In (25c) and (25d), we see adjuncts that are able to avoid the Principle C effects by presumably by Neglect being able to target the lower copies of these adjuncts. This is strange when considering this alongside the evidence in (22) (repeated as (25e-f) however. What we know for sure in (25f) is that *of Steve* must reconstruct into the object of the matrix clause due to the ungrammaticality caused by the Principle C effect. Sportiche (2016) argues that this is evidence that pseudo-complements must reconstruct to the lowest possible position. However, these behave differently from true complements, which we can see in (25e) and (25a). In these sentences, *of linguistics* is obligatorily reconstructed in both cases because *of linguistics* is obligatorily interpreted in the lowest position, be it within the relative clause or in the object of the matrix VP. From this data, we can see a distinction between the reconstruction patterns of true noun complements and pseudo-complements. Complements obligatorily reconstruct, both into the matrix clause (25e) and into a relative clause (25a). On the other hand, pseudo-complements must reconstruct into the matrix clause (25f), but this is not true for reconstruction into a relative clause (25b). So while Sportiche's insight into the reconstruction

of predicates into pseudo-complements is insightful, it does not capture the whole story. There must be an explanation for why *of Steve* is capable of avoiding reconstruction in (25b), but is not capable of avoiding reconstruction in (25f). Since we cannot assume that *of Steve* is a complement from its lack of reconstruction in (25b), as well as other evidence that shows it patterns with adjuncts, we are left to assume that these pseudo-complements are in fact adjuncts.

5.5 Implications for deriving relative clauses

Late adjunction, as an operation that can violate the Extension Condition, should be considered problematic to any theory of the syntax. Sportiche's (2016) theory of Neglect does the important work of replacing late adjunction as an account of anti-reconstruction facts and other data that has traditionally relied on late adjunction. Since Neglect can be independently motivated, and provides for a more principled theory than late adjunction, I will be moving forward under the assumption that late adjunction is not available to the computational system. However, the data in (25) show that Sportiche's theory of relative clauses is incomplete. Since his theory implies a raising analysis of relative clauses, he should predict the obligatory reconstruction of pseudo-complements into the relative clause. We can see in (25b) that this is not the case. We can also see from (25a) that complements do obligatorily reconstruct into the relative clause. This means that there are two patterns to describe. Also, since we find instance like (25a) of obligatory reconstruction into relative clause, there cannot be some exception built into the theory for Neglect within relative clauses. This demands a reworking of our understanding of how relative clauses are constructed.

One crucial difference between complements and pseudo-complements is the nature of the structure they create. We know from the tests in section 5.4.1 that the head noun of a

pseudo-complement is itself an independent NP, whereas in a complement structure, it is not. This provides us with some leverage in the syntax to find a distinction in their derivation. The difference suggests that the NP head noun in pseudo-complement constructions might be able to move independently of its pseudo-complement, while in (25a), a head noun cannot move independently of its head noun. This difference is unable to be leveraged if we believe that both the pseudo-complement and the regular complement both are attached to the head noun within a relative clause. This is the prediction of Sportiche's raising analysis of relative clauses. However, the NP/N distinction can be leveraged if there is movement into the relative clause through sideward movement.

I propose that relative clauses are not created by raising the head noun from within the relative clause, but that the head noun is base generated in the head noun position and is later moved into the relative clause via sideward movement. This proposal is a departure from the assumptions by Nunes (2004, 2012) and Chomsky (2004), along with many others who assume that the derivation and movement must proceed from more embedded contexts to less embedded contexts. Recall that the theory outlined in Chapter 3 accounts for the directionality of movement in parasitic gap constructions by spelling out original copies of a moved object. This theory makes no inherent predictions about the directionality of movement. So instead of being a departure from a previous prediction, movement into a more embedded environment fills a conceptual gap, if indeed such a structure can be made and converges. In order to illustrate the point, I will first show the derivation of a typical relative clause in (26).

(26) The man who John hit

a. Head noun introduced into derivation

K = [NP man]

b. Relative D introduced

K = [NP man]

L = who

c. Sideward movement of *man* into relative DP

K = [NP man]

M = [DP who man]

d. Relative vP built up

K = [NP man]

N = [vP John hit [DP who man]]

e. Movement of relative DP to vP edge

K = [NP man]

O = [vP [DP who man][vP John hit [DP who man]]]

f. Relative clause C introduced

K = [NP man]

P = [C' C [TP John [vP [DP who man][vP John hit [DP who man]]]]

g. *Who man* moved to the specifier of the relative CP

K = [NP man]

Q = [CP [DP who man] [TP John [vP [DP who man][vP John hit [DP who man]]]]

h. Relative clause adjoined to head noun

R = [NP [NP man] [CP [DP who man] [TP John [vP [DP who man][vP John hit
[DP who man]]]]]

i. D merged to NP

S = [DP the [NP [NP man] [CP [DP who man] [TP John [vP [DP who man][vP John
hit [DP who man]]]]]]]

In (26a), we can see that the head noun is introduced into the derivation. Since it has no selectional requirements (it is a maximal projection), the NP *man* can be targeted for movement. In (26b) the relative determiner is introduced into the derivational workspace. Much as DPs can be moved from the parasitic domain into the matrix domain in order to satisfy the selectional requirements of the matrix verb, the head noun is moved into the relative clause to satisfy the selectional requirements of the relative determiner in (26c). This is similar to Bianchi's (2000) analysis of the role of the relative determiner in raising structures. Bianchi argues that the head noun within the relative clause is simply an NP, and the relative determiner selects for that NP. It then moves up through the relative clause. Bianchi argues that the NP, in this high position within the relative clause can also check the selectional features of the head determiner. What I am proposing is that the NP begins in the structure that will eventually be the larger DP, and then moves into relative DP in order to satisfy those selectional features.

One of the notable things about building the structure this way is its contrast with how parasitic gaps are constructed. In parasitic gap constructions, one of the key problems was understanding why there was movement within the parasitic domain. If objects were allowed to move into the parasitic domain, then there would be no motivation for movement within the parasitic domain, and accounting for island effects within the parasitic domain would be more difficult. However, when we assume that movement came out of the parasitic gap, movement could be motivated through a necessity to check the WH feature of the moved object, which could not be checked in the parasitic domain. In order to do so, it needed to move up to the ‘escape hatch’ at the edge of the parasitic domain in order to be extracted to the matrix domain to be checked. The motivation to escape is not necessary within relative clause structures, since movement can be motivated by the unchecked features of the relative determiner. Since relative clauses have a determiner separate from the head determiner of the larger DP, that determiner can have properties independent of the noun that merges with them. This is different from the case of parasitic gaps, where the DP in the parasitic domain is precisely the same DP in the matrix clause. Therefore, relative determiners can have their own unchecked features that must be checked through movement within the relative clause. In this way, movement in the relative clause is motivated by the requirements of the relative determiner, not the NP that has moved to into the relative clause. There is no need to motivate the movement through an escape hatch to the matrix domain. Therefore, island effects are the result of the movement motivated by the relative DP reaching the specifier of the relative CP.

After the sideward movement in (26c), the relative clause is built up in (26d-g), with the relative DP being moved cyclically through the relative clause in order to be checked in the relative CP. In (26h), the relative clause is merged to the head noun. This step requires some

explanation, since the theory we constructed for parasitic gap constructions and chain formation makes some predictions about the nature of this structure. Since the NP *man* moved into the relative clause in (26c), this should spell out the head noun. I propose that this still leaves a structural residue of the spelled out material in the derivational workspace that still allows for relationships to be made with that residue. This residue may be the label, or whatever projection still remains in order to satisfy the requirements of the determiner that selects the combined NP in (26i). Another crucial aspect of the theory created for parasitic gaps was the mechanism that creates chains. It must be the case that the head NP c-commands into the relative clause. In this case, it is not problematic or controversial, but this will become important later.

Interestingly, once the chain between the head noun and the noun within the relative DP is made, it is not the most recently made copy, but the higher copy in terms of c-command that is pronounced. This is a credit to Nunes' formulation of Form Chain, since it accurately predicts the pronunciation of the high copy. Other theories that rely on a movement history might predict the pronunciation of the highest copy within the relative clause since the movement history would depict that copy as the most recently made.

This account of the derivation of relative clauses also allows us to account for the differences between complements and pseudo-complements with regard to their reconstruction into relative clauses. First, let's examine why complements must obligatorily reconstruct into the relative clause by looking at the relevant portions of the derivation in (27).

(27) *the student of linguistics_i that it_i was hard for

a. Head NP constructed

K = [_{NP} [_N student] of linguistics]

b. Relative D introduced

$K = [_{NP} [_{N} \text{student}] \text{ of linguistics}]$

$L = \emptyset_{\text{rel}}$

c. Sideward movement of *student of linguistics*

$K = [_{NP} [_{N} \text{student}] \text{ of linguistics}]$

$M = [_{DP} \emptyset_{\text{rel}} [_{NP} [_{N} \text{student}] \text{ of linguistics}]]$

d. Construction of the relative TP

$K = [_{NP} [_{N} \text{student}] \text{ of linguistics}]$

$M = [_{TP} \text{it was hard for } [_{DP} \emptyset_{\text{rel}} [_{NP} [_{N} \text{student}] \text{ of linguistics}]]]$

In (27a) we see the noun phrase containing the head noun is built up. Since *of linguistics* is a true complement to the noun, *student* is not a noun phrase without it. In (26b), the relative determiner (in this case an unpronounced determiner), is introduced. In (26c), the entire NP *student of linguistics* is moved into the relative clause. Since *student* is not a NP on its own, it cannot be targeted for movement. It is only the whole NP, which includes *of linguistics* that can be moved. Sportiche's theory states that complements must be interpreted in the lowest positions for local predicate satisfaction. This means that *linguistics* must be interpreted in a position where it is c-commanded by *it*. This will result in a Principle C violation, which will cause a crash.

Pseudo-complements are not obligatorily reconstructed into the relative clause because they are never in the clause to begin with. Before I demonstrate this with a derivation, I must

reiterate that following Kayne (2010) and Sportiche (2016), I am assuming that pseudo-complements are a type of relative clause. As a result, it should be the case that their construction mirrors that of other relative clauses. We can see this in the beginning of the derivation in (28).

(28) The picture of Steve_i that he_i hated

a. Introduction of the head noun

$$K = [_{NP} \text{ picture}]$$

b. Introduction of the small clause head

$$K = [_{NP} \text{ picture}]$$

$$L = \emptyset_{SC}$$

c. Sideward movement of *picture* into the small clause

$$K = [_{NP} \text{ picture}]$$

$$M = [_{SC} \text{ Steve } \emptyset_{SC} [_{NP} \text{ picture}]]$$

d. Pseudo-complement constructed

$$K = [_{NP} \text{ picture}]$$

$$M = [_{CP} \text{ of } [_{SC} \text{ Steve } [_{NP} \text{ picture}]]]$$

e. Pseudo-complement adjoined to head noun

$$N = [_{NP} [_{NP} \text{ picture}] [_{CP} \text{ of } [_{SC} \text{ Steve } [_{NP} \text{ picture}]]]]$$

f. Relative determiner introduced into the derivation

N = [NP [NP ~~picture~~] [CP of [SC Steve [NP picture]]]]

O = \emptyset_{rel}

g. Sideward movement of *picture* into the relative DP

N = [NP [NP ~~picture~~] [CP of [SC Steve [NP picture]]]]

P = [DP \emptyset_{rel} [NP picture]]

h. Relative clause constructed

N = [NP [NP ~~picture~~] [CP of [SC Steve [NP ~~picture~~]]]]

P = [CP [DP \emptyset_{rel} [NP picture]] that [TP he [VP [~~DP \emptyset_{rel} [NP picture]]] [VP he hated
[~~DP \emptyset_{rel} [NP picture]]]]]]]]~~~~

i. Relative clause merged to head noun

Q = [NP [NP [NP ~~picture~~] [CP of [SC Steve [NP ~~picture~~]]] [CP [DP \emptyset_{rel} [NP
picture]] that [TP he [VP [~~DP \emptyset_{rel} [NP picture]]] [VP he hated [~~DP \emptyset_{rel} [NP
picture]]]]]]]]]]~~~~

j. Merger of D

R = [DP the [NP [NP [NP ~~picture~~] [CP of [SC Steve [NP ~~picture~~]]] [CP [DP \emptyset_{rel} [NP
picture]] that [TP he [VP [~~DP \emptyset_{rel} [NP picture]]] [VP he hated [~~DP \emptyset_{rel} [NP
picture]]]]]]]]]]~~~~

In this relative clause structure, we first begin with the head noun. Since it has no selectional requirements, it is an NP on its own in (28a). Next, the small clause head is introduced into the derivation in (28b). This allows the NP *picture* to be copied and moved into the small clause in (28c). In this derivation, I have shown more explicitly how this will cause the Spell-out of the head noun. This is in order to demonstrate that we can build up several relative clauses (be they pseudo-complements or typical relative clauses) with the same head noun. In (28d), the rest of the small clause and its complementizer (*of*, following Kayne 2010) is built up in the derivational workspace. I am agnostic as to whether there is movement within the small clause and about whether there are phases within pseudo-complements. This theory does predict that if there are phases within the small clause that *picture* would be able to move up within the relative clause to escape. In (28e) the pseudo-complement is adjoined to the remnant of the head noun. (28f) marks the beginning of the larger relative clause as the relative determiner is introduced. In (28g), *picture* is copied from the pseudo-complement to merge with the relative determiner. From here, the rest of the relative clause is built up in (28h). Lastly, the relative clause is adjoined to the head noun as well in (28i), and then the determiner is merged to the structure in (28j).

This final structure requires that some assumptions be made about the relationship between the head noun and its adjuncts. In order for Form Chain to make a chain out of all of the relevant copies of *picture*, the head noun copy must c-command all of the other copies within the two relative clauses. If we assume that a complete phrase c-commands all of its adjuncts, then this is not a problem, even if there are intervening adjuncts within the structure. We must be careful, however. It cannot be the case that different adjuncts c-command each other. This is crucial, since in parasitic gap structures, the copy of the WH phrase that is adjoined to the vP

cannot c-command into the parasitic domain. If it did c-command into the parasitic domain, it would permit a variety of problematic parasitic gap structures that were ruled out in Chapter 3. If this claim is sustainable, chains can be formed between the head noun and all of the copies within its adjuncts. Since it is not in a checking position, the restrictions on Form Chain regarding interveners should not apply. This might be especially relevant in pseudo-complements, where other portions of the small clause might contain features that could intervene for something targeting nouns.

5.5.1 Accounting for islands within the relative clause

Since island effects are also found within relative clauses, it is necessary to show that these can be derived as well. These island effects can be found in the sentences in (29).

- (29) a. *I saw the man_i that John asked whether *t_i* left
 b. *I saw the man_i that pictures of *t_i* embarrassed Suzy
 c. *I saw the man_i that John left after *t_i* arrived.

These sentences reflect the presence of WH islands (29a), subject islands (29b), and adjunct islands (29c) within the relative clause. We can see the problematic portion of the derivation of (29a) below in (30).

- (30) a. Head noun introduced

K = [NP man]

- b. Relative determiner introduced

K = [NP man]

$$L = \emptyset_{\text{rel}}$$

c. Sideward movement of *man* into relative DP

$$K = [\text{NP } \mathbf{man}]$$

$$M = [\text{DP } \emptyset_{\text{rel}} \mathbf{man}]$$

d. Embedded CP built up

$$K = [\text{NP } \mathbf{man}]$$

$$N = [\text{CP } \mathbf{whether} [\text{TP } [\text{DP } \emptyset_{\text{rel}} \mathbf{man}] \mathbf{arrived}]]$$

e. VP constructed

$$K = [\text{NP } \mathbf{man}]$$

$$P = [\text{VP } \mathbf{asked} [\text{CP } \mathbf{whether} [\text{TP } [\text{DP } \emptyset_{\text{rel}} \mathbf{man}] \mathbf{arrived}]]]$$

f. *v* introduced to relative clause, Spell-out of TP

$$K = [\text{NP } \mathbf{man}]$$

$$Q = [\text{v}' \mathbf{v} [\text{VP } \mathbf{asked} [\text{CP } \mathbf{whether} [\text{TP } [\text{DP } \emptyset_{\text{rel}} \mathbf{man}] \mathbf{arrived}]]]]$$

In (30a) we see the head noun introduced, and in (30b) the relative determiner is introduced into the derivation. In (30c) the head noun moves into the relative DP. In (30d) the rest of the CP is built up, including *whether* filling the specifier of CP. Since this blocks the movement of the relative DP, it remains in the specifier of TP. Then the relative VP is built up in (30e). When the *v* is introduced into the relative clause in (30f), this triggers the Spell-out of the complement of CP. This is the TP that includes the relative DP. Since there is no longer a

copy of the relative DP, it cannot check its relative feature in the higher CP, and the derivation will crash.

We can see that something similar happens to the derivation of (29b), shown in (31).

(31) a. Head noun introduced

$K = [_{NP} \text{man}]$

b. Relative determiner introduced

$K = [_{NP} \text{man}]$

$L = \emptyset_{rel}$

c. Sideward movement of *man* into relative DP

$K = [_{NP} \text{man}]$

$M = [_{DP} \emptyset_{rel} \text{man}]$

d. Relative vP built up

$K = [_{NP} \text{man}]$

$N = [_{vP} \text{pictures of } [_{DP} \emptyset_{rel} \text{man}] \text{ embarrassed Suzy}]$

e. Relative T merged

$K = [_{NP} \text{man}]$

$O = [_{T'} T [_{vP} [_{DP} \emptyset \text{ pictures of } [_{DP} \emptyset_{rel} \text{man}]] \text{ embarrassed Suzy}]$

f. Movement of relative DP, Spell-out related to movement

K = [NP ~~man~~]

P = [TP [VP [DP \emptyset pictures of [DP \emptyset_{rel} man]]] T [VP [DP \emptyset pictures of [DP \emptyset_{rel} ~~man~~]] embarrassed Suzy]

Once again, the head noun and relative determiner are introduced into the derivation in (31a) and (31b) respectively, and the head noun moves into the relative DP in (31c). In (31d), we can see the relative vP is built up, and the relative DP is contained within the unmoved subject. The relative T is introduced into the derivation in (31e), and this triggers the movement of the subject in (31f). We can see that this causes the Spell-out of the entire subject in the low copy. In addition, since movement causes the Spell-out of the complement of the moved object, the NP *picture of man* is spelled out as well. This eliminates all copies of the relative DP from the derivational workspace. Therefore, the relative D cannot check its features in the specifier of CP and the derivation will eventually crash.

The derivation of adjunct islands like (29c) proceeds similarly in (32).

(32) a. Head noun introduced

K = [NP man]

b. Relative determiner introduced

K = [NP man]

L = \emptyset_{rel}

c. Sideward movement of *man* into relative DP

K = [NP ~~man~~]

M = [DP \emptyset_{rel} man]

d. Relative adjunct constructed

K = [NP ~~man~~]

N = [PP after [DP \emptyset_{rel} man] arrived]

e. Relative vP constructed

K = [NP ~~man~~]

N = [PP after [DP \emptyset_{rel} man] arrived]

O = [vP John left]

f. Relative PP adjoins to vP

K = [NP ~~man~~]

P = [vP [vP John left] [PP after [DP \emptyset_{rel} man] arrived]]

g. Relative TP constructed

K = [NP ~~man~~]

Q = [TP John [vP [vP ~~John~~ left] [PP after [DP \emptyset_{rel} man] arrived]]

h. C merges into relative clause, Spell-out of VP and adjunct CP

K = [NP ~~man~~]

Q = [C' C [TP John [vP [vP ~~John~~ left] [PP after [DP \emptyset_{rel} man] arrived]]]

Again, the head noun is introduced in (32a), followed by the relative determiner in (32b). The head noun moved into the relative DP in (32c). The adjunct that the relative DP is part of is built up in (32d). Then the derivation moves on to build the relative vP in (32e). Since there is no position to sideward move the relative DP into, the relative DP remains within the adjunct, which is then adjoined to the vP in (32f). The relative TP is built up in (32g), and finally the relative C is introduced in (32h). The introduction of the phase-head C into the structure causes all of the complements of lower phase heads to spell out. Notably, this includes the complement of *after*, which includes the relative DP. We can see in (32h) that there are no remaining copies of the relative DP, and it cannot check its uncheck features, and the derivation will crash.

5.5.2 Summarizing relative clause structure

This section has shown us how relative clauses themselves must be constructed. We know that Sauerland and Hulsev's (2006) dual structure approach cannot be correct due to the existence of relative clauses that display both matching and raising effects. Nunes' (2004) account of sideward movement demands movement out of the relative clause into a vacuum, with no motivating syntactic objects. This theory also obscures the relationship between the DP that takes a thematic role and the object that we find in the specifier of CP. Henderson's (2007) sideward movement theory preserves the relationship between the thematic position a DP moves from and the CP position it moves into, but relies on late adjunction. Lastly, Sportiche's account of effects that are traditionally analyzed as late adjunction cannot straightforwardly account for the reconstruction asymmetries between pseudo-complements and true complements of the head noun. The theory I have proposed relies on sideward movement in order to leverage the structural asymmetries between complements and pseudo-complements and provide an account for the reconstruction asymmetries in relative clauses. In the structures discussed in this section,

I have assumed that all of the copied material can be interpreted in any position. In the next section, I will show that the present theory can use Neglect to account for raising effects, matching effects, as well as relative clauses in which both occur within the same structure.

5.6 Using Neglect for raising and matching effects

In order for the present theory of relative clauses to successfully account for relative clauses, it must be able to account for sentences with raising effects (such as idioms and Principle A reconstruction) as well as matching effects, which is to say anti-reconstruction effects. As Henderson (2007) points out, a complete theory of relative clauses must be able to account for structures that display raising effects at the same time as matching effects. This can all be done through a combination of sideward movement and the Neglect operation on the final structure.

In the previous section, I showed that the smallest NPs can be targeted for movement in the case of pseudo-complements. This does not always have to be the case. This can be seen when we look at raising effects in relative clauses. We can see three types of raising effects in (33).

- (33) a. The picture of himself_i that John_i hated scared the children.
b. The headway that John made impressed his boss.
c. The picture of his_i mother that [every boy]_i drew was displayed on the wall.

We see in these sentences examples of Principle A reconstruction (33a), idiom reconstruction (33b), and variable binding (33c). For the moment, let's focus on those that contain pseudo-complements, since the derivation of (33b) is trivial within the current theory. In

(33a) and (33c), we can see that the crucially reconstructed element is found within a pseudo-complement. In the last sections I showed that pseudo-complements do not need to be copied into the relative clause. This is only an option, however, and we can see that the pseudo-complements do reconstruct in (33a) and (33c).

In order to provide an example of how this theory would account for such effects, I will present the derivation of the relative clause from (33a) below in (34).

(34) The picture of himself_i that John_i hated

a. Head noun introduced

$K = [_{NP} \text{ picture}]$

b. Small clause head introduced

$K = [_{NP} \text{ picture}]$

$L = \emptyset_{SC}$

c. Sideward movement into small clause

$K = [_{NP} \text{ picture}]$

$M = [\emptyset_{SC} [_{NP} \text{ picture}]]$

d. Pseudo-complement built up

$K = [_{NP} \text{ picture}]$

$N = [_{CP} \text{ of } [\text{himself } \emptyset_{SC} [_{NP} \text{ picture}]]]$

e. Pseudo-complement adjoined to head noun

$$O = [_{NP} [_{NP} \text{ picture}] [_{CP} \text{ of } [_{\text{himself } \emptyset_{SC}} [_{NP} \text{ picture}]]]]$$

f. Relative D introduced

$$O = [_{NP} [_{NP} \text{ picture}] [_{CP} \text{ of } [_{\text{himself } \emptyset_{SC}} [_{NP} \text{ picture}]]]]$$
$$P = \emptyset_{rel}$$

g. Sideward movement of *picture of himself* into relative DP

$$O = [_{NP} [_{NP} \text{ picture}] [_{CP} \text{ of } [_{\text{himself } \emptyset_{SC}} [_{NP} \text{ picture}]]]]$$
$$Q = [_{DP} \emptyset_{rel} [_{NP} [_{NP} \text{ picture}] [_{CP} \text{ of } [_{\text{himself } \emptyset_{SC}} [_{NP} \text{ picture}]]]]]$$

h. Construction of relative CP

$$O = [_{NP} [_{NP} \text{ picture}] [_{CP} \text{ of } [_{\text{himself } \emptyset_{SC}} [_{NP} \text{ picture}]]]]$$
$$R = [_{CP} [_{DP} \emptyset_{rel} [_{NP} [_{NP} \text{ picture}] [_{CP} \text{ of } [_{\text{himself } \emptyset_{SC}} [_{NP} \text{ picture}]]]]] \text{ that } [_{TP}$$
$$\text{John hated } [_{DP} \emptyset_{rel} [_{NP} [_{NP} \text{ picture}] [_{CP} \text{ of } [_{\text{himself } \emptyset_{SC}} [_{NP} \text{ picture}]]]]]]]$$

i. Relative clause adjoins to head noun

$$S = [_{NP} [_{NP} [_{NP} \text{ picture}] [_{CP} \text{ of } [_{\text{himself } \emptyset_{SC}} [_{NP} \text{ picture}]]]] [_{CP} [_{DP} \emptyset_{rel} [_{NP} [_{NP}$$
$$\text{picture}] [_{CP} \text{ of } [_{\text{himself } \emptyset_{SC}} [_{NP} \text{ picture}]]]]] \text{ that } [_{TP} \text{John hated } [_{DP} \emptyset_{rel} [_{NP}$$
$$[_{NP} \text{ picture}] [_{CP} \text{ of } [_{\text{himself } \emptyset_{SC}} [_{NP} \text{ picture}]]]]]]]$$

j. D merged

T = [_{DP} the [_{NP} [_{NP} [_{NP} picture] [_{CP} of [himself \emptyset_{SC} [_{NP} picture]]]] [_{CP} [_{DP} \emptyset_{rel}
 [_{NP} [_{NP} picture] [_{CP} of [himself \emptyset_{SC} [_{NP} picture]]]]] that [_{TP} John hated [_{DP}
 \emptyset_{rel} [_{NP} [_{NP} picture] [_{CP} of [himself \emptyset_{SC} [_{NP} picture]]]]]]]]]]]

The beginning of this derivation resembles the other derivation of a pseudo-complement we saw in (28). This begins with the head noun being built up in (34a), and then moved to fill in the small clause in (34c). In (34d) the small clause is completed, and then adjoined to the head noun in (34e). While in (28) I showed that the smallest NP can be selected for movement, it is also ok for the larger NP, *picture of himself*, to be moved, which is what we see in (34g). From here, the relative clause is built up in (34h). Then it is adjoined to the head noun in (34i) and the determiner is merged in (34j). The ability to move the larger of the two NPs in (34g) is what allows *himself* to end up in the scope of *John* to be bound and satisfy Principle A. This optionality is crucial to the theory. This also implies that the syntactic object *picture of himself* in the head noun can c-command into the relative clause in the same way that just *picture* could in (28). While one adjunct cannot c-command into the other, a phrase and its adjunct can c-command into a second adjunct. This allows a chain to be formed for later reduction.

Matching effects can be accounted for straightforwardly simply using Neglect. This is shown in the derivation for (35) below (the relative clause is abbreviated in (35g-j) for readability)

(35) Which man that John hit did he despise?

a. Head noun introduced

K = [_{NP} man]

b. Relative determiner introduced

K = [NP man]

L = \emptyset_{rel}

c. Sideward movement of head noun into relative DP

K = [NP man]

M = [DP \emptyset_{rel} man]

d. Relative clause constructed

K = [NP man]

Q = [CP [DP \emptyset_{rel} man] that [TP John [vP [DP \emptyset_{rel} man][vP John hit [DP \emptyset_{rel} man]]]]

e. Relative clause adjoined to the head noun

R = [NP [NP man] [CP [DP \emptyset_{rel} man] that [TP John [vP [DP \emptyset_{rel} man][vP John hit [DP \emptyset_{rel} man]]]]]

f. *Which* merged to NP

S = [DP which [NP [NP man] [CP [DP \emptyset_{rel} man] that [TP John [vP [DP \emptyset_{rel} man][vP John hit [DP \emptyset_{rel} man]]]]]]]

g. Matrix vP constructed

T = [vP he despise [DP which man [RC]]]

h. WH phrase moves to vP edge

$$U = [_{vP} [_{DP} \text{which man } [RC]]] [_{vP} \text{he despise } [_{DP} \text{which man } [RC]]]$$

i. Construction of matrix clause, introduction of C

$$V = [C' \text{did } [_{TP} \text{he did } [_{vP} [_{DP} \text{which man } [RC]]] [_{vP} \text{he despise } [_{DP} \text{which man } [RC]]]]]$$

j. WH phrase moves to specifier of CP

$$W = [_{CP} [_{DP} \text{which man } [RC]]] \text{did } [_{TP} \text{he did } [_{vP} [_{DP} \text{which man } [RC]]] [_{vP} \text{he despise } [_{DP} \text{which man } [RC]]]]]$$

Like all of our other relative clauses, the head noun is introduced into the derivation first (35a) and then after moving the head noun into the relative DP (35c), the relative clause is built up (35d), merged to the head noun (35e), and then the determiner (in this case *which*) is merged into the derivation (35f). Then the matrix clause is built up as the WH cyclically moves to the matrix CP (35 g-j). There are two different chains that must be accounted for in this structure. The WH chain is a chain of three DPs that are all formed into one chain for later reduction. Separate from these is the NP chain of *man* that must also be reduced. There is a third chain that is the relative DP also. Each of these chains is separately accounted for with Form Chain, which allows for Chain Reduction to pronounce each element in a different place relative to the chain under consideration. Both the head noun chain and the relative DP chain are pronounced high in the structure due to the matrix WH chain preventing the pronunciation of the lower copies in the matrix clause. Within this highest DP, the head noun is pronounced in the highest NP position. This is notably pronounced in a different position from the relative determiner. In this particular

sentence, there is no pronounced relative determiner, but in (26), *which* is pronounced within the relative clause.

While we should suspect that Reduce Chain is the PF form of Neglect, we know that the targets of Neglect at LF are quite different. In this case, the relative clause must be neglected in all but the highest position in the matrix clause in order to avoid a Principle C effect. This follows Sportiche's account of matching effects precisely, as adjuncts do not need to be interpreted in a low position to satisfy local predicates. As a result, as long as they are interpreted somewhere (in this case, the relative clause copy in the specifier of CP), then Full Interpretation can be satisfied.

Lastly, we must be sure that both raising and matching effects can be accounted for simultaneously. I will show how this theory can account for such a sentence with Henderson's (2007) example in (36). Once again, the relative clause will be abbreviated in (36g-j).

(36) What headway that John_i made did he_i later regret?

a. Head noun introduced

$K = [_{NP} \text{headway}]$

b. Relative determiner introduced

$K = [_{NP} \text{headway}]$

$L = \emptyset_{rel}$

c. Sideward movement of head noun into relative DP

$K = [_{NP} \text{headway}]$

M = [DP \emptyset_{rel} [NP headway]]

d. Relative clause constructed

K = [NP headway]

N = [CP [DP \emptyset_{rel} [NP headway]] that [TP John [vP [DP \emptyset_{rel} [NP headway]]][vP John made [DP \emptyset_{rel} [NP headway]]]]

e. Relative clause adjoined to head noun

O = [NP [NP headway] [CP [DP \emptyset_{rel} [NP headway]] that [TP John [vP [DP \emptyset_{rel} [NP headway]]][vP John made [DP \emptyset_{rel} [NP headway]]]]]

f. *What* merged

P = [DP what [NP [NP headway] [CP [DP \emptyset_{rel} [NP headway]] that [TP John [vP [DP \emptyset_{rel} [NP headway]]][vP John made [DP \emptyset_{rel} [NP headway]]]]]]]

g. Matrix vP constructed

Q = [vP later he regret [DP what headway [RC]]]

h. WH phrase moved to vP edge

R = [vP [DP what headway [RC]] [vP later he regret [DP what headway [RC]]]]]

i. C introduced

S = [C did [TP he did [vP [DP what headway [RC]] [vP later he regret [DP what headway [RC]]]]]]]

j. WH phrase moves to specifier of CP

T = [CP [DP what headway [RC]] did [TP he did [_{vP} [DP what headway [RC]]
[_{vP} later he regret [DP what headway [RC]]]]]]]

Much like our other derivations, the relative clause is built up through sideward movement from the head noun to combine with the relative determiner in (36c), and then the relative clause is built up (36d) and merged with the head noun (36e). Since *headway* is moved into the relative clause, it can be interpreted in that position and license the idiom. This ensures the raising effects are accounted for. Then the matrix clause is built up normally, with WH movement to the specifier of the matrix CP in (36g-j). Once again, Neglect can target all of the lower copies of the relative clause that might be c-commanded by *he*. This obviates the Principle C violation.

These derivations show that we can derive both relative clauses and pseudo-complements using sideward movement. Also, this theory accounts for the obligatory reconstruction of complements into relative clauses and in matrix clauses. It accounts for the obligatory reconstruction of pseudo-complements in the matrix clause, but not into relative clauses. It also accounts for raising effects and matching effects in relative clauses, as well as relative clauses which have both matching and raising effects. In addition, this theory is based in Sportiche's theory of the Neglect operation, which has even more explanatory breadth for reconstruction and anti-reconstruction facts, and does not rely on late adjunction.

5.7 Conclusion

The theory laid out in this chapter provides yet another domain in which we can use sideward movement to produce a more explanatory and deep theory of the syntax. Relative clauses are a complex phenomenon, and there are many more details in the literature that must eventually be incorporated into such a theory. However, this theory benefits greatly from being based on the same principles and operations that were developed elsewhere in this dissertation for a separate problem, that of parasitic gap and ATB constructions.

It also serves to fill in a gap produced by the account of parasitic gaps. The theory does not make an explicit prediction about the directionality of sideward movement, and so it would stand to reason that structures might exist where the movement could go from less embedded to more embedded environments under the right circumstances. That prediction is borne out in this theory of relative clauses. The combination of (i) Form Chain predicting that sideward movement must create chains that will eventually c-command each other and (ii) the derivation only being able to proceed in building structure cyclically, most sideward movement structures should indeed see movement from more embedded to less embedded environments. However, the right environment can allow for the opposite, as we have seen with relative clause.

This theory incorporates Sportiche's (2016) theory of Neglect, which allows for the effects of 'late adjunction' to be accounted for within a theory that strictly obeys the Extension Condition. This theory also incorporates the empirical coverage of Nunes (2004), Sauerland and Hulsey (2006), Henderson (2007), and Sportiche (2016). In addition, it expands the empirical coverage to account for the distinct reconstruction patterns of true nominal complements and pseudo-complements.

6 Conclusion

6.1 The theory as it stands

My goal in this dissertation was to take a set of principles that were as atomic and independently motivated as possible, and follow them to understand how they can be used to understand the computational system and its outputs. In this dissertation, I have followed the predictions of a minimal theory of Merge. As discussed in Chapter 2 and in many other places (Nunes 1995, 2001, 2004; Hornstein & Nunes 2002, 2008; and Hornstein 2001; and Boeckx et al. 2010), sideward movement seems to be a natural consequence of a minimal theory of Merge that allows for copies to be made from within a structure to extend a structure. Since Merge is composed of several smaller operations that can independently select their targets, there is no reason to assume a constraint that says the structure you make a copy from must be the one that is extended. As a result, sideward movement is predicted. On the other hand, it is easy to imagine how such an operation might be too powerful if left unrestricted.

In order to explore the consequences of this operation and how it might be restricted, I follow Nunes (2004, 2012) in using parasitic gap constructions as a domain in which to test the power of such an idea. Nunes' theory provides a great deal of insight into how a theory of parasitic gaps must be constructed. He quite successfully shows that much of the overgeneration that we might worry about when considering an operation like sideward movement is ruled out by many of the restrictions we have independently motivated within the grammar previously. However, as Haida and Repp (2012) points out, there are empirical holes in Nunes' account of parasitic gaps. Additionally, I presented empirical and conceptual problems with Nunes' account

of parasitic gaps, ranging from unaccounted for island effects within the parasitic domain to requiring that the derivation be able to look ahead to know what the structure will look like later.

One of the major problems that both Haida and Repp (2012) and I point out is the problem that Nunes' theory has accounting for island effects within the parasitic domain. Ultimately, both Haida and Repp (2012) and Nunes (2004, 2012) propose that there are special Spell-out operations that can apply to adjuncts in order to account for islands within the parasitic domain. Nunes also stipulates the necessity that movement proceeds from more embedded to less embedded environments, a claim which must rely on a knowledge about a structure's status later in the derivation. While these two theories of parasitic gaps are compelling their coverage, I show that the data they do not cover is a result of not following through on following basic, motivated operations to find answers to the problem at hand.

As I have shown in Chapters 2-4, this minimal theory of Merge does not exist within a vacuum, either conceptually or empirically. Empirically, we can see from the large array of previous research laid out in Chapter 2 that there are many restrictions on the resulting structures created from sideward movement. Conceptually, Merge exists within a larger linguistic mechanism, that has both language specific components and domain general components. I argued in Chapter 3 that these 'third factor' requirements on language motivate the use of phases and the cyclic application of Spell-out, which reduce the computational load. This is supported by the empirical evidence for cyclic movement and computation we see ubiquitously in language. These facts together support the existence of a phase architecture that acts on the totality of the computational system to reduce computational load. In Chapter 3 I explicitly explored the possibilities that exist for phasal architectures. I showed that some of these possibilities fail to allow for a cyclic computation. Among the remaining, motivated options, I

showed that only one of these possibilities would allow for the empirical coverage that was required to (i) account for a sideward movement theory of parasitic gaps and (ii) account for adjunct islands using a general, cyclic rule of phase Spell-out. This second point is important because it once again delivers on the promise that we can derive complex phenomena from a smaller set of atomic operations. This is simply the consequence of breaking down the theory to only its most essential parts and following those to their natural conclusion.

Another compelling aspect of this theory is that it brings parasitic gaps as evidence for the phasehood of prepositional phrases. Abels (2003) provides a compelling account for their phasehood in his account of preposition stranding. Additionally, van Riemsdijk (1979) provides an analysis of movement out of prepositional phrases by certain types of NPs. His analysis relies on many of the characteristics that would eventually be associated with phasehood, such as movement to the edge and inaccessibility of syntactic objects within the prepositional phase. The theory presented in this dissertation provides additional explanatory power in accounting for adjunct islands as typical cyclic movement islands in the same way that those cyclic applications of Spell-out can account for WH islands. This additional explanatory power should add some persuasiveness to the case for prepositions as phase-heads.

Under the phasal architecture built in Chapter 3, I provided a modified account of parasitic gap constructions that relies on a cyclic Spell-out operation and sideward movement to provide a more empirically complete and conceptually sound account of parasitic gaps and the island effects we find within them. One of the crucial differences between this theory and Nunes' (2004,2012) theory is that there is movement in the parasitic domain. Nunes accounted for islands with the adjunct Spell-out rule given by Nunes and Uriagereka (2001), which I argued was an unnecessary extra operation. Additionally, Nunes' theory has trouble accounting for WH

islands within the parasitic domain, since there is no movement within the parasitic domain. My theory accounts for all of these island effects with movement and cyclic phase Spell-out to ensure that these island effects are accounted for. Parasitic gap constructions are licit when the parasitic element is able to move to the edge of the parasitic domain, which then allows for sideward movement out of the parasitic domain where it can check its WH feature. In this way, WH movement in the parasitic domain is motivated like cyclic movement of WH phrases through clauses that will not themselves check its WH feature. The WH phrase moves to a position where it can continue up to a CP that will check the WH feature of the moved phrase. In the case of parasitic gaps, this involves upward movement through the parasitic domain and then a sideward movement to the matrix clause.

Chapter 4 broadens the scope of the theory presented in Chapter 3, both by expanding its coverage of the various restrictions on parasitic gaps and extending it to discuss ATB constructions. Nunes' (2004) theory of parasitic gap constructions relies on the operation Form Chain and its restrictions in order to understand the structural restrictions on parasitic gaps. I show that this can be extended to the theory of parasitic gaps presented in Chapter 3, where the specific status of intervening DPs changes significantly due to movement within the parasitic domain. I also show that adjustments must be made to Nunes' assumptions about case marking within the parasitic domain. Given the sensitivity that some languages show to the case marking found in parasitic gaps, it stands to reason that case must be assigned to DPs within the parasitic domain. However, I show that the S-structure licensing, the c-command condition on parasitic gaps, and the restriction against A-chains licensing parasitic gaps can all still be understood under the assumptions of Form Chain.

The continued usefulness of Nunes Form chain operation can be credited to its atomized nature. We can see in the analysis found in Chapter 4 that Nunes' conception of Form Chain survives the problems I point out with his analysis of the derivation of parasitic gap construction. Additionally, we can see in Chapter 5 that Form Chain has advantages over theories where chains are simply a history of movement. For example, theories of chains that rely on a movement history require additional conceptual machinery. They need something be able to keep track of all of the copies of a syntactic object, even across cyclic Spell-out and the transfer of that information to any interfaces that might need such information (LF and PF). This is not a trivial addition to the theory. While one might argue that Form Chain simply replaces one mechanism with another, Form Chain also makes more accurate empirical predictions in a theory which includes sideward movement. We see later, in Chapter 5, that sideward movement into relative clauses would be problematic for a theory of chains that relies on a movement history. If those histories designate the latest copy as the head of the chain, then they will predict that the wrong copy is pronounced in relative clause structures. If they somehow designate the highest copy as the head of a chain, I would say that this is in fact an argument for Nunes' Form Chain operation, since the operation that determines the order of the chain must be added to the theory, in addition to the movement history. If in fact both are required, then the movement history is a redundant mechanism.

In addition, the combination of Form Chain and sideward movement produces a natural account of the parasitic gap phenomena. One of the primary mysteries of parasitic gap phenomena is the nature of analyzing a forked chain. Theories that restrict movement to being 'upward only' should not ever predict the forked chains we see in parasitic gap constructions. Once you allow for sideward movement, however, there is still a need for additional restriction

that an atomic Form Chain operation provides. Movement histories should predict a much wider variety of chains, since the mechanism that keeps track of chains should have no inherent reason to be sensitive to the intervention effects we find in parasitic gap constructions. The separation of movement from Form Chain provides a principled reason for the restrictions we find on Form Chain, instead of needing to find reasons that certain chains are illicit.

Chapter 4 also expands this sideward movement theory to ATB constructions. When comparing ATB and parasitic gap constructions, one of the most salient details is the asymmetry in the types of phrases that can be moved out of ATB and parasitic gap constructions. Nunes (2004) argues that this can be accounted for by stating that the types of NPs that are prohibited in parasitic gap constructions are those that do not receive thematic roles from the matrix verb when they are targeted for sideward movement. If this were the case, it might account for their inability to move, but it would not account for their ability to move in ATB constructions. Nunes suggests that Last Resort can be satisfied through a parallelism requirement in conjunctions. I argue that this cannot be the motivation for movement at a particular point in the derivation, since it would require knowledge about the eventual structure of the conjunction. Since the computational system should not have that knowledge at the point in the derivation at which the movement would take place, we cannot predict that there would be movement. Instead, I use Munn's (2001) account of the restrictions on NPs in parasitic gaps. He relies on Szabolcsi and Zwarts' (1997) analysis of weak islands to suggest that there are scopal elements within parasitic gap constructions which prohibit non-referential NPs from scoping out of the parasitic gap. In adopting this analysis, we should not expect any such restriction in ATB construction, since they will not be weak islands in the same way. This requires no knowledge of the structure, and therefore can account for the asymmetry between parasitic gaps and ATB

constructions. I also argue that sideward movement is therefore not motivated by Last Resort, but is free. It is only restricted by those elements which are available within the derivational workspace.

Lastly, in Chapter 5, I showed how the theory presented extends to relative clause structures. Nunes (2004) and Henderson (2007) provide sideward movement accounts of the derivation of relative clause structures. I show that Nunes' analysis is problematic for its loss of the connection between the copy in the thematic position (the WH-marked DP) and the CP that eventually merges into the specifier of CP. It also assumes that there is a special derivation for WH phrases with relative clauses that is unavailable for other WH phrases. Henderson, on the other hand, provides an account that allows for late adjunction, and allows for relative clauses to have both raising and matching effects simultaneously. His analysis is compelling, but relies on late adjunction. While many interesting phenomenon, especially anti-reconstructions effects, extraposition of relative clauses, and ACD constructions, have been analyzed using late adjunction, it is a conceptually problematic operation, since it assumes that some merger can violate the Extension Condition. Sportiche (2016) provides an account of these phenomena in terms of Neglect, which allows for material to be ignored at the interfaces. He argues that an object not being interpreted in a low position is the result of Neglect operation targeting that structure. He argues that adjuncts can be the target of Neglect in low positions, but not arguments, due to the need to locally satisfy predicates. My theory of the derivation of relative clause structures depends on Sportiche's conception of Neglect, but also relies on data from pseudo-complements. Given the asymmetry between true complement and pseudo-complements with regard to their reconstruction into the relative clause, I argue that in order to leverage the

structural difference between complements and pseudo-complements, there must be sideward movement into the relative clause.

First, this theory claims, contrary to other theories, that the derivation may proceed from less embedded environments to more embedded environments. Therefore, sideward movement should also be able to proceed in this direction under the right conditions. Since the relative clause is embedded within the NP that it will eventually merge to, the derivations presented in Chapter 5 suggest that a more embedded structure can be built up later than the less embedded material.

Additionally, movement from a less embedded environment to a more embedded environment fills in a gap in the predicted possibilities of the theory laid out in Chapter 3. For parasitic gap constructions, sideward movement is restricted by the Spell-out of moved objects. Since a copy of the moved object is present elsewhere in the derivation, I argue that this allows for the Spell-out of the original copy in order to removed unnecessary syntactic material from the derivational workspace. This makes the spelled-out object unavailable for later movement. Therefore, movement into the parasitic domain would result in that moved object being caught in an adjunct island without checking the moved object's WH feature. However, sideward movement out of the parasitic domain allows for the WH to be checked in the parasitic domain. Therefore, the effect that this has of restricting movement into parasitic gap constructions is specific to parasitic gap constructions. The need to motivate movement is not a problem in the case of relative clauses. There is evidence of movement within relative clauses, but we know that relative determiners within the relative clause are present. It is not unreasonable to assume that they can have features that drive the movement within the relative clause. This is especially reasonable given their resemblance to WH determiners in most languages. Since the restrictions

on parasitic gap and ATB constructions are not reflected in relative clause structures, it stands to reason that the directionality of movement could be different. The evidence from pseudo-complements also compels us to have a theory in which relative clauses are constructed this way. Finally, I show that the theory outlined in Chapter 5, which is composed of a sideward movement account of the derivation of relative clauses and Sportiche's theory of Neglect, can account for the data surrounding relative clause structure given in Nunes (2004), Sauerland and Hulsev (2006), Henderson (2007), and Sportiche (2016), while also expanding the empirical coverage of the theory with regards to the asymmetry in the reconstruction facts between complements and pseudo-complements.

6.2 Contextualizing this research

The theory presented here benefits from appealing to the efficiency constraints placed on the computational system. These general principles of efficient computation give motivation for the phase-by-phase Spell-out over the course of the derivation. While cyclic rule application can be motivated by empirical data, one advantage of a phasal system of computation is that it can be attributed to these computational principles. Other analyses of these cyclic phenomena such as barriers and subjacency suffer from a problem of conceptual motivation. Such 'third factor' explanations for the restrictions and patterns we find in language will hopefully prove to deepen our understanding of the phenomena we examine.

While the claims in this dissertation are fairly specific to the structures at hand (namely parasitic gap constructions, adjunct islands, ATB constructions, and relative clauses), it should be the case that the accounts and the methodology here can be extended into other domains. One of the most fruitful is found by looking at the commonality we find in some of these phenomena.

Parasitic gap constructions and relative clauses have both been analyzed by assuming that null operators create relationships between pronounced objects and gaps in these domains. One of the advantages of a sideward movement analysis of these phenomena is that the relationship between the gap in question and its filler is transparent, since they are both copies of the same syntactic object. This is a conceptual advantage of the theory presented in this dissertation. It is beyond the scope of this dissertation to take on all potential operator constructions and show that they can be analyzed using sideward movement, but it would be a boon to this analysis and to our understanding of these phenomena if that were true. The theory presented here may be a case study in that endeavor.

Even as a relatively young field, the broader syntactic literature is large and complicated. Many years of research have been poured into understanding, describing, and explaining the syntactic phenomena we see. The goal of the Minimalist Program is to do more with less. By taking a small number of operations and assumptions about the syntax that can be well motivated, the hope is to find deeper explanation than what has been found previously. I believe that this dissertation has been a successful exercise in that endeavor. From well-motivated assumptions about Merge, Agree, phases within the computational system, and the requirements of the interfaces, including the conditions on Form Chain and the application of Neglect, I have shown that a wide variety of data can be accounted for. This has not been a success because of a large expansion of empirical coverage, though I would argue that some has indeed been gained. Where it has been successful is in providing a deeper understanding of the phenomena accounted for. This is the steep climb that the Minimalist Program has ahead. Can the vast array of descriptions, explanations and understandings of language that have been developed over the history of syntax truly be accounted for with less assumptions and not more? I think the success

of the Minimalist Program so far points toward the affirmative, but there is so much yet to be accounted for. But the measure of success for the Minimalist Program until then is whether it is providing a deeper explanation than the theories that have come before it. This is the measure by which I think this dissertation makes its most meaningful contribution.

REFERENCES

REFERENCES

- Abels, K. (2003). *Successive Cyclicity, Antilocality and Adposition Stranding*. Ph.D. dissertation, University of Connecticut, Storrs, CT.
- Aoun, J., & Clark, R. (1985). On non-overt operators. *Southern California Occasional Papers in Linguistics*, 10, 17-36.
- Baker, C. L. (1978). *Introduction to generative transformational syntax*. Englewood, Cliffs, NJ: Prentice Hall.
- Barss, A. (1986). *Chains and anaphoric dependencies*. Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA.
- Bhatt, R. (2002). The raising analysis of relative clauses: Evidence from adjectival modification. *Natural Language Semantics*, 10(1), 43-90.
- Bianchi, V. (2000). The Raising Analysis of Relative Clauses: A Reply to Borsley. *Linguistic Inquiry*, 31(1), 123-140.
- Bobaljik, J. D. (2008). Where's phi? Agreement as a post-syntactic operation. In D. Adger, D. Harbour, & S. Béjar (Eds.), *Phi theory: Phi-features across interfaces and modules* (pp. 295-328). Oxford: Oxford University Press.
- Boeckx, C. (2003). *Islands and chains: Resumption as stranding*. Amsterdam: John Benjamins Publishing.
- Boeckx, C., Hornstein, N., & Nunes, J. (2010). *Control as movement*. New York: Cambridge University Press.
- Brame, M. K. (1968). *A New Analysis of the Relative Clause: Evidence for an Interpretive Theory*. Unpublished Manuscript, MIT.
- Chomsky, N. (1977). On Wh-movement. In P. Culicover, T. Wasow, & A. Akmajian (Eds.), *Formal Syntax* (pp. 71-132). New York: Academic Press.
- Chomsky, N. (1982). *Some concepts and consequences of the theory of government and binding*. Cambridge, MA: MIT press.
- Chomsky, N. (1986). *Barriers* (Vol. 13). Cambridge, MA: MIT Press.
- Chomsky, N. (1995). *The Minimalist Program*. Cambridge, MA: MIT Press.
- Chomsky, N. (2000). Minimalist Inquiries: The Framework. In R. Martin, D. Michaels, & J. Uriagereka (Eds.), *Step by Step: Essays on minimalist syntax in honor of Howard Lasnik* (pp. 89-155). Cambridge, MA: MIT Press.
- Chomsky, N. (2001). Derivation by Phase. In M. Kenstowicz (Ed.), *Ken Hale: A life in language*. (pp. 1-52). Cambridge, MA: MIT Press.

- Chomsky, N. (2004). Beyond Explanatory Adequacy. In A. Belletti (Ed.), *Structures and beyond* (pp. 104-131). Oxford: Oxford University Press.
- Chomsky, N., & Lasnik, H. (1993). The theory of principles and parameters. In J. Jacobs, A. von Stechow, W. Sternefeld, & T. Vennemann (Eds.), *Syntax: An international handbook of contemporary research* (pp. 506-569). Berlin: de Gruyter.
- Cinque, G. (1990). *Types of A'-dependencies*. Cambridge, MA: MIT Press.
- Contreras, H. (1984). A note on parasitic gaps. *Linguistic Inquiry*, 15(4), 698-701.
- Culicover, P. W. (2001). Parasitic Gaps: A history. In P. W. Culicover, & P. M. Postal (Eds.), *Parasitic gaps* (pp. 3-68). Cambridge, MA: MIT Press.
- Engdahl, E. (1983). Parasitic gaps. *Linguistics and Philosophy*, 6(1), 5-34.
- Fox, D. (1995). Economy and scope. *Natural Language Semantics*, 3(3), 283-341.
- Fox, D. (2002). Antecedent-contained deletion and the copy theory of movement. *Linguistic Inquiry*, 33(1), 63-96.
- Frampton, J. (1990). Parasitic gaps and the theory of wh-chains. *Linguistic Inquiry*, 21(1), 49-77.
- Franks, S. (1993). On parallelism in across-the-board dependencies. *Linguistic Inquiry*, 24(3), 509-529.
- Freidin, R. (1986). Fundamental issues in the theory of binding. In B. Lust, *Studies in the acquisition of anaphora* (pp. 151-188). Dordrecht: Reidel.
- Fukui, N. (1986). *A theory of category projection and its applications*. Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA.
- Fukui, N. (1988). Deriving the differences between English and Japanese: A case study in parametric syntax. *English Linguistics*, 5, 249-270.
- Fukui, N., & Speas, M. (1986). Specifiers and projection. *MIT Working Papers in Linguistics*, 8, 128-172.
- Geis, M. (1970). *Adverbial Subordinate Clauses in English*. Doctoral dissertation, MIT, Cambridge, MA.
- Haida, A., & Repp, S. (2012). *Locality restrictions on sideward movement: An investigation of parasitic gaps, ATB constructions and question word coordinations*. unpublished. Retrieved Jan 2012, from <http://ling.auf.net/lingbuzz/001666>
- Harley, H. (1995). Abstracting away from abstract case. In J. Beckman (Ed.), *NELS 24: Proceedings of the North Eastern Linguistics Society*. 25, pp. 207-222. Amherst, MA: GLSA.
- Henderson, B. (2007). Matching and Raising Unified. *Lingua*, 117(1), 202-220.

- Hofmeister, P., Culicover, P., & Winkler, S. (2015). Effects of processing on the acceptability of “frozen” extraposed constituents. *Syntax*, 18(4), 464-483.
- Holmberg, A., & Hróarsdóttir, T. (2004). Agreement and movement in Icelandic raising constructions. *Lingua*, 114(5), 651-673.
- Hornstein, N. (2001). *Move! A minimalist theory of construal*. Oxford: Wiley-Blackwell.
- Hornstein, N., & Nunes, J. (2002). On asymmetries between parasitic gap and Across-the-Board constructions. *Syntax*, 5, 26-54.
- Hornstein, N., & Nunes, J. (2008). Adjunction, labeling, and bare phrase structure. *Biolinguistics*, 2(1), 57-86.
- Hornstein, N., Nunes, J., & Grohmann, K. K. (2005). *Understanding minimalism*. Cambridge: Cambridge University Press.
- Huang, C.-T. J. (1982). *Logical relations in Chinese and the theory of grammar*. Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA.
- Hulsey, S., & Sauerland, U. (2006). Sorting out relative clauses. *Natural Language Semantics*, 14(2), 111-137.
- Jackendoff, R. S. (1977). *X' syntax: A study of phrase structure*. Cambridge, MA: MIT Press.
- Kayne, R. S. (1983). Connectedness. *Linguistic Inquiry*, 14(2), 223-249.
- Kayne, R. S. (1984). *Connectedness and binary branching*. Dordrecht: Foris Publications.
- Kayne, R. S. (1994). *The antisymmetry of syntax*. Cambridge, MA: MIT Press.
- Kayne, R. S. (2010). *Comparisons and contrasts*. Oxford University Press on Demand.
- Kiss, K. É. (2001). Parasitic chains revisited. In P. W. Culicover, & P. M. Postal (Eds.), *Parasitic gaps* (pp. 99-124). Cambridge, MA: MIT Press.
- Kitahara, H. (1994). Restricting ambiguous rule-application. In M. Koizumi, & H. Ura (Eds.), *Formal Approaches to Japanese Linguistics 1* (pp. 179-209). Cambridge, MA: MITWPL.
- Kuroda, S. Y. (1988). Whether we agree or not: A comparative syntax of English and Japanese. In W. J. Poser (Ed.), *Papers from the Second International Workshop on Japanese Syntax* (pp. 103-143). Stanford, CA: CSLI Publications.
- Larson, R. (1990). Extraction and Multiple Selection in PP. *The Linguistic Review*, 7, 169-182.
- Lasnik, H., & Saito, M. (1991). On the subject of infinitives. In L. Dobrin, L. Nichols, & R. M. Rodriguez (Eds.), *Papers from the 27th regional meeting of the Chicago Linguistic Society 1991. Part 1: The General Session* (pp. 324-343). Chicago, IL: Chicago Linguistics Society.

- Lebeaux, D. (1988). *Language acquisition and the form of the grammar*. Ph.D. dissertation, University of Massachusetts, Amherst.
- 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). San, Diego, CA: Academic Press.
- McFadden, T. (2004). *The position of morphological case in the derivation: A study on the syntax-morphology interface*. Doctoral dissertation, University of Pennsylvania.
- Montague, R. (1974). English as a formal language. In R. H. Thomason (Ed.), *Formal Philosophy: Selected Papers of Richard Montague*. New Haven: Yale University Press.
- Müller, G. (2009). Ergativity, accusativity, and the order of Merge and Agree. In K. Grohmann (Ed.), *Explorations of Phase Theory. Features and Arguments* (pp. 269-308). Berlin: Mouton de Gruyter.
- Müller, G. (2011). *Constraints on displacement: A phase-based approach*. John Benjamins Publishing.
- Munn, A. (1992). A Null Operator Analysis of ATB Gaps. *The Linguistic Review*, 9, 1-26.
- Munn, A. (1993). *Topics in the Syntax and Semantics of Coordinate Structures*. Doctoral dissertation, University of Maryland, College Park.
- Munn, A. (1994). A Minimalist Account of Reconstruction Asymmetries. In M. Gonzalez (Ed.), *NELS 24: Proceedings of the North Eastern Linguistics Society* (pp. 397-410). Amherst, MA: GLSA.
- Munn, A. (2001). Explaining parasitic gap restrictions. In P. W. Culicover, & P. M. Postal (Eds.), *Parasitic Gaps* (pp. 369-392). Cambridge, MA: MIT Press.
- Narita, H. (2011). *Phasing in full interpretation*. Doctoral dissertation, Harvard University, Cambridge, MA.
- Nunes, J. (1995). *The copy theory of movement and linearization of chains in the Minimalist Program*. Doctoral dissertation, University of Maryland at College Park.
- Nunes, J. (2001). Sideward movement. *Linguistic Inquiry*, 32(2), 303-344.
- Nunes, J. (2004). *Linearization of Chains and Sideward Movement*. Cambridge, MA: MIT Press.
- Nunes, J. (2012). Sideward movement: Triggers, timing, and outputs. In M. Uribe-Etxebarria, & V. Valmala (Eds.), *Ways of Structure Building*. (pp. 114-142). Oxford: Oxford University Press.
- Nunes, J., & Uriagereka, J. (2000). Cyclicity and extraction domains. *Syntax*, 3, 20-43.
- Partee, B. (1975). Montague Grammar and Transformational Grammar. *Linguistic Inquiry*, 6, 203-300.

- Postal, P. M. (1993). Parasitic gaps and the across-the-board phenomenon. *Linguistic Inquiry*, 24(4), 735-754.
- Quine, W. V. (1960). *Word and Object*. Cambridge, MA: MIT Press.
- Richards, M. D. (2011). Deriving the edge: What's in a phase? *Syntax*, 14(1), 74-95.
- van Riemsdijk, H. C. (1978). *A case study in syntactic markedness: The binding nature of prepositional phrases*. Dordrecht: Foris.
- van Riemsdijk, H., & Williams, E. (1981). NP-structure. *The Linguistic Review*, 1(2), 171-218.
- Rizzi, L. (2004). *On the form of chains: Criterial positions and ECP effects*. Ms., University of Siena.
- Schachter, P. (1973). Focus and Relativization. *Languague*, 49(1), 19-46.
- Sigurðsson, H. Á. (1993). Agreement as head visible feature government. *Studia Linguistica*, 47(1), 32-56.
- Sigurðsson, H. Á. (1996). Icelandic finite verb agreement. *Working Papers in Scandinavian Syntax*, 57.
- Sportiche, D. (2016). *Neglect*. unpublished. Retrieved Feb 2016, from <http://ling.auf.net/lingbuzz/002775>
- Stepanov, A. (2007). The end of CED? Minimalism and extraction domains. *Syntax*, 10(1), 80-126.
- Szabolcsi, A., & Zwarts, F. (1997). Weak islands and an algebraic semantics for scope taking. In A. Szabolcsi (Ed.), *Ways of Scope Taking* (pp. 217-262). Dordrecht: Kluwer.
- Takahashi, D. (1994). *Minimality of movement*. Ph.D. dissertation, University of Connecticut, Storrs, CT.
- Taraldsen, K. T. (1981). The theoretical implications of a class of marked extractions. In A. Belletti, L. Brandi, & L. Rizzi (Eds.), *Theory of markedness in generative grammar* (pp. 475-516). Pisa: Scuola Normale Superiore.
- Uriagereka, J. (1999). Multiple spell-out. In S. D. Epstein, & N. Hornstein (Eds.), *Working minimalism*. Cambridge, Mass.: MIT Press.
- Wexler, K., & Culicover, P. (1980). *Formal principles of language acquisition*. Cambridge, MA: MIT Press.
- Williams, E. (1978). Across-the-board rule application. *Linguistic Inquiry*, 9(1).
- Williams, E. (1990). The ATB theory of parasitic gaps. *The Linguistic Review*, 6(2), 265-279.