Paths: the Ghost of Features Past

Elise Newman & Kenyon Branan

Abstract

We discuss a set of cases where an adjunct or specifier is neither uniformly opaque nor uniformly transparent for extraction, but is rather sensitive to other properties of its local context. These cases pose a problem for classic analyses of the CED, as non-complements are usually taken uniformly to be islands for extraction. To describe these variable transparency effects, we first suggest that each exception to the CED occurs in a multiple specifier environment, like that proposed by Nissenbaum (2000) for parasitic gap constructions. In addition, we propose a theory of locality which makes fine-grained predictions about what phrases are accessible for operations external to them. The proposal features a path-based view of locality in dependency formation (Kayne 1981, Pesetsky, 1982, McFadden et al. 2019, a.o.), where projection of a feature from the goal to the probe is necessary for a dependency to be well-formed. On our view, the feature projection algorithm allows goals to project their features only when their sisters have certain properties, predicting complements and second specifiers, but not first specifiers, to be transparent for extraction.

Keywords: syntax, control, wh-movement, islands, scrambling

Data Availability Statement: All original data generated by this study are given explicitly in the text and in the supplementary material.

Acknowledgements: First and foremost, we thank the members of the LASER team for their support and feedback on this project: Thomas McFadden, Sandhya Sundaresan, Rob Truswell, Hedde Zeijlstra. We also benefited greatly from the questions and attention of the participants at GLOW in Asia XIII, and audiences at MIT and Stony Brook University. Finally, we want to thank the anonymous reviewers for their thoughtful comments and

questions, which were instrumental in helping us test the proposal in new and important ways. All mistakes are our own.

1. Introduction

In this paper, we discuss some exceptions to the CED, such as (1). In (1), wh-movement is permitted from an adjunct clause, despite the fact that adjuncts are typically thought to be islands for extraction.

(1) What is the flower$_i$ open [PRO$_i$ to attract ___]?
(cf. *The flower is open to attract passing pollinators.*)

Interestingly, this counterexample to the CED is sensitive to interpretation – while the obligatory control (OC) interpretation of (1) permits wh-movement out of the adjunct clause, a corresponding non-obligatory control (NOC) variant, as in (2), blocks wh-movement from the adjunct clause (Truswell 2011).

(2) *What is the door$_i$ open [PRO$_{arb}$ to listen to ___]?
(cf. *The door is open to listen to confessions.*)

Taking inspiration from Müller (2010) and Landau (2021), we suggest that (1) and other exceptions to the CED are multiple specifier constructions (treating adjuncts as a kind of specifier). Unlike these previous works, however, we offer a theory of how multiple specifier constructions obviate island effects that also takes into account the significance of interactions between different dependencies, such as movement and control, as in (2).

The analysis is built on the idea that (all) syntactic dependencies make use of an operation: Search (Chomsky 2004). Importantly for the present approach, Search is not blind, but is guided by the distribution of checked features in the clause. An algorithm for projecting said features therefore makes predictions about which parts of the structure are accessible to Search and which aren't, thus producing both transparent and opaque domains. As we will see, the particular algorithm that we propose, combined with the above assumptions about Search, make nuanced predictions about when a specifier or adjunct will be transparent for extraction or control. One of these predictions is that first specifiers of a head are opaque, while second specifiers are transparent. We argue that this theory explains both basic CED effects, as well as exceptions to the CED, more successfully and in a broader sense than alternative theories.

An outline of the paper is as follows: §2 presents the empirical evidence that multiple specifier environments obviate the CED, with examples of wh-extraction/parasitic gap licensing across control adjuncts, as well as extraction from specifiers (*Melting*, Müller 2010). §3 presents the analysis, which is to treat these effects through the lens of a theory of probing. According to our proposal, probes use the distribution of projected checked features to guide them to goals, where dependency formation is contingent on successfully finding a goal. §4 shows how the analysis captures the phenomena in §2. §5 discusses alternative approaches to the CED and concludes.

2. Variable transparency in multiple specifier constructions

First we discuss an observation from Truswell (2011) that wh-movement out of adjuncts tracks the obligatory/non-obligatory control distinction: control adjuncts that are transparent for wh-movement are obligatorily controlled, while control adjuncts that are opaque to wh-movement are non-obligatorily controlled.

Second, we observe that parasitic gaps also track the obligatory/non-obligatory control distinction. Since Nissenbaum (2000)'s independently proposed structures for parasitic gap constructions are also multiple specifier configurations, we suggest that they should receive a common analysis with the wh-extraction facts. In both cases, the adjunct appears to be transparent for multiple dependencies, such as binding of an operator/wh-extraction and obligatory control, and in both cases, the adjunct appears in a multiple specifier environment. Thus the same explanation that accounts for correlations between wh-movement and control could extend to parasitic gaps and control.

Third, we discuss melting, a phenomenon described extensively in Müller 2010, in which scrambling an object across a subject licenses extraction out of the subject (in violation of the CED). We propose, following Müller and others, that scrambling involves a step of object-movement through the edge of *v*P. Since the subject is also generated at the edge of *v*P, scrambling creates a multiple specifier environment, which we propose is analogous to the first two case studies. Thus, the same explanation that accounts for extraction from control adjuncts can extend to melting examples.

2.1 Control and adjunct (non)-islands

The obligatory/non-obligatory distinction is shown in (3).

(3) a. The flower$_i$ is open [PRO$_i$ to attract passing pollinators].
b. The door$_i$ is open [PRO$_{arb}$ to listen to confessions].

The non-agentive, inanimate subjects in (3) may corefer with an embedded PRO, as in (3a), or not, as in (3b). In the latter case, the embedded PRO is interpreted as referring to an arbitrary individual/group who might serve as a *listener* in this context. Insights from Chomsky (1981), Williams (1992), and Landau (2013, 2021) teach us that an inanimate interpretation for PRO requires c-command by a controller, while an animate interpretation for PRO does not. To reflect this difference, we call (3a) a case of *Obligatory* control (henceforth OC), and (3b) a case of *Non-obligatory* control (henceforth NOC).[1]

McFadden and Sundaresan (2018) have argued that obligatory and non-obligatory control, despite appearances in (3), are in complementary distribution. On that view, a

1. Interestingly, these control adjuncts show a Weak Island effect – they permit extraction of a DP but not an adjunct.

(i) *How did the flower open [in order to attract pollinators ____]
→ A: with a particular UV pattern

While we do not offer a theory of Weak Island-hood here, see Appendix A for some possible views of Weak Islands on the present theory.

better description of (3) would be that (3a) is a case of genuine control, while (3b) is what happens when control cannot be established (i.e. an *elsewhere* construction). When the predicate of the adjunct clause is strongly biased towards an interpretation where its subject has sentience — as is the case with *listen* — the preference for the elsewhere construction emerges when there is no licit controller. This view of the obligatory/non-obligatory control distinction is further motivated by the observation that OC adjuncts are transparent for wh-movement, while NOC adjuncts are opaque.

(4) a. What is the flower$_i$ open [PRO$_i$ to attract ____]?

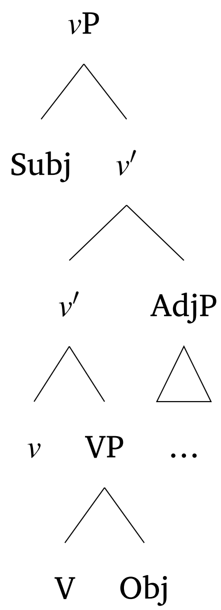
b. *What is the door$_i$ open [PRO$_{arb}$ to listen to ____]?

The examples in (4) show that infinitival adjuncts can either be fully transparent for wh-movement and control or fully opaque. We propose that this variable transparency of control adjuncts results from a structural ambiguity in their attachment sites.

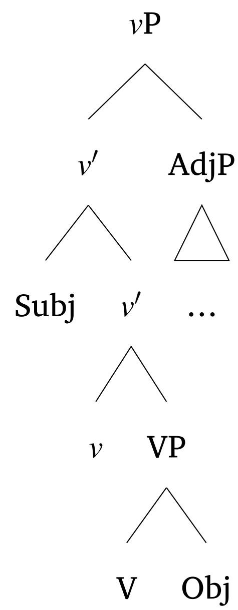
Following Landau (2021) and references there, we propose that the controlled adjuncts under discussion are attached within *v*P. In addition, we follow Landau in assuming that adjuncts which are ambiguous between an OC/NOC interpretation have an ambiguous position within *v*P. Landau suggests that the two relevant attachment positions for a controlled adjunct of this sort are either above or below the base position of the external argument (illustrated in (5)). Each choice is compatible with a different control outcome.

(5) Landau's OC vs. NOC

a. OC: subject is second specifier



b. NOC: subject is first specifier



This proposal suggests that the variable transparency of an adjunct clause is tied to its position. An adjunct that is in the right position to participate in obligatory control is also in the right position to be transparent to wh-movement. An adjunct that is in the wrong position to participate in obligatory control will conversely be opaque to wh-movement. We now discuss other cases where the position of an adjunct is shown to affect its accessibility to dependencies across it.

2.2 Control and parasitic gaps

We have discussed how wh-movement out of adjuncts correlates with obligatory control into them. We now observe that this correlation between wh-movement and control is not specific to wh-movement out of adjuncts. Parasitic gaps inside adjuncts show the same effect. Observe in (6) that parasitic gaps are possible in OC adjuncts, but not in NOC adjuncts.

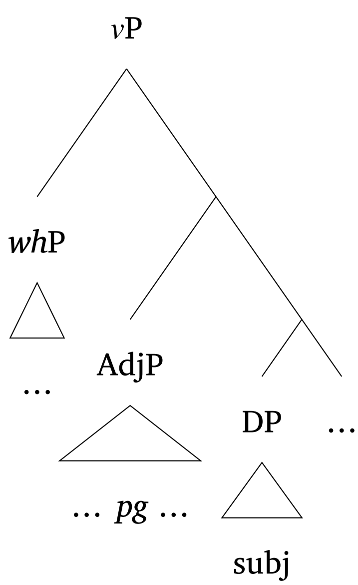
(6) a. [What direction]$_i$ was the flower$_j$ opened to ~~what direction~~
[OP$_i$ PRO$_j$ in order to attract passing pollinators from ~~OP~~]?

b. * [What sort of person]$_i$ was the door$_j$ opened to ~~what sort of person~~
[OP$_i$ PRO$_{arb}$ in order to listen to confessions from ~~OP~~]?

Before, we speculated that OC adjuncts are transparent as a consequence of the position that they are merged in. Adjuncts that are adjoined in a position that makes them available for OC are transparent both for control as well as *wh*-movement. The same adjunct merged in a different position will be opaque for both control and *wh*-movement, forcing an NOC interpretation of PRO.

Nissenbaum (2000) develops a theory of parasitic gap licensing that is compatible with this general approach to (6). For independent reasons, Nissenbaum proposes that parasitic gap-containing adjuncts must merge in a particular position: immediately above the subject that controls PRO, and immediately below a position occupied by the *wh*-element which licenses the parasitic gap. In other words: parasitic gap-containing adjuncts must be second specifiers of *v*P, schematized below.

(7) **Nissenbaum's parasitic gap licensing structure**



To recap, according to Nissenbaum's analysis, if the adjunct merges as a second specifier, it will be accessible to parasitic gap licensing. Since parasitic gaps are only licensed in OC, not NOC, adjuncts in second specifier position must also allow the adjunct to participate in OC. NOC adjuncts must not be in this position, since they block parasitic gap licensing. On this view, if the adjunct is in a different position, it will no longer be accessible for parasitic gap licensing or OC, accounting for the correlation of these different dependencies into the adjunct clause.

We have now seen two different phenomena where control appears to correlate with another kind of dependency into an adjunct clause. On the one hand, we saw that control adjuncts permit wh-extraction out of them while NOC adjuncts don't. Similarly, we saw that control adjuncts also tolerate parasitic gaps while NOC adjuncts don't. In both cases, there is precedent from the literature to trace accessibility to these dependencies to the position of the adjunct: adjuncts in a particular position are accessible to OC/parasitic gaps, suggesting that adjuncts in other positions are opaque to such dependencies.

Unfortunately, we have reached a conflict. Landau proposed that it was the *first* specifier position that made an adjunct accessible to OC, while Nissenbaum suggested that it was the *second* specifier position that made an adjunct accessible to parasitic gap licensing. For these two dependencies to correlate, we need to decide which of these positions is the right one for adjuncts that are transparent to OC, wh-movement, and parasitic gaps. We resolve this issue in §2.4, after discussing a third example of multiple specifier constructions with variable transparency effects.

Before moving on, we want to address an analogy that we are drawing between dependencies of different types. We follow Chomsky (1986), Larson (1988), Postal (1998), and Nissenbaum (2000) in assuming that parasitic gap constructions do not involve ATB wh-movement out of both matrix and adjunct clauses, but rather involve binding of an operator that moves adjunct-internally. Nonetheless, we consider it to be significant that parasitic gaps pattern with wh-movement with respect to their relationship to OC. Thus, we think their similarity in this respect supports a view of operator binding as being subject to the same locality conditions as wh-movement. In §4, we propose a theory of locality that applies uniformly to dependencies such as parasitic gap-licensing, wh-movement and control.

2.3 Melting – from adjuncts to specifiers

Müller (2010) discusses a class of exceptions to the CED, which he calls *Melting* effects. He observes that external arguments in German and Czech are typically opaque to extraction, as expected for specifiers, according to the CED. However, he shows that scrambling an object to the left of the external argument has the effect of making the external argument transparent for extraction. In other words, object scrambling obviates the CED for transitive subjects. This is shown in (8) and (9) for German and Czech respectively. Wh-extraction out of the subject is only available when the object appears to its left.[2]

(8) *German wh-extraction* (ex.36)

2. Müller observes that this effect is not limited to extraction of a DP, but also of PPs. See also Heycock (1991), chapter 3 for examples of pseudo-melting.

(ii) *German PP extraction* (Müller 2010, ex.37)

a.

*[$_{PP1}$	Über	wen]	hat	[$_{DP3}$	ein	Buch	t_1]	[$_{DP2}$	den	Fritz]	beeindruckt?
	about	whom	has		a	book.NOM			the	Fritz.ACC	impressed

intended: "About whom did a book impress Fritz?"

b.

[$_{PP1}$	Über	wen]	hat	[$_{DP2}$	den	Fritz]	[$_{DP3}$	ein	Buch	t_1]	t_2	beeindruckt?
	about	whom	has		the	Fritz.ACC		a	book.NOM			impressed

"About whom did a book impress Fritz?"

(iii) *Czech PP extraction* (ex.44)

a.

*[$_{PP1}$	O	starých	autech]	oslovila	[$_{DP3}$	kniha	t_1]	Petra$_2$.
	about	old	cars	fascinated		book.NOM		Petr.ACC

intended: "A book about old cars fascinated Petr."

b.

(?)[$_{PP1}$	O	starých	autech]	oslovila	Petra$_2$	[$_{DP3}$	kniha	t_1]	t_2.
	about	old	cars	fascinated	Petr.ACC		book.NOM		

"A book about old cars fascinated Petr."

a. *Was$_1$ haben [$_{DP3}$ t_1 für Bücher] [$_{DP2}$ den Fritz] beeindruckt?
what have for books.NOM the Fritz.ACC impressed
intended: "What kind of books impressed Fritz?"

b. Was$_1$ haben [$_{DP2}$ den Fritz] [$_{DP3}$ t_1 für Bücher] t_2 beeindruckt?
what have the Fritz.ACC for books.NOM impressed
"What kind of books impressed Fritz?"

(9) *Czech split DP constructions* (Müller 2010, ex.42)

a. *Stará$_1$ neudeřila [$_{DP3}$ žádná t_1] Petra$_2$.
old.NOM hit no.NOM Petr.ACC
intended: "No old one hit Petr."

b. (?)Stará$_1$ neudeřila Petra$_2$ [$_{DP3}$ žádná t_1] t_2.
old.NOM hit Petr.ACC no.NOM
"No old one hit Petr."

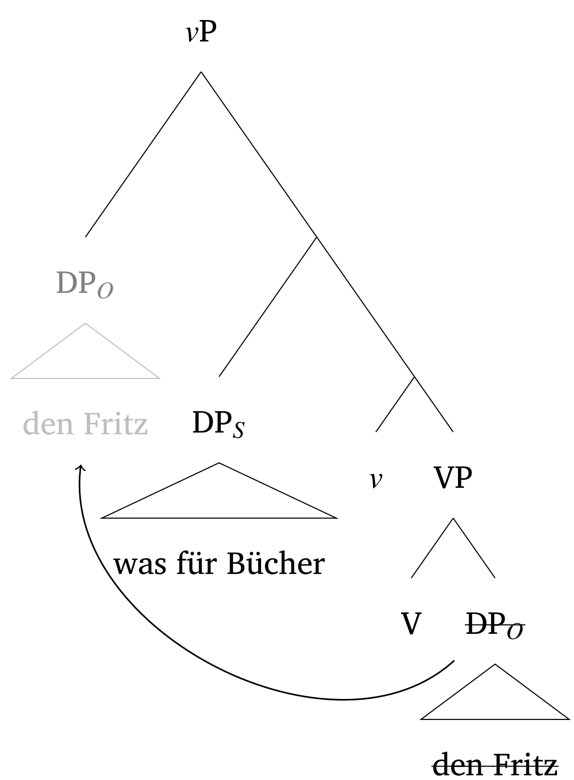
Importantly, Müller cites evidence from Grewendorf (1989) suggesting that the subject of a psych verb like *beeindrucken* is a regular external argument in German, and not a VP-internal argument. Thus, it must be a specifier, making (8b) a true counterexample to the CED. What is surprising about (8) and (9) is that the exact same specifier (e.g. *was für Bücher*) can be opaque in (8/9a) but transparent in (8/9b), solely based on the position of the *object*. The surface position of the object presumably does not affect the specifier-hood of the subject, suggesting that island effects have more to do with local context than the complement/non-complement distinction.

Müller proposes that this occurs because melting examples involve multiple specifier constructions. A starting assumption is that object movement proceeds successive cyclically through the edge of *v*P. Thus, a scrambled object must arrive in the edge of *v*P at some point in the derivation, in which case the (b) examples in (8) and (9) differ from the

(a) examples with respect to the total number of specifiers *v*P has. When no scrambling takes place, the external argument is the only argument to ever occupy the edge of *v*P, while in scrambling derivations, *v*P has *two* specifiers at some point in the derivation.

Müller presents a phase-based theory of the CED, in which phases can only produce escape hatches as long as they are incomplete. The last-merged element in a phase completes the phase, and blocks it from producing an escape hatch. As a result, his theory predicts that only the last-merged specifier of a phase is opaque for extraction. All earlier-merged material is transparent, including specifiers, because they merge early enough for an escape hatch to be produced. In non-scrambling contexts, the subject is the last-merged specifier of *v*P, while in scrambling contexts, Müller proposes that the object is the last-merged specifier of *v*P, making the subject transparent. In other words, his theory requires the following configuration of specifiers in *v*P in order to capture melting effects.

(10) Müller's *v*P in Melting contexts: only the highest specifier is opaque $\rightarrow$ highest specifier must be the scrambled object



Following Moltmann (1990), Grewendorf and Sabel (1999), McGinnis (1999), and Yoshida (2001, a.o.), with evidence from quantifier scope and the position of negation and adverbs, Spec *v*P is not the final landing site for objects scrambled to the left of subjects – a higher position, like Spec TP is. Regardless of the order of specifiers of *v*P, we therefore expect the object to be able to surface in a position that derives the surface word order OS. Since word order is not conclusive to prove the order of specifiers at Spec *v*P, we need another metric.

It is sometimes argued that the second movement step in German scrambling has Ā-properties (Grewendorf 1988; Webelhuth 1992; Müller and Sternefeld 1994), in which case we might be able to diagnose the order of specifiers in *v*P with reconstruction tests. Here, the results suggest ambiguity. Supposing the first movement step may have mixed properties (as it targets Spec *v*P), the first movement step might be able to affect binding relations but the second step should not, providing some insight into the order of specifiers in Spec *v*P. As it turns out, a scrambled anaphoric object may be bound by a subject as in (11a) (with no Condition C effect), consistent with the order of specifiers SO, while a scrambled quantificational object may bind a pronoun in the subject in (11b), motivating the opposite. This may indicate that specifier ordering is generally ambiguous, just as the position of adjuncts is, where the choice of position affects extraction and binding possibilities.[3]

(11) a. ...dass sich$_i$ Hans$_i$ nie rasiert.
that REFL.ACC Hans.NOM never shaves
"...that Hans never shaves himself." (Yoshida 2001, ex.68)

3. One might wonder why scrambling cannot be terminal in Spec *v*P. In other words, why couldn't the object scramble to an inner specifier and just stay there, melting the subject without changing the word order? We propose that a constraint on string vacuous scrambling must rule this possibility out (Hoji 1985).

b.	(?)...weil	jeden_i	sein_i	Hund	gebissen	hat.
	because	everyone.ACC	his	dog.NOM	bitten	has

"...because everyone has been bitten by his dog." (Moltmann 1990, ex.130)

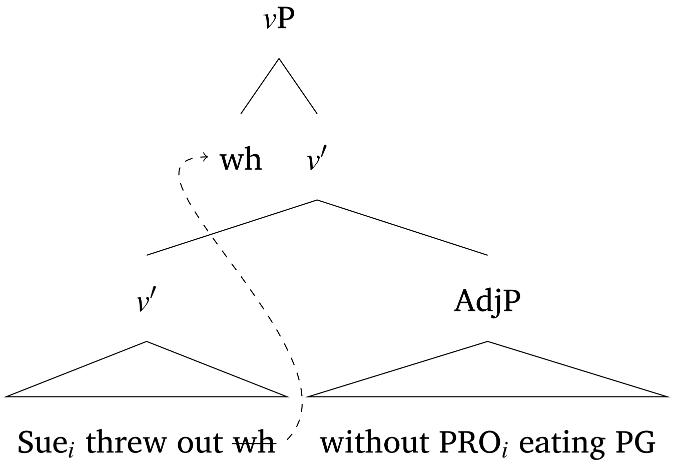
In sum, we have seen three cases in which multiple specifier constructions feed exceptions to the CED. Adjuncts and specifiers that are normally opaque to extraction/parasitic gap licensing may become transparent if they are in a multiple specifier environment. What remains to be decided is which specifier position gets to be exceptionally transparent to such dependencies and what principles underlie this choice. We will argue, contra Müller/Landau but with Nissenbaum, that second specifier position is transparent.

What follows is a discussion regarding the position of adjuncts that contain parasitic gaps, contrasting Landau and Nissenbaum's approaches. The former, but not the latter, is consistent with the proposal that the second specifier position is transparent. Readers who are willing to follow us and Nissenbaum may proceed directly to section 3.

2.4 Which specifiers are transparent?

We propose the following: for dependencies like obligatory control and wh-movement to cross an adjunct clause boundary, the adjunct must merge as the second specifier of *v*P and the subject (controller) must merge as the first specifier of *v*P. We use the term *subject* rather than *external argument* on the assumption that subjects always occupy Spec *v*P at some point in the derivation. When the subject is an external argument, we assume that it externally merges in Spec *v*P; when the subject is an internal argument (as in a passive/unaccusative), we assume that it moves through Spec *v*P en route to Spec TP, following Legate (2003) and Sauerland (2003). Furthermore, we propose that whether the subject has been internally or externally merged in Spec *v*P, the same ambiguity is available to adjuncts: adjuncts may merge either before or after the subject has formed a specifier of *v*P, leading to two available configurations. If the derivation chooses the option in which the subject is a first specifier, then we predict the adjunct to be transparent for OC and wh-movement. This configuration of specifiers is consistent with Nissenbaum 2000, who likewise argues that (OC) adjuncts with parasitic gaps merge higher than the subject, creating the configuration that we want.

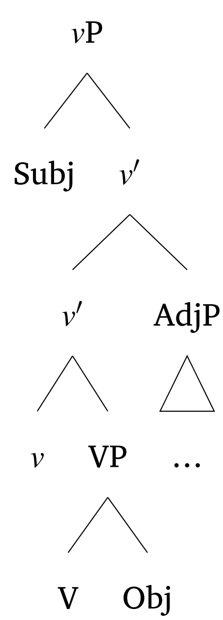
(12) Nissenbaum's configuration of specifiers with PG-containing OC adjuncts
(e.g. *What did Sue throw out without eating?*)



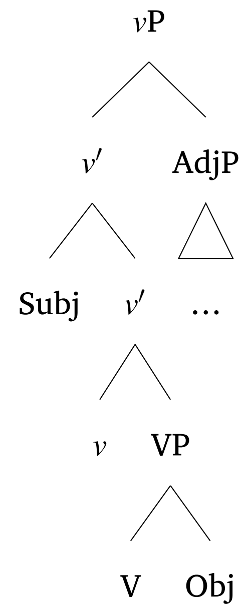
However, this configuration is the exact opposite order of specifiers proposed by Landau 2021, who suggests that OC can only be established if the adjunct merges *below* the subject; NOC arises when the adjunct merges above it.

(13) Landau's OC vs. NOC

a. OC: subject is second specifier



b. NOC: subject is first specifier



Here we will explore each analysis in closer detail and see whether the insights from Landau can be made consistent with our proposal. Starting with Nissenbaum 2000, Nissenbaum offers two main reasons to put PG-containing adjuncts where they are: 1) constituency/binding tests showing they are at least as high as Spec *v*P, and 2) placing them above the subject allows us to treat adjunction as interpreted via predicate modification. He provides examples like (14), which show that the adjunct outscopes material internal to the verb phrase, such as the verb and internal arguments.

(14) Nissenbaum 2000, ex. 27a, 29c, p. 37-8

a. John [filed the papers and shelved the books] without reading them.

b. We gave him$_i$ a book [without talking to John$_i$'s mother].

He argues that (14) shows us that the adjunct must be at least as high as Spec *v*P. He then argues that having the adjunct be a second specifier, where the external argument is a first specifier, gives us the right semantics.

Nissenbaum draws an analogy between parasitic gap containing adjuncts and constructions with operator-movement, such as relative clauses. In a relative clause, an operator moves clause-internally, creating a one-place predicate, which modifies the relative noun and gets interpreted within the scope of the higher determiner. In a parasitic gap construction, he argues that a similar process happens, following Chomsky 1986, Larson 1988, and Postal 1998: there is adjunct-internal operator movement, which creates a one-place predicate. When the adjunct merges in Spec *v*P, its open argument is saturated by the copy of the wh-phrase that moves successive cyclically through Spec *v*P. For this analogy to hold, the adjunct must be of type $<e,t>$, and it's closest c-commanding phrase must be the wh-element.

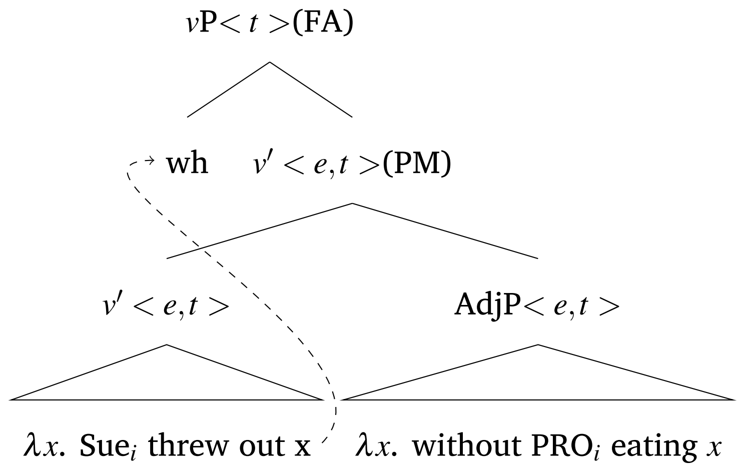
(15) Parasitic gaps as derived by operator movement

a. Relative clause: the book [Op [(that) John threw out t]]

b. Parasitic gap adjunct: [Op [without PRO reading t]]

He also draws on the intuition that adjuncts are interpreted via predicate modification (PM). As a result, in order for the above configuration to be interpretable, the sister of the adjunct must have the same type, namely it must also have type $<e,t>$ (in need of saturation by the wh-element). These conditions are both easily met if the subject is internal to the sister of the adjunct clause, as in (16). Wh-movement in the main clause through Spec *v*P leads to lambda abstraction over the verb phrase. As long as the adjunct merges below the wh-element but above the subject, predicate modification and subsequent saturation proceed straightforwardly.

(16) Adding semantics to Nissenbaum's configuration



As a result, Nissenbaum's analysis of parasitic gap licensing requires the opposite of Landau's specifier ordering. Let's now explore Landau's reasons for putting specifiers in that order to see whether any common ground can be found. To foreshadow, Landau independently needs to permit specifiers to move freely to create his configuration, in which case his results may not be contingent on having a particular base order of Merge.

2.4.1 *Why Landau's specifiers can be flexible*

Landau 2021 takes a significantly different approach to the interpretation of control adjuncts compared to Nissenbaum. While Nissenbaum proposed that they compose via predicate modification (like other modifiers), Landau proposes that the adjuncts we have been looking at compose with the matrix clause via functional application – he assigns adjunct heads a higher type, which first selects its own clause and then the matrix clause.[4] On his view, adjuncts are not uniform in type either; OC adjuncts and NOC adjuncts are headed by elements of different types with different selectional requirements.

4. It is important to note that Landau does not propose this analysis for *all* control adjuncts, just those that he describes as alternating between OC/NOC. Adjuncts that can only receive a strict OC interpretation are treated differently, but we don't discuss those here.

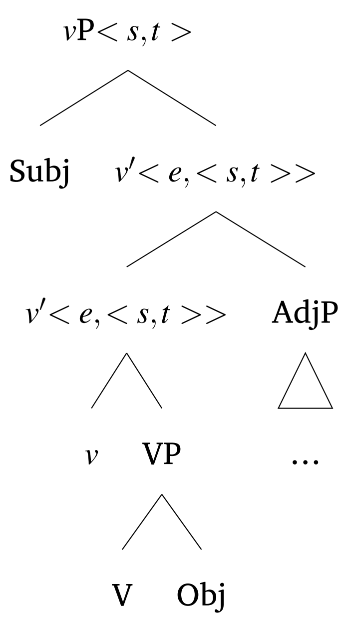
(17) Landau's semantic types for OC/NOC adjuncts

	OC	NOC
Adj head	$<<e,<s,t>>,<<e,<s,t>>,<e,<s,t>>>>$	$<<s,t>,<<s,t>,<s,t>>>$
AdjP	$<<e,<s,t>>,<e,<s,t>>>$	$<<s,t>,<s,t>>$

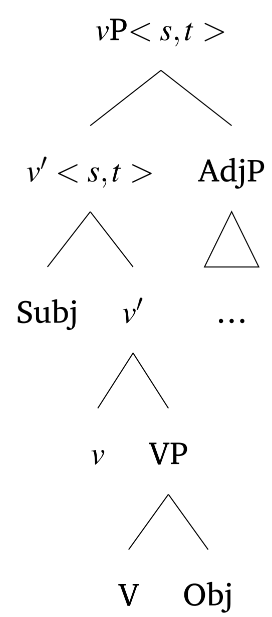
Due to the semantic types he assigns, OC adjuncts have to combine with the main clause before the matrix predicate has merged the subject. By contrast, NOC clauses have to merge after the subject.

(18) Landau's OC vs. NOC

a. OC: subject is second specifier

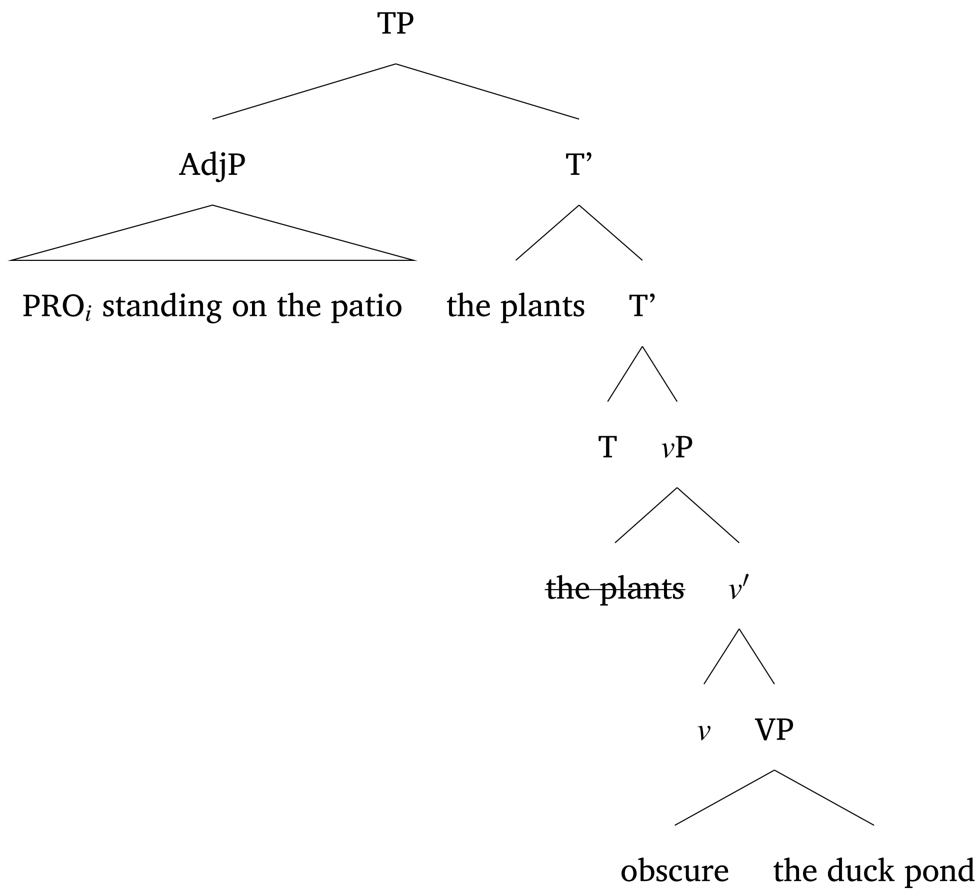


b. NOC: subject is first specifier



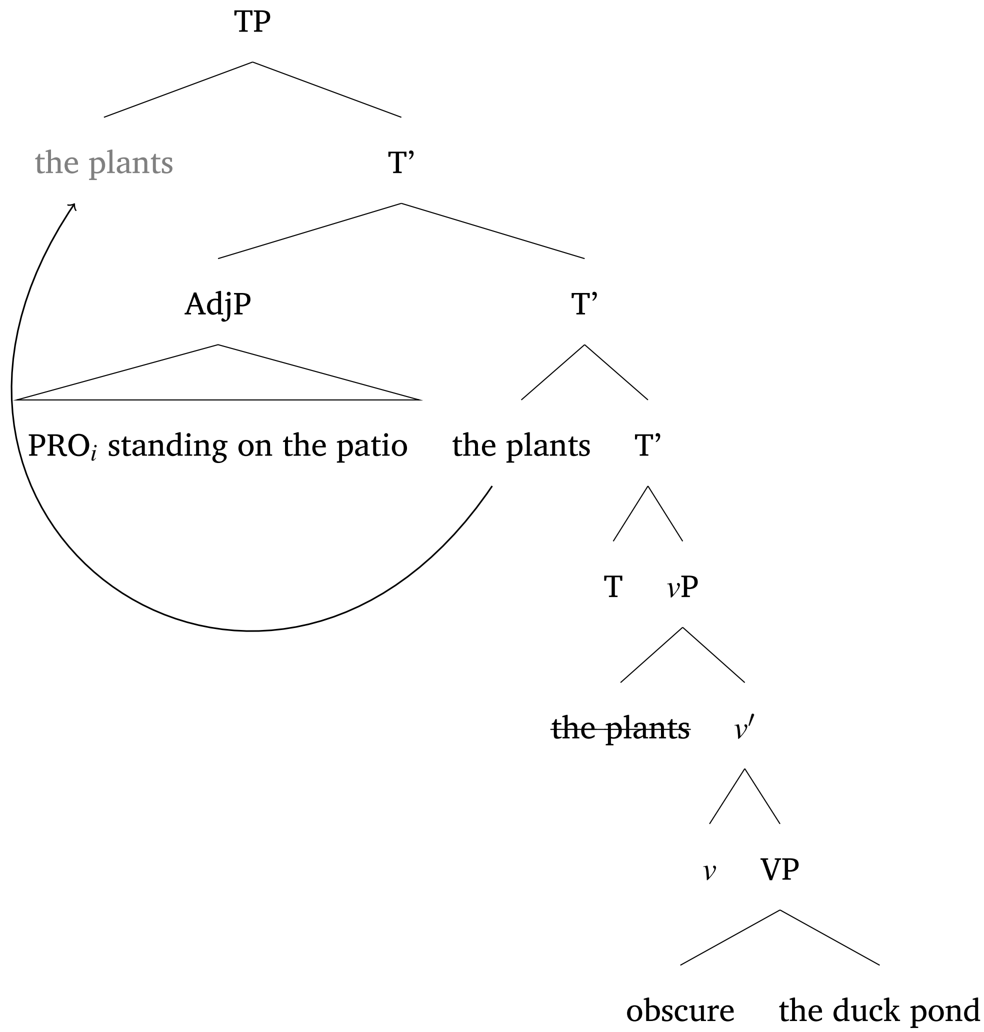
Landau discusses a challenge from clause-initial OC adjuncts, however. In (19), the OC adjunct appears to c-command the surface position of the subject, despite the fact that its semantics should require it to adjoin to the matrix clause below the subject.

(19) [PRO$_i$ standing on the patio], the plants$_i$ obscure the duck pond.



To capture cases like these, Landau proposes that LF movement of the subject applies, taking the subject from its surface position to a higher position to feed the semantics. This movement is both unpronounced and insensitive to syntactic rules that might prevent a specifier from moving to a new specifier position of the same projection.

(20) [PRO$_i$ standing on the patio], the plants$_i$ obscure the duck pond.



Given this amendment, it isn't obvious that Landau's semantic approach actually restricts the order of specifiers in the narrow syntax. In other words, if our analysis is right that OC adjuncts are generated above the subject (as in Nissenbaum 2000), which is what allows dependencies into the adjunct clause, Landau's analysis could be easily made compatible with our approach by just assuming that his LF movement applies whenever necessary to get the semantics right.

In sum, given that OC adjuncts are sometimes clearly above the subject, and any theory of control must be able to account for this, we will simply assume that (12) is the baseline configuration for OC: the controller is the first specifier and the PRO-containing adjunct is the second specifier. This follows Nissenbaum's proposed structure, and may not affect

Landau's semantics if we permit LF movement.[5]

The takeaway is that the *syntax* enforces the specifier order in (12), whenever the adjunct receives an OC interpretation or permits wh-extraction/parasitic gaps. Analogously, we expect a transparent subject in the melting cases to be transparent only if it is a second specifier of *v*P.

3. Moving towards paths

Thus far, we have observed that multiple specifier environments can obviate the CED. Furthermore, we have proposed that it is *second* specifiers, rather than first specifiers, which become exceptionally transparent in these environments. To understand why second specifier position might be special in this way, we suggest that the kinds of dependencies under consideration (wh-movement, control, PG-licensing) are governed by locality principles that constrain Search.

On this view, there must be something special about second specifiers that makes their contents searchable by a higher probe, while first specifiers are opaque. We propose a particular approach to Search and probing that makes this so, which is grounded in a theory of feature projection. On our view, principles of feature projection make the features of second, but not first specifiers visible to higher heads.

The coming sections will describe the analysis in detail, but we first want to clarify why we chose this approach. This analysis is a preliminary attempt to make use of existing probing tools to understand an unusual empirical pattern. The challenge with this kind of pattern is that it refers to specifier number in a way that syntax should not be able to do. We know that grammars are not able to "count", and as such should not be able to treat specifiers differently from each other in a way that references only their number. The present approach therefore tries to single out what is unique about the structural context of a second specifier and exploit that in the analysis. As we will show later, we

5. At this point, it is still not obvious whether Landau's assumptions about the semantics of control can be made compatible with Nissenbaum's when it comes to parasitic gap licensing in control adjuncts.

hypothesize that second specifiers are special due to how the feature projection algorithm labels projections that already contain a specifier. While there may be other ways to approach this locality profile, we think this one has the advantage of making nuanced predictions with existing mechanisms.

3.1 An interlude on Search and Path

This paper develops a novel theory of locality rooted in the notion of *path* (Kayne 1981; D. M. Pesetsky 1982; McFadden and Sundaresan 2019). Long distance dependencies, on this approach, must be mediated by a sequence of local dependencies.

(21) **Path**

For a probe A to enter into a dependency with a licit goal B, there must be a path of local relationships between A and B.

We propose that, underlying this notion of *path* is a more general notion of *economy* in dependency formation. Paths, as we define them, allow the grammar enough information to know whether or not a search procedure, on which long-distance dependencies are contingent, will succeed or fail.

Chomsky (2004, et seq.) suggests that *probing* involves an operation of "Minimal Search". Minimal, for the purposes here, means that the search procedure will halt once a match has been found, the hope being that the right specification of the search procedure will capture Relativized Minimality effects (Rizzi 1990). A number of algorithms for Minimal Search have been proposed and discussed in the literature, see Branan and Erlewine (2021) for an overview and Preminger (2019), Ke (2019), Atlamaz (2019), Krivochen (2022), and Chow (2022) for more specific proposals. The basic idea is that nodes in a syntactic tree are sequentially "examined" to see if they are a match for what the probe is specified to look for, with the sequential search algorithm only being able to move to sisters or daughters of failed matches.

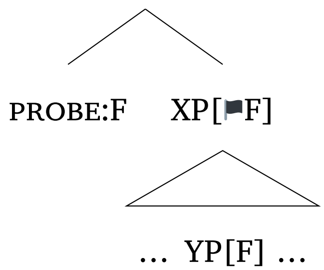
Why should Search be minimal? One reason — as Chomsky suggests — might be for reasons of computational efficiency. Searching the tree involves *examining* a number of nodes to see if they are a match for the probe. We can define a cost in terms of the number of failed examinations that take place prior to success. Examining as few nodes as possible would be desirable, given that the process of examining a node to see if it is a match bears some computational cost.

With this in mind, consider a scenario in which a probe is destined to fail because there is no corresponding goal anywhere in the structure. The probe's failure cannot be determined, and the derivation cannot proceed to the next step, until every node in the tree is examined. In terms of computational cost, this is the worst case scenario: every node in the tree must be examined, but doing so does not produce any observable change to the structure.

We suggest that the grammar is designed to avoid costly failed searches of the type above.[6] In the abstract, we suggest that the grammar is endowed with a set of *flags* that provide (limited) information about the makeup of a constituent. For instance, in the case of a probe specified for a feature F, the daughters of a node will only be examined by the search procedure if the node itself bears [⚑F].

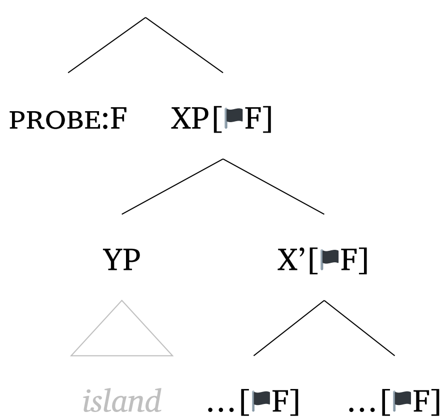
(22) **The probing configuration**

6. This question is ultimately orthogonal to the question of whether or not probing may fail without leading to a derivational crash (see Preminger (2014) for some discussion), and is in principle compatible with either view. Failed searches are consistently costly because they require the entire search space of a probe to be exhausted: each node must be examined to see if it matches the needs of the probe, and the examination of each node is that which bears the cost. In a world where probing may fail, knowing that a particular instance of it will fail allows the derivation to proceed to the next step without incurring the cost associated with Search. In a world where failed probing leads to crash, knowing that a particular instance of it will fail allows the derivation to be thrown out without incurring the additional cost discussed above.



A desirable consequence of this is that it provides a new perspective on island effects: (some) islands, on this approach, would simply be phrases that lack a flag for the relevant sort of feature. As schematized below, the internal components of a node lacking [⚑F] will not be subject to Search. In the case below, probe-goal relationships for [F] will be impossible into YP, since it does not bear the relevant flag. Note that this entails a local relationship between a probe and its goal: every node in the sister of the probe that dominates the goal must bear a flag for a feature on that goal.

(23) **An island configuration**



This raises a number of questions, the most pressing of which we hope to answer. In what follows, we suggest that the *flags* in (23) are checked selectional features — presumably an independently necessary component of the grammar.[7] Checked selectional

7. This proposal has some surface similarity to H/GPSG (see e.g. Gazdar 1981) approaches to long-distance dependencies. On both proposals, some information about the relationship between heads and

features provide a record of the derivation — the presence of a checked selectional feature on a maximal projection serves as a *flag* that either the specifier or complement of that phrase is of a particular sort. The chief innovation is an algorithm for determining whether or not checked selectional features are able to project past the maximal projection of the head they originated on. Crucially, this decision is *local*: it creates paths of local relationships between a probe and a licit goal, in the sense of (21). We show that the theory captures the basics of the classic CED: adjuncts and specifiers are, in the basic case, opaque for extraction, while complements are not. We show also that the theory avoids what we term the "escape hatch problem" for phase-based approaches to the CED, a stipulation which requires adjunct islands to both be phases and consistently lack an edge feature.

3.2 Feature checking and feature projection

As discussed in the previous section, we propose that a notion of *path* mediates Search. As Search underlies the establishment of long-distance dependencies, paths become preconditions for long distance dependencies by extension. One of the consequences of a path-based approach to Search is that there are many scenarios in which Search may fail at the outset, before it has examined any nodes. For example, if the sister to a probe does not bear a flag for the relevant feature, the probe won't bother to search its sister's daughters at all, given that there is no path of checked selectional features leading from the probe to its sister's daughters in this case.

For a probe A to establish a dependency with a goal B, then, as a minimal first step, A must be able to identify the relevant flag on its sister before continuing the search. An algorithm for projecting features checked by B from the head that selected B to the

their arguments is projected up the clausal spine to a relevant probe. Our theory uses this idea differently, however, because we assume that dependencies may be formed through movement, contra H/GPSG, where dependencies are always formed through external Merge. As a result, the information that is projected is a *checked* selectional feature rather than something representing an unmet selectional requirement (e.g. the *slash* in GPSG). In that sense, dependencies in our theory crucially rely on a notion of internal Merge — without a previously merged instance of some phrase, a path of the relevant checked selectional features could never be created.

sister of A establishes a series of local relationships that links probe and goal. A projection algorithm thus ensures that if a feature has reached A's sister, it must have been projected at every node between A's sister and the goal B, thus creating a path between the two. We can thus operate with a shorthand definition of *Path*, shown in (25). Here we will discuss the predictions of the approach for dependencies involving selected elements. We return to the extraction of unselected elements, e.g. adjuncts, in §4.

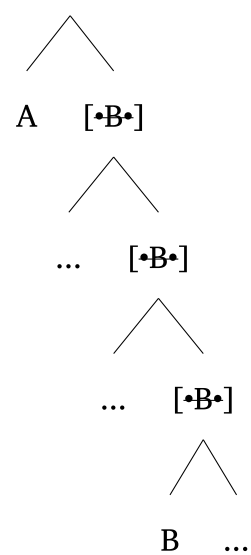
(24) **Accessibility**

A probe A searching for a goal B may only initiate Search for B if there is a path from A to B.

(25) **Path** (shorthand)

There is a Path from A to B if A's sister bears a feature checked by B.

(26) A long-distance path from A to B



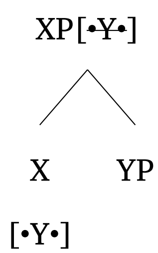
If for some reason A's sister does not bear a feature checked by B (i.e. because there is no local B), Search fails at the outset, without examining any nodes in the tree. Thus, Search never applies unnecessarily.

We begin by establishing some assumptions about clause construction, and show how a modified theory of feature projection creates long-distance dependencies according to (24) and (25). Adopting the notation of Müller (2010), we represent the features that drive Merge as in (27). A head that selects for a YP, for example, might bear a feature [•Y•], which may be checked when that head (or a projection of it) merges with a YP. In other words, a head with an unchecked [•Y•] feature that merges with a YP produces a projection bearing the checked version of that feature, [~~•Y•~~], as in (28). As will become important later, we follow Heck and Müller (2007), Müller (2010), Longenbaugh (2019), and Newman (2024) in assuming that these features may drive *any* kind of Merge, representing not only external Merge, but movement (internal Merge) as well.

(27) **Merge features**

[•Y•] = an instruction to Merge with an element bearing Y

(28) **Selection for YP**



These checked features are inactive in the sense that they may no longer drive syntactic operations. In other words, [~~•X•~~] does not count as an element that bears [X], and thus cannot feed X-merge. Similarly, [~~•X•~~] does not count as a selectional feature, which would require checking by an element bearing [X]. It simply suggests that X-merge has taken place. However, contra Chomsky (2000), Adger (2003), and Asudeh and Potts (2004), a.o., we suggest that these checked features do not disappear from the derivation. Instead, we propose that they remain present throughout the computation to serve as a pseudo-record of selection. On this view, XP provides more information to higher heads than just its own category feature. It also bears checked Merge features, which tell higher heads something about the elements inside XP, for instance that XP contains a YP in the case of (28).

So far, we have seen how checked features may be projected by a head to its own maximal projection, when it merges with elements it selects for. These features do not delete, and are thus visible to whatever subsequently merges with XP. According to the conditions in (24) and (25), for YP to be accessible to anything beyond XP's sister, [~~•Y•~~] must be able to project *past* XP. Only if [~~•Y•~~] projects past XP can it ever appear on the sister to a higher probe, making YP accessible to that probe.

We propose that feature projection past maximal projections is conditioned by the local context of that maximal projection. More specifically, maximal projections whose sisters are what we call *Indivisible Feature Bundles* get to project their checked selectional features to higher nodes, making their contents accessible to later operations (29). Indivisible feature bundles are defined in (30) – they are essentially nodes whose features locally come from a single source. The intuition guiding this approach is the belief that language is binary: at every step of the derivation, feature projection should only project two bundles of features at a time. If one sister already locally projects two feature bundles, the other cannot project at all. If one sister locally projects one or fewer feature bundles, the other can project one as well.

(29) **Checked Feature projection**

A feature bundle {[~~•F•~~],[~~•G•~~]...} on a maximal projection may project iff its sister is an *indivisible feature bundle*.

Features of non-maximal projections may always project.

(30) **Indivisible feature bundle**:

a. a feature bundle that comes straight from the lexicon

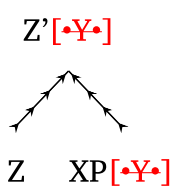
→ e.g. a terminal node (Matushansky 2006), OR

b. a feature bundle that has projected to a node from only one daughter

On this view, lexical items are always indivisible feature bundles. Thus, sisters of lexical items (i.e. complements) will always be permitted to project their checked selectional features. Non-terminal nodes, by contrast, might or might not be indivisible feature bundles – that depends on whether their daughters were allowed to project their features. Thus, sisters to non-terminal nodes (specifiers/adjuncts) might or might not be allowed to project their features.

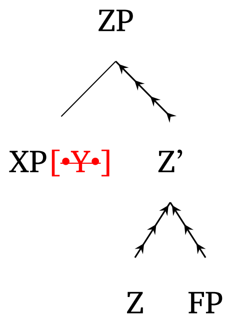
Recalling the XP maximal projection from (28), we can now calculate the predicted effects of context on whether XP gets to project its [•~~Y~~•] feature to higher nodes. If XP is the first-merged element with a head (othwerwise known as a *complement*), as in (31), the rule in (29) states that XP can project its [•~~Y~~•] feature – its sister is an indivisible feature bundle.

(31) XP projects [•~~Y~~•] to a higher node if it is a complement



If XP is the second-merged element in ZP, i.e the first specifier of ZP (32), its sister is *not* an indivisible feature bundle. The Z′ sister to XP projects from two daughters: the terminal node (which always projects) and its sister (complements get to project, according to (31)). Since Z′ is not an indivisible feature bundle, XP does not get to project [•~~Y~~•] in this context, rendering YP inaccessible to operations external to XP. The theory thus accounts for basic CED effects: complements permit subextraction but first specifiers do not.

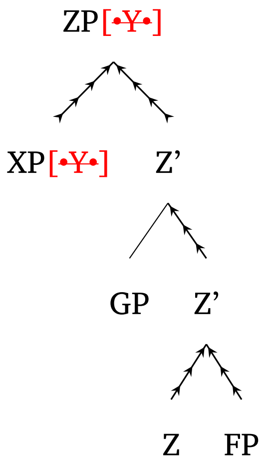
(32) XP does not project [•~~Y~~•] to a higher node if it is a first specifier



The theory makes a surprising prediction for third-merged elements, however. If XP merges as a second specifier (33), rather than a first specifier, its sister now only locally projects from *one* daughter. Its sister may bear features that originally came from multiple sources, but the notion of indivisibility that we pursue only examines a node's local context, namely whether it projects from its immediate daughters. On this view, the sister to XP in this case only projects from one daughter because first specifiers cannot project, according to (32). The first node that dominates a first specifier is therefore an indivisible feature bundle according to (30b), which licenses projection from a second specifier.[8]

(33) XP projects [•~~Y~~•] to a higher node if it is a *second* specifier

8. The present discussion assumes that Merge and feature projection proceed cyclically: each successive specifier extends the clause, and the feature projection algorithm follows the order of Merge. This may not be a necessary feature of the system, however. If multiple specifiers instead *tuck in*, as in Richards (1997), we could imagine reformulating the approach so that the feature projection algorithm and its consequences for Search inform representational constraints on movement rather than derivational ones. In a world with tucking in, properties of a phrase's sister could still inform whether that phrase projects, but projection would not proceed according to the order of Merge, but rather according to the resulting constituent structure.



This approach therefore generates an on-again off-again profile. If some maximal projection is allowed to project, it often creates a context in which the next merged maximal projection cannot project. If a maximal projection does not project, it often creates a context in which the next merged maximal projection can project, and so on. Thus, we expect the time of Merge to determine transparency for higher operations more so than the complement-specifier distinction. We leverage this context sensitivity to explain the variable opacity of adjuncts and specifiers in different contexts.[9]

In sum, we propose that the distribution of checked features on nodes creates paths between probes and goals, where paths are a precondition for Search. A probe whose sister bears a feature checked by its goal may initiate Search for that goal, in which each successive node is examined for features checked by the goal until the goal is found. If the probe's sister has no relevant checked features, Search fails before it starts, avoiding unnecessary and costly searches. We proposed that the distribution of checked features is controlled by the rules of feature projection outlined here: maximal projections may

9. A question arises: what happens in the case of "simple" phrases, e.g. phrases that have themselves failed to select anything? One approach would be to deny the existence of simple phrases of this sort: on this view, every functional item would enter into some sort of selectional dependency with something else, while lexical items would minimally consist of a root and categorizing head (see Marantz (1997) for a proposal along these lines). Another approach would be that they fail to project features, which could potentially have consequences down the line if the phrase that they are a complement of later takes a specifier: the first specifier in this case should be allowed to project in the way a complement normally would. We leave investigation of these possibilities to future research.

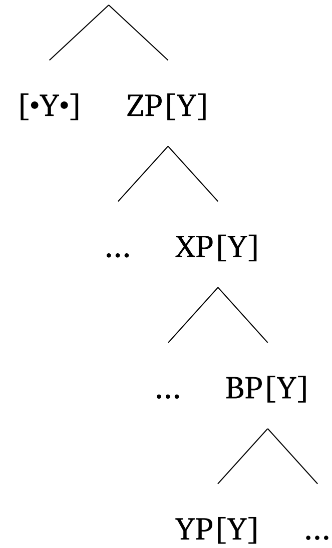
project their checked features if their sisters are indivisible feature bundles but not otherwise. Successive projection of checked features creates paths.

It is important to recall that [•~~Y~~•] is not equivalent to the YP that checked it. [•~~Y~~•] is a feature that was checked/rendered inactive by an element bearing Y. By contrast, YP is a phrase that can check some set of features on a probe, including [•Y•]. Thus, a probe whose sister bears [•~~Y~~•] has not "found" YP before searching, because [•~~Y~~•] can never feed syntactic operations like Y-merge/agree. The probe must still search for the YP that checked the feature in order to satisfy the probe.

At this point, one might wonder why we don't simply invoke the feature [Y] in path formation, instead of checked selectional features like [•~~Y~~•]. If we had a projection algorithm that projected instances of [Y] (i.e. the property of Y that makes it a goal for a [•Y•]-probe), then the resulting paths might look like (34), in which [Y] projects to ZP and beyond. The problem with an alternative like this comes from the literature on pied-piping: if a higher head seeks [Y], minimality considerations will cause that probe to attract/agree with the highest instance of [Y] in the structure, rather than the phrase [Y] originated on. The result should therefore be a kind of pied-piping: ZP should move/agree, where ZP contains the original YP, but the YP itself should not be available for subextraction due to minimality considerations. Thus, invoking [Y] instead of [•~~Y~~•] would produce a feature-percolation-type approach to pied-piping, and would not produce a theory that allows the intended goal to move on its own.[10] We therefore need some other feature to mediate path formation, one that does not intervene for the dependency being formed. Checked selectional features do just this, without adding to the existing typology of features: they record what kinds of phrases are dominated by a certain node, without making that node an intervener for the goal of the probe.

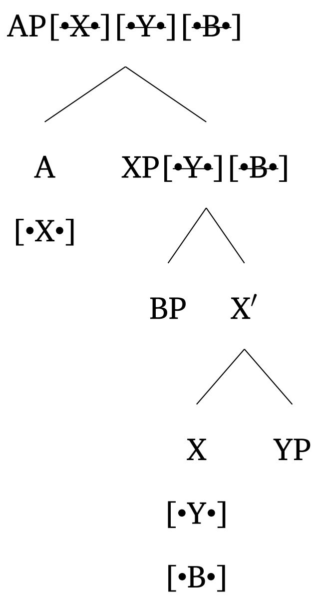
10. For arguments against a feature-percolation approach to pied-piping, see Heck 2008; Cable 2010, 2012.

(34) If [Y] projected instead of [~~•Y•~~]



Lastly, though the presentation here only discussed a case where *one* checked feature was projected, we assume that feature projection is *wholesale* in general. What we mean by this is that multiple checked features on a phrase get projected together as a bundle – a maximal projection cannot selectively project some of its features but not others, as illustrated in (35).

(35) **Projection is wholesale**



Because projection is wholesale, we expect maximal projections to be opaque or transparent to higher operations in a very general sense. To reiterate, the tree in (35) illustrates how these projection rules create long-distance dependencies: if XP is selected as the complement of A, then the newly-formed AP not only bears a [•X•] indicating the presence of an XP inside it, but also inherits any checked features borne by XP itself, allowing whatever selects for AP to search into XP for YP and BP.[11] A transparent maximal projection is transparent for potentially multiple dependencies across itself – it projected every feature it had, so everything inside it that checked a projected feature is visible to higher heads. An opaque maximal projection is similarly opaque for every imaginable dependency – if a maximal projection projects no features past itself, there can be no paths leading into it. We will see that this all or nothing approach captures correlations between different dependencies that cross adjunct boundaries.

Before moving on, we will clarify certain assumptions that we make about features that drive syntactic operations, which underpin much of the discussion that follows. We assume that certain elements bear a [D] feature, which marks them as an argument of a clause. This is the same feature which allows certain elements but not others to satisfy the EPP in English, and bears similarity to the Case feature of van Urk and Richards (2015), and the φ feature assumed in van Urk (2015) and Longenbaugh (2019). We use [wh] as a feature for elements that enter into Ā-dependencies.

11. It is worth noting that the "wholly-tranparent"/"wholly-opaque" nature of certain domains is not inherent to the theory developed here, but only if the wholesale nature of projection is assumed. We could, of course, imagine more elaborate theories of feature projection that don't require wholesale feature projection of the sort here. The consequence of this would be that some domains would be transparent for some dependencies but not others (see Keine (2019) for some discussion of such patterns). We acknowledge this here as a point of interest for future work, but do not develop such elaborations here beyond what has here been said. For now, we will proceed with a fairly simple subtheory of feature projection, to highlight the — to our mind interesting fact that the "parity" of specifiers/adjuncts determines whether or not they may project — and acknowledge that a more intricate subtheory of feature projection might make more intricate predictions.

4. Dependencies through Paths

We suggest that variable projection from specifiers and controlled adjuncts accounts for their variable transparency to OC, wh-movement, and parasitic gap licensing. As discussed in §2, we observed (following Müller, Nissenbaum, Landau, etc.) that violations of the CED often occur in multiple specifier environments. We furthermore concluded with Nissenbaum, contra Landau and Müller, that second specifiers were always exceptionally transparent.

To understand this pattern, we first propose that *wh*-movement, parasitic gap licensing, and OC control involve the establishment of a long-distance syntactic dependency through Search. Illustrating the proposal for the wh-movement/OC correlation, we propose that *wh*-movement arises when interrogative C searches its complement for a phrase bearing a [wh] feature, which is used to check a [•wh•] feature on itself. OC likewise involves a syntactic dependency formation that is also contingent on successful Search, where the complement of a potential binder for PRO is subject to Search for PRO (see Ke 2019 for a similar proposal for reflexive binding). Consequently, there must be a *path* between PRO and its controller for OC to arise, and a path between interrogative C and a *wh*-phrase for movement to occur.

As proposed in §3, such path formation is contingent on successful projection of checked selectional features from the sister to the goal to the sister of the probe. The proposed algorithm for feature projection predicts that features from first specifiers do not project to their mothers, making the contents of first specifiers opaque to higher operations. The features of second specifiers, by contrast, *do* project, creating paths into them. Thus, second specifiers are predicted to be transparent for dependencies like wh-movement and control, accounting for the facts in §2.

Importantly, when an adjunct/specifier projects its features, there are paths into it for every feature that it projects. As a result, when an adjunct is transparent for control, it is also transparent for other dependencies like wh-movement. In what follows, we discuss the specific features we have in mind for each of these dependencies and explore some other multiple specifier contexts.

4.0.1 *Which features/paths?*

On the theory developed here, the requirement for there to be a path between two elements "linked" through Search should be seen as a way to ensure that Search will succeed. We now explain how Search interacts with movement to account for the correspondent transparency effects. Recall, as discussed beforehand, that both internal and external Merge are licensed only when they check [•F•]s. Movement — or internal merge — requires an invocation of Search on the sister for some matching feature, followed by Merge of the goal at the root of the tree.

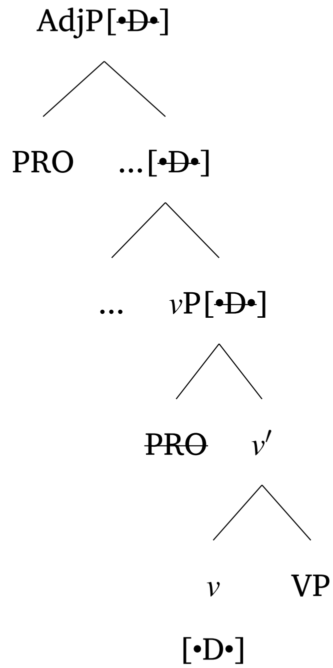
Not only must there be a path of checked features between the two elements in question, but the target of Search must have checked the sort of feature that Search is looking for. In other words, a probe with a feature [•X•] must find a path of [•~~X~~•] features to its goal, not just any path of features checked by its goal.[12]

At this point, one might wonder *which* features actually establish these paths between the matrix subject/PRO and C/the wh-element, and how those features get checked/projected. For PRO, the answer is straightforward: PRO is presumably selected as the external argument of the adjunct clause. We can therefore imagine that it checks a [•D•] feature on adjunct *v*, which gets projected to the highest node in the adjunct clause. We have illustrated PRO as the highest specifier of AdjP on the assumption that PRO moves to the edge of its clause (Heim and Kratzer 1998). As long as control is mediated by a search for DPs, if that [•~~D~~•] feature projects to the sister of the matrix subject, the matrix subject may find and control PRO.[13]

12. Note that the Search-based system developed here includes, but is not limited to, canonical probe-goal relationships. For the instances of operator binding and control of PRO, we could equally well assume that these elements are subject to a well-formedness constraint requiring them to be local to their binder. The idea, then, would be that that the evaluation of this condition would done through Search. Consequently, we would expect these syntactic relationships to display the same locality profile as probe-goal relationships that trigger movement operations.

13. An equivalent alternative is that whichever feature attracts PRO to the edge of the adjunct clause is

(36) PRO checks [•D•] on *v*, which projects to AdjP



For wh-elements, we propose that their visibility for wh-movement is regulated by the distribution of heads bearing [•wh•]. On the assumption that all phase heads have the necessary machinery for hosting successive cyclic wh-movement, we propose (following Longenbaugh 2019; Newman 2024) that these probes for movement are represented as Merge-inducing features specified to be checked by wh-phrases: [•wh•]. If heads like *v*, C, and possibly D are endowed with such features, then the manner in which [•wh•] gets checked and projected as [~~•wh•~~] is as follows: wh-phrases could undergo a step of [•D•]-driven movement to the specifier of an intermediate phase head in the clause, before moving to Spec CP.[14]

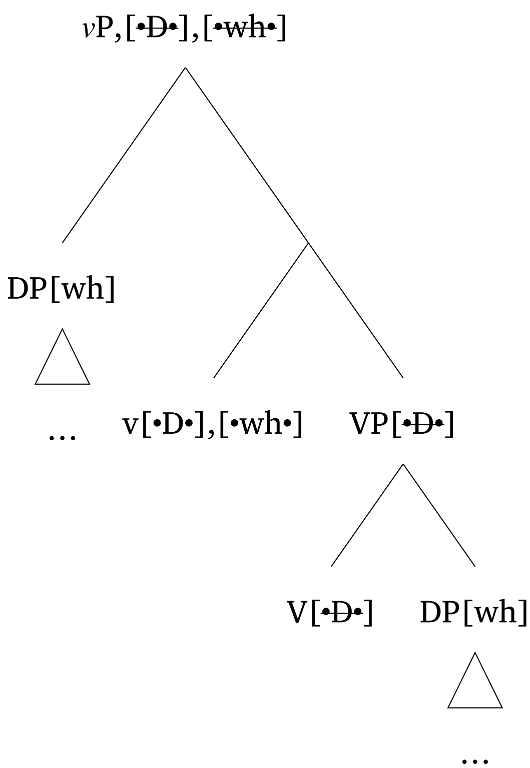
For concreteness, consider the case below. Here, *v* bears both a [•D•] and [•wh•]

what establishes the path between the matrix subject and PRO. If that feature is also [•D•], however, there is no meaningful difference between the two options. If some other feature is responsible for adjunct-internal movement of PRO, then some other feature could be responsible for the control path, but we won't speculate about what that feature could be here.

14. We do not rule out the possibility that other heads (e.g. V) have [•wh•], in which case wh-objects could just check [•wh•] upon being selected by V. We pursue the present option to show that the account is also compatible with a more restrictive theory of the distribution of [•wh•], in which only phase heads have access to such features.

feature. Internal merge to satisfy [•wh•] is not possible: the complement of *v* does not bear a [~~•wh•~~] feature, so it may not be searched for [wh]. The complement does, however, bear a [~~•D•~~] feature, so it may be searched for an element bearing [D], in which case the object will be found. Subsequent merge of the object in Spec *v*P will check both the [•D•] as well as the [•wh•] on *v*. Consequently, the [~~•wh•~~] will be able to project higher in the tree from this *v*P, creating a path between the wh-phrase in Spec *v*P and higher elements in the tree.

(37)



For the sake of having a concrete analysis, we will adopt this approach here, though this may not be the only possible solution. The present approach, however, has the advantage that it has precedent in the literature from Canac Marquis (1994), van Urk and Richards (2015) and Longenbaugh (2017). The first two propose that Ā-movement chains involving non-subjects always involve a step of A-movement within VP. Canac Marquis (1994) suggests that English Ā-movement of objects is analogous to *tough*-movement, involving A-movement of a null operator to the inner specifier of the phrase that the element undergoing Ā-movement is introduced to as an outer specifier. This step of operator movement allows the moved element to be linked to its gap; the position which the moved object is itself initially Merged in is presumably motivated by a need to check the [wh] feature of the relevant projection.[15]

van Urk and Richards (2015) propose that *wh*-movement of objects in Dinka (Nilotic; Sudan) is parasitic on a prior step of A-movement to a position low in the clause, based on facts about the exceptional absence of an internal argument in an otherwise obligatorily filled preverbal slot, in contexts involving Ā-movement of an internal argument. The idea, presented using the feature ontology assumed throughout, is that movement to this preverbal position may in principle check both [•wh•] and [•D•] features. In the absence of an element which bears [wh], this is driven solely to satisfy the needs of [D] on the relevant head. When an element bears both, checking of the [•wh•] feature on the head that triggers movement may take place, with the path licensing movement involving a [•D•] feature.

Longenbaugh (2017), like Canac Marquis (1994), discusses a derivation like this in the context of English tough constructions, though the details of the analyses differ. On Longenbaugh's view, tough movement involves successive cyclic *composite* movement through Spec *v*P, which has both A and Ā-properties. On the present view, this "mixed" movement is represented by the multiple checking of two different features on *v*: [•D•] and [•wh•].

The proposal that wh-movement is mediated by DP-movement raises questions about wh-movement of non-DPs, such as PP arguments and adjuncts.

15. One might wonder why, if wh-phrases may undergo movement to an intermediate position before moving to Spec CP, this intermediate position is not occupied by a moved phrase in non-interrogative contexts. This is a challenge for theories of successive cyclicity, in which non-interrogative phase heads must bear the necessary machinery to host wh-movement, but only if there is an interrogative C somewhere else in the structure. We assume that many of these cases will be ruled out at the interfaces. Assuming that a wh-moved element needs to be interpreted within the scope of an interrogative element, if there is no such element, then perhaps the wh-phrase isn't licensed, regardless of where it has moved. As for cases with multiple wh-phrases, it may be a matter of conditions on pronunciation that force the lower element to appear in its base position, even if it has covertly moved to a higher position.

(38) a. To whom did John first speak?

b. On which day did John first speak?

For adjuncts, a fairly straightforward analysis would be to propose that they consistently externally merge in Spec *v*P, at least in cases where they undergo *wh*-movement. For argument PPs the way forward is less straightforward. One possibility is that wh-PP arguments, like adjuncts, often have the option of initially merging with a functional head like *v*P, ensuring that the PP argument checks a *wh* feature (see Newman 2024 for a proposal along these lines). Another possibility is that PP arguments are required to exit the VP for independent reasons (see Stowell 1981 for such a proposal). Assuming this movement is feature-driven, subsequent Merge of a PP argument with *v*P consequently checks v's [•wh•] feature in cases where the PP bears [wh].

In sum: both PRO and *wh*-elements must respectively check [•D•] and [•wh•] if they are to be visible for subsequent Search operations. In the case of PRO, this is relatively trivial: PRO checks a [•D•] when it is initially merged. In the case of *wh*-elements, this means that the *wh*-element must first undergo movement for independent reasons to an intermediate projection, with checking of [•wh•] on this intermediate position taking place as a side effect. Only after movement to such a position will there be a path of [~~•wh•~~] features to the *wh*-element, rendering it visible for subsequent Search.

Before moving on, we want to address another dimension to this issue of which features to invoke in path-formation. Here we have focused on the *type* of feature, for example whether it is of category [D] or [wh], etc. However, one could ask about *tokens* of features as well. Does the derivation track *which* DP a given [~~•D•~~] corresponds to?

The two options make different predictions when the matrix verb is transitive. In a transitive clause, regardless of the position of the external argument within Spec *v*P, a [~~•D•~~] from the internal argument should project to every node within *v*P, including sister to the adjunct. If path-formation only cares about the presence of [~~•D•~~] on nodes, and doesn't care whether that feature was actually created by the controller, then we might expect all adjuncts that modify transitive clauses to be OC adjuncts. However, NOC is available in (39), suggesting that there is a derivation available with no path to the adjunct.[16]

(39) The bus has a seatbelt [PRO_{arb} to wear].

This result could teach us either of two things: 1) maybe we need checked features to be indexed with the element that checked them, where paths are sensitive to these indices, or 2) maybe heads that select for DPs suppress existing instances of [•Đ•] from their sisters, breaking the chain of [•Đ•] until they introduce their own arguments. For concreteness, we assume the first possibility throughout, but we are optimistic that the proposal could also work without indices.

4.0.2 *Stacked adjuncts: a loose end*

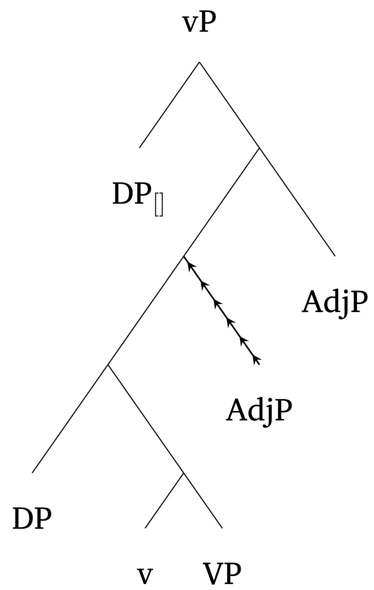
Before moving on, it is worth tying up a loose end. As noted by Nissenbaum, one instance of *wh*-movement may license parasitic gaps in more than one adjunct.

(40) Who will you hire ____ [after interviewing ____] [if they recommend strongly ____]?

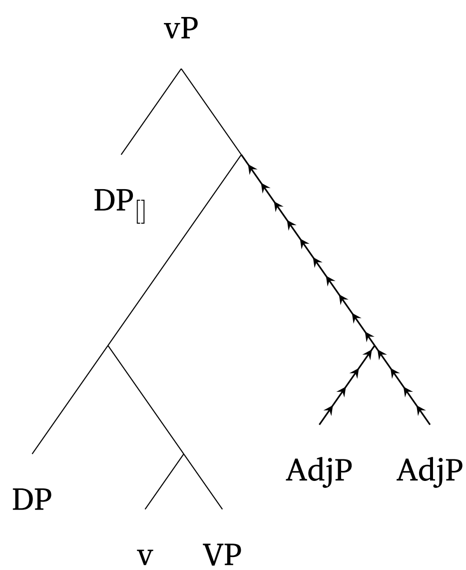
As pointed out by a reviewer, an analysis of stacked adjuncts like (41) poses a potential problem for the theory developed here — only one of the two adjuncts on such an analysis would be of the right parity to project its features and license the parasitic gap within. We believe there is good reason, then, to favor an analysis like that in (42) for "stacked" adjuncts of the sort discussed here, following Chomsky 2019. In this structure, the two adjuncts Merge with each other prior to their Merge with vP. Since the "combined" adjunct is an even parity specifier of vP, it will be able to project its features from this position. Furthermore, since the adjuncts that make up the combined adjunct themselves have a specifier — the null operator which gives rise to the parasitic gap — they will be able to project their features up to the "combined" projection.

16. Thanks to an anonymous reviewer for bringing this example to our attention.

(41)



(42)



This theory also accounts for Nissenbaum's observation that parasitic gap-containing adjuncts must appear closer to the predicate they are construed with than adjuncts that do not, as shown below.[17] If Nissenbaum is right that parasitic gap-containing adjuncts are of a distinct semantic type, then they would only be able to conjoin with adjuncts

17. A reviewer points out that our theory makes a prediction divergent from Nissenbaum involving constructions with three adjuncts. We expect such sentences like the following, where the innermost and outermost adjunct contain parasitic gaps to the exclusion of the middle adjunct, to be acceptable.

of the same type (i.e. gap-containing adjuncts can conjoin with other gap-containing adjuncts).[18] Given this, the adjuncts in (43) must form different specifiers of *v*P. The inner adjunct may be of even parity provided the Ā-moved object lands above it in vP; the outer adjunct, conversely, may not. It would have to be above both the inner adjunct and the subject, but below the wh-phrase in order to license a parasitic gap (according to Nissenbaum). From this position, however, it couldn't project, blocking a parasitic gap. As a result, when there are two adjuncts, one with a gap and one without, the adjunct with the gap must be the inner adjunct.

(43) a. *Who will you hire ____ [after interviewing someone else] [if they recommend ____]?

b. Who will you hire ____ [after interviewing ____] [if they recommend someone else]?

What we have seen, then, is that multiple kinds of dependencies which Search plausibly underlies — control and Ā-dependencies such as *wh*-movement and binding of null operators — are allowed into adjuncts/specifiers only when those adjuncts/specifiers appear in a particular context. Moreover the theory captures the fact that one and the same adjunct/specifier may be opaque or transparent, given that such elements may merge as second specifiers or not. In §5 we discuss further implications of the theory we've developed, and compare it to other theories with comparable empirical coverage.

(i.) Who will you hire ____ [without interviewing ____] [if John recommends him] [despite criticizing ____]?

This seems to be the case for a number of the speakers we consulted. We leave a fuller account of the speakers who diverge from this judgement a topic for future work.

18. See also Gould (2020) for discussion of parasitic gap data where Nissenbaum's generalization appears to hold, but his semantics do not.

5. Discussion and Conclusion

What we have seen so far is a novel theory of locality in which locality domains are determined by their local context. §2 examined several exceptions to the CED and shows that they tend to arise in multiple specifier environments. §3 developed a theory of feature projection that captures something like the classical CED, but which makes finegrained predictions about when the CED can be obviated. We showed that the theory is able to account for a number of exceptions to the classical CED, and furthermore explains a hitherto unexplained correlation between extraction from adjuncts and the possibility of a non-obligatory control interpretation for the adjunct in question. Having motivated and developed this theory of locality, we now compare our approach to previous literature, and sketch ways forward for future work.

5.1 Other Approaches to the CED

Since Cattell (1976), it has been common to treat specifiers and adjuncts as islands for extraction as a matter of definition. The CED, shown in (44), states that any non-complement should be opaque for extraction.

(44) *The Condition on Extraction Domain* (CED) (Huang 1982; Chomsky 1986; Cinque 1990; Manzini 1992):
Movement may not cross a barrier XP, unless XP is a complement.

The CED raises several questions: first, many have shown that it is not exceptionless (see e.g. Stepanov (2007) for discussion). In particular, we have just discussed examples of wh-extraction out of adjuncts and specifiers, both of which are clear violations of (44). These counterexamples refute the generality of (44), and suggest that we need a more fine-grained metric for island-hood besides the complement/non-complement distinction. Second, existing attempts to derive (44) face a conceptual disadvantage compared to the present theory.

A popular approach to the CED is to treat adjuncts and specifiers as subject to different rules than complements. For example, Uriagereka (1999), Johnson (2003), Sheehan (2013), and Privoznov (2021) suggest that non-complements must spell-out when they merge, rendering their contents inaccessible to further operations.

This approach requires some elaboration to theories of spell-out, given that complement clauses are also often proposed to spell-out at particular points in the derivation. Phases (including complement clauses) are typically assumed to be opaque to operations external to them after their time of spell-out. However, unlike adjuncts/specifiers, phasal complements are thought to have an *escape hatch*. Elements that move to that escape hatch become accessible to later operations, despite the fact that the phase has "spelled out". In order to capture the contrast between adjuncts/specifiers and complements, adjuncts/specifiers must therefore lack an escape hatch.

One could imagine several ways to encode the escape-hatch property on a phrase, such that complements have them but adjuncts/specifiers do not. For example, we could stipulate that complementation triggers spell-out of the *complement* of the phase head, while adjunction/specifier-Merge triggers spell-out of the entire phrase. Complements therefore have a specifier position which has not spelled out, while adjuncts/specifiers do not. Alternatively we could propose that edges of spelled-out phrases are always accessible, but that only certain heads have the ability to attract elements to their edge — adjuncts/specifiers routinely lack these edge features, in contrast to complements.

Both of these possibilities require us to stipulate a distinction between complements and non-complements in a way that the theory outlined in this paper does not. The present theory treats complements as first-merged elements with a head, specifiers as second-merged elements, and so on, reducing the number of ad hoc distinctions we need between different phrases. Moreover, the present theory is able to account for *variable* island-hood of adjuncts and specifiers, without stipulating special properties of those adjuncts and specifiers. Instead we propose that every phrase (complement or non-complement) is subject to the projection algorithm, which yields different results depending how many feature bundles are present on each node.

5.2 Conclusion

In sum, this paper examined a number of exceptions to the CED and offered a novel theory of locality designed to account for these exceptions. On the proposed approach to locality, a specifier or adjunct is rendered transparent or opaque based on the properties of its sister. This allowed us to explain why the same adjunct or specifier might be transparent in some context but opaque in another — aspects of context more subtle than specifier/adjuncthood determine opacity.

The proposed theory made use of a modified projection rule, which conditionally allows features to percolate higher than the maximal projection of a head. When the conditions for projection past the maximal level are met, the contents of that phrase become visible to higher probes, allowing dependencies to target them. Crucially, we proposed that probes could not search for a goal in the absence of such a *path* of features (created by feature projection). Phrases whose features get suppressed by the projection rule are therefore predicted to be opaque to dependencies crossing them.

Importantly, the projection rule does not reference the complement/non-complement distinction — it instead references the local feature context of a phrase's sister. As a result, specifiers/adjuncts are not uniformly predicted to be opaque: only adjuncts/specifiers whose local context suppresses feature projection are opaque, accounting for observed exceptions to the CED.

This approach to locality of course raises many questions, which we have not had space to discuss here. For example, we have looked at the transparency/opacity of phrases in their base positions, but have not yet considered what the approach should look like for phrases derived by movement. To be more specific, consider the melting cases discussed by Müller: Müller shows that object scrambling licenses extraction out of in situ subjects, but not those that have raised to a higher position, as diagnosed by the presence of intervening adverbs.

(45) Müller 2010, ex. 48c, 49b, p. 68-9

a. Was haben [den Fritz] **denn** [___ für Bücher] beeindruckt?
   what have the F. PRT for books impressed
   "What sort of books have impressed Fritz?"

b. *Was haben [den Fritz] [___ für Bücher] **denn** beeindruckt?
   what have the F. for books PRT impressed

This kind of a freezing effect raises several questions about our proposal, such as: How do copies of phrases interact with the projection rule? And in cases where movement alters the parity of a specifier, can a path of features created by projection from one position find those elements in their new positions? We hope to explore these and other related questions in future research.[19]

While there are certainly further details to develop, the theory as developed so far already makes interesting and nuanced predictions in a traditionally tricky empirical domain, and has the advantage of unifying CED effects with other kinds of locality effects analyzed by a Search procedure (e.g. intervention effects). As such, we believe it holds promise for a more unified approach to locality, grounded in the nature of the combinatorial system itself rather than the typology of adjunct/specifier phrases.

References

Abrusán, Márta. 2014. *Weak Island Semantics.* Vol. 3. OUP Oxford.

Adger, David. 2003. *Core syntax.* Oxford: Oxford University Press.

19. A theory of freezing would help us explain, among other things, why adding NOC adjuncts to sentences in English doesn't license melting — English subjects always raise to Spec TP, and are thus subject to freezing.

Asudeh, Ash, and Christopher Potts. 2004. “Honorific marking: Interpreted and Interpretable.” Paper presented at the Phi Workshop, McGill University.

Atlamaz, Ümit. 2019. “Agreement, Case, and Nominal Licensing.” PhD diss., Rutgers University-School of Graduate Studies.

Branan, Kenyon, and Michael Yoshitaka Erlewine. 2021. “Locality and Minimal Search.”

Cable, Seth. 2010. *The Grammar of Q: Q-particles, Wh-Movement, and Pied-Piping.* Oxford University Press.

———. 2012. “Pied-piping: Introducing two recent approaches.” *Language and Linguistics Compass* 6 (12): 816–832.

Canac Marquis, Rejean. 1994. “A/A-bar Chain Uniformity.” PhD diss., University of Massachusetts, Amherst.

Cattell, Ray. 1976. “Constraints on movement rules.” *Language,* 18–50.

Chomsky, Noam. 1981. *Lectures on government and binding.* Foris: Dordrecht.

———. 1986. *Barriers.* MIT Press (MA).

———. 2000. “Minimalist inquiries: the framework.” In *Step by Step: Essays in Minimalist Syntax in Honor of Howard Lasnik,* edited by R. Martin, D. Michaels, and J. Uriagereka, 89–155. Cambridge, MA: MIT Press.

———. 2004. “Beyond Explanatory Adequacy.” In *Structures and Beyond,* edited by Adriana Belletti. Oxford University Press.

———. 2019. “The UCLA lectures.”

Chow, Keng Ji. 2022. “A Novel Algorithm for Minimal Search.” *Snippets,* no. 42.

Cinque, Guglielmo. 1990. *Types of Ā-dependencies.* Cambridge, MA: MIT Press.

Gazdar, Gerald. 1981. "Unbounded dependencies and coordinate structure." *Linguistic Inquiry* 12 (2): 155–184.

Gould, Isaac. 2020. "Multiple Movement Dependencies and Parasitic Gaps." *Canadian Journal of Linguistics/Revue canadienne de linguistique* 65 (1): 110–121. Accessed October 16, 2024.

Grewendorf, Günther. 1988. *Aspekte der deutschen Syntax.* Tübingen: Narr.

———. 1989. *Ergativity in German.* Dordrecht: Foris.

Grewendorf, Günther, and Joachim Sabel. 1999. "Scrambling in German and Japanese: Adjunction versus multiple specifiers." *Natural Language & Linguistic Theory* 17 (1): 1–65.

Heck, Fabian. 2008. *On pied-piping: Wh-movement and beyond.* Berlin: Mouton de Gruyter.

Heck, Fabian, and Gereon Müller. 2007. "Extremely local optimization." In *Proceedings of WECOL34,* edited by Erin Bainbridge and Brian Agbayani, 170–182. California State University, Fresno.

Heim, Irene, and Angelika Kratzer. 1998. *Semantics in generative grammar.* Blackwell Publishing Ltd.

Heycock, Caroline. 1991. *Layers of predication: The non-lexical syntax of clauses.* University of Pennsylvania.

Hoji, Hajime. 1985. "Logical Form Constraints and Configurational Structures in Japanese." PhD diss., University of Washington.

Huang, C.-T. J. 1982. "Logical relations in Chinese and the theory of grammar." PhD diss., MIT.

Johnson, Kyle. 2003. "Towards an etiology of adjunct islands." *Nordlyd* 31 (1).

Kayne, Richard S. 1981. "ECP extensions." *Linguistic inquiry* 12 (1): 93–133.

Ke, Hezao. 2019. "The Syntax, Semantics and Processing of Agreement and Binding Grammatical Illusions." PhD diss.

Keine, Stefan. 2019. "Selective Opacity." *Linguistic Inquiry* 50 (1): 13–62.

Krivochen, Diego. 2022. "The Search for Minimal Search: A Graph-Theoretic View."

Landau, Idan. 2013. *Control in generative grammar: A research companion.* Cambridge University Press.

———. 2021. *A selectional theory of adjunct control.* MIT Press.

Larson, Richard K. 1988. *Light predicate raising.* Center for Cognitive Science, Massachusetts Institute of Technology …

Legate, Julie Anne. 2003. "Some Interface Properties of the Phase." *Linguistic Inquiry* 34(3):506–516.

Longenbaugh, Nicholas. 2017. "Composite A/Ā-movement: evidence from English tough movement." MIT. http://ling.auf.net/lingbuzz/003604..

———. 2019. "On Expletives and the Agreement-Movement Correlation." PhD diss., Massachusetts Institute of Technology.

Manzini, Rita. 1992. *Locality, a theory and some of its empirical consequences.* Cambridge, MA: MIT Press.

Marantz, Alec. 1997. "No Escape from Syntax: Don't Try Morphological Analysis in the Privacy of Your Own Lexicon." *University of Pennsylvania working papers in linguistics* 4 (2): 201–225.

Matushansky, Ora. 2006. "Head Movement in Linguistic Theory." *Linguistic Inquiry* 37, no. 1 (January): 69–109.

McFadden, Thomas, and Sandhya Sundaresan. 2018. "Reducing pro and PRO to a Single Source." *The Linguistic Review* 35 (3): 463–518.

———. 2019. "Deriving selective opacity via path-based locality." Handout presented at Syntax Mini-course, University of Cambridge. https://www.sndrsn.org/copy-of-agree-ment.

McGinnis, Martha. 1999. "A-Scrambling Exists!" *University of Pennsylvania Working Papers in Linguistics* 6 (1, Article 20).

Moltmann, Friederike. 1990. *Scrambling in German and the specificity effect. Ms.*

Müller, Gereon. 2010. "On Deriving CED Effects from the PIC." *Linguistic Inquiry* 41 (1): 35–82.

Müller, Gereon, and Wolfgang Sternefeld. 1994. "Scrambling as A-bar movement." In *Studies on scrambling,* edited by Norbert Corver and Henk van Riemsdijk, 331–385. Berlin: Bouton de Gruyter.

Newman, Elise. 2024. *When arguments merge.* The MIT Press.

Nissenbaum, Jonathan W. 2000. "Investigations of Covert Phrase Movement," Massachusetts Institute of Technology.

Pesetsky, David. 1987. “Wh-in-Situ: Movement and Unselective Binding.” *The representation of (in) definiteness* 98:98–129.

Pesetsky, David Michael. 1982. “Paths and categories.” PhD diss., Massachusetts Institute of Technology.

Postal, Paul Martin. 1994. “Contrasting Extraction Types.” *Journal of Linguistics* 30 (1): 159–186.

———. 1998. *Three investigations of extraction.* Vol. 29. mit Press.

Preminger, Omer. 2014. *Agreement and Its Failures.* MIT Press.

———. 2019. “What the PCC Tells Us about ‘abstract’ Agreement, Head Movement, and Locality.” *Glossa-an International Journal of Linguistics* 4 (13): 1–42.

Privoznov, Dmitry. 2021. “A Theory of Two Strong Islands.” PhD diss., Massachusetts Institute of Technology.

Richards, Norvin Waldemar. 1997. “What moves where when in which languages?” PhD diss., Massachusetts Institute of Technology.

Rizzi, Luigi. 1990. *Relativized Minimality.* The MIT Press.

Sauerland, Uli. 2003. “Intermediate adjunction with A-movement.” *Linguistic Inquiry* 34(2):308–313.

Sheehan, Michelle. 2013. “The resuscitation of CED.” In *Proceedings of NELS,* 40:135–150.

Stanton, Juliet. 2016. “Wholesale Late Merger in Ā-movement: Evidence from Preposition Stranding.” *Linguistic Inquiry* 47 (1): 89–126.

Stepanov, Arthur. 2007. "The end of CED? Minimalism and extraction domains." *Syntax* 10 (1): 80–126.

Stowell, Timothy Angus. 1981. "Origins of Phrase Structure." PhD diss., Massachusetts Institute of Technology.

Szabolcsi, Anna. 1997. "Strategies for Scope Taking." In *Ways of Scope Taking,* 109–154. Springer.

Szabolcsi, Anna, and Frans Zwarts. 1993. "Weak Islands and an Algebraic Semantics for Scope Taking." *Natural language semantics* 1 (3): 235–284.

Truswell, Robert. 2011. *Events, Phrases, and Questions.* OUP Oxford.

Uriagereka, Juan. 1999. "Multiple spell-out." *Current Studies in Linguistics Series* 32:251–282.

van Urk, Coppe. 2015. "A Uniform Syntax for Phrasal Movement: A Case Study of Dinka Bor." PhD diss., Massachusetts Institute of Technology.

van Urk, Coppe, and Norvin Richards. 2015. "Two components of long-distance extraction: Successive cyclicity in Dinka." *Linguistic Inquiry* 46 (1): 113–155.

Webelhuth, Gert. 1992. *Principles and parameters of syntactic saturation.* Oxford: Oxford University Press.

Williams, Edwin. 1992. "Adjunct control." In *Control and grammar,* edited by Richard Larson, Sabine Iatridou, Utpal Lahiri, and James Higginbotham, 297–322. Springer.

Yoshida, Mitsunobu. 2001. "Scrambling in German and Japanese from a minimalist point of view." *Linguistic Analysis* 30:93–116.

A. Weak Islands

The theory discussed so far is able to account for, among other things, the fact that *wh*-movement from certain adjuncts is in principle possible. Interestingly, however, not all types of *wh*-movement are possible: movement of an adjunct is considerably degraded when compared with movement of an argument. The answers in (46) are provided to force parses where both gaps originate within the control adjunct.

(46) **A weak island effect**

a. What did the flower open [in order to attract ___]

→ A: passing pollinators

b. *How did the flower open [in order to attract pollinators ___]

→ A: with a particular UV pattern

(46) is puzzling. The answer to this puzzle, in part, depends on whether or not we want our theory to be a general theory of weak islands. As the facts below suggest, weak islandhood is not straighforwardly connected to being a non-specifier: (47a) shows that complement clauses may be weak islands, while (47b) shows that the complement of a Neg head is comparably a weak island.

(47) **Complement weak islands**

a. *How do you regret that [John fixed the car ___]?

b. *How didn't [Mary arrive at the party ___]?

For the theory at hand, we can state the weak island property as something like the following, stated below. This description is vague enough to allow for either a syntactic (see D. Pesetsky 1987; Cinque 1990; Rizzi 1990) or a semantic (see Szabolcsi and Zwarts 1993; Szabolcsi 1997; Abrusán 2014, a.o) approach to weak island-hood.

(48) **Weak islandhood**

Operations making reference to a path defined by [~~•wh•~~] are barred from certain domains.

For us, given the generalization about weak islands above, the puzzle is why (46a) is acceptable. We sketch here a theory that will allow (46a), making use of certain assumptions about *wh*-movement first developed in §4, and having much in common with antecedent proposals about escape from weak islands originating in Rizzi (1990).

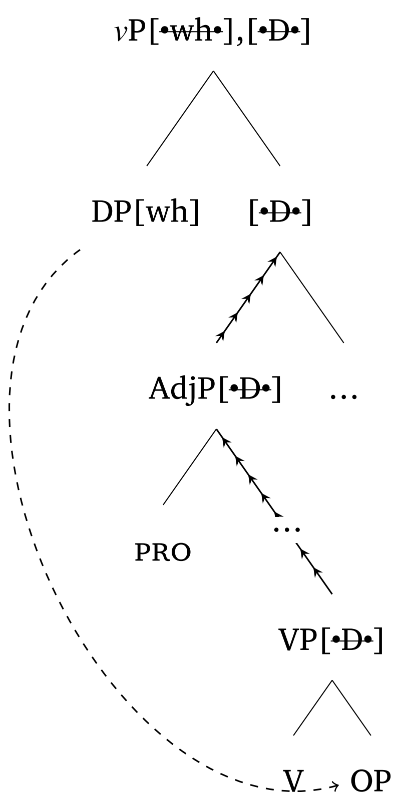
The core idea is that *wh*-movement out of weak islands is contingent on a feature [~~•D•~~] instead of a feature [~~•wh•~~]. Recalling our discussion of [wh] feature projection from §4, we saw that projection of a [~~•wh•~~] checked by the object was contingent on movement to an intermediate position motivated by checking of [•D•].

Wh-movement of objects, then, must involve two well-formed paths: one between the *wh*-phrase and its base position, defined by the feature [~~•D•~~], and another between the *wh*-phrase and the final landing site, defined by the feature [~~•wh•~~]. This, notably, contrasts with *wh*-movement of adjuncts, which do not occupy a position where they check [•D•].

Much work on *wh*-movement, at least in English, suggests that movement of *wh*-arguments may involve either a "true" movement dependency, or binding between the moved *wh*-element and something like a null pronominal (see D. Pesetsky (1987), Rizzi (1990), Postal (1994), and Stanton (2016) for discussion along these lines). Notably, the same is not true for movement of *wh*-adjuncts, where the binding strategy is not generally available. We suggest that the binding strategy — which only arguments may make use of — involves Search through a domain bearing [~~•D•~~], rather than [~~•wh•~~], and that it is this distinction which allows *wh*-movement of arguments to avoid being blocked by (48).

Consider the structure below, where AdjP is taken to be a weak island as in (46), and thus subject to (48). Following Rizzi (1990) and Postal (1994), a null element — represented here by OP — may in principle be merged in a position where it checks [•D•], provided it is subsequently bound by a *wh*-phrase of some sort. Binding requires a path of [•D•] features between the binder and bindee, similar to the binding of PRO and null operators discussed earlier in this paper. As we see below, the *wh*-phrase may in principle be generated in Spec *v*P of the matrix clause, so long as the adjunct containing OP appears in a position from which it may project its features. The *wh*-phrase may bind OP from this position, via the path of [•D•] features between the two. The *wh*-phrase is also Merged in a position where it checks a [•wh•] feature in the matrix clause, creating a path of [~~•wh•~~] to matrix Spec CP that does not traverse a weak island.

(49)



In contrast, such a derivation is not available for adjunct *wh*-phrases such as *how*, as in the case of (46b). Such an adjunct could be merged in Spec *v*P of the matrix clause, but it would be unable to bind a comparable OP in the adjunct clause. Such an adjunct could also be merged in Spec *v*P of the adjunct clause, but subsequent movement of the adjunct would run afoul of (48).