Deconstructing merge and move to make room for adjunction

Tim Hunter

Abstract. This paper explores the syntactic and semantic properties of movement and adjunction, and proposes a new analysis of these two phenomena according to which they are closely related to each other. The basic pieces of grammatical machinery that carry out movement, also carry out adjunction. Building on independently appealing ideas concerning neo-Davidsonian semantic composition and "move as re-merge", the proposed framework naturally accounts for (i) adjuncts' puzzling status "either inside or outside" a maximal projection, (ii) adjuncts' ability to escape reconstruction, and (iii) the prohibition on extraction from adjuncts.

1. Introduction

My aim in this paper is to explore the syntactic and semantic properties of movement and adjunction, and in particular to propose a new analysis of these two phenomena according to which they are closely related to each other. In a precise sense, the basic pieces of grammatical machinery that give rise to movement, also give rise to adjunction. In the system I propose, there is no atomic movement operation and no atomic adjunction operation; the terms "movement" and "adjunction" serve only as convenient labels for certain combinations of other, primitive operations. As a result the system makes non-trivial predictions about how movement and adjunction phenomena should *interact*, since — in contrast to systems with distinct atomic operations for movement and adjunction — we do not have the freedom to stipulate arbitrary properties of movement while leaving the properties of adjunction unchanged, or vice-versa.

The empirical focus is on three properties of adjuncts: their ability to be either included in or excluded from a maximal projection targeted for syntactic manipulation, their ability

This paper is a condensed and less technical presentation of the central results of Hunter 2010 and Hunter 2011b. Foremost among the many who have contributed helpful comments are Bob Frank, Norbert Hornstein, Paul Pietroski, Amy Weinberg and Alexander Williams; thanks also to the editors and anonymous reviewers.

to attach "counter-cyclically", and their islandhood. While each of these properties is relatively well-known, my central aim here is to propose an analysis in which these three apparently independent properties all follow from a single underlying distinction between adjuncts and non-adjuncts. The particular underlying derivational distinction on which the explanation hinges emerges as a logical possibility when we adopt a system that fleshes out the intuition that movement might be usefully thought of as "re-merge", and is supported by a restrictive, independently-motivated theory concerning the composition of neo-Davidsonian logical forms.

The rest of this paper is organised as follows. In §2 I outline the descriptive generalisations concerning adjunction that I aim to account for. I give brief sketch of the proposal's central ideas and intuitions in §3, before fleshing out the details in §4 and §5. I then show in §6 how the resulting system accounts naturally for the generalisations discussed in §2.

2. Adjunct puzzles

2.1 Adjuncts and constituent structure

I begin with the following facts because they illustrate what is arguably considered the characteristic properties of adjuncts; they are sometimes provided as *diagnostics* for adjunct-hood in syntax textbooks. But as Chametzky (2000:12–13) and Hornstein & Nunes (2008) note, the question of why these characteristics correlate with the other properties we now associate with adjunct-hood remains open.

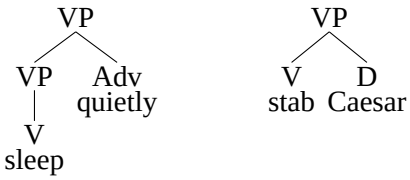
Consider first the contrast between (1) and (2). What is it about the two verb phrases, 'sleep quietly' and 'stab Caesar', that permits 'quietly' to be either included in or excluded from the constituent targeted for fronting in (1), but does not permit 'Caesar' the same flexibility in (2)?

(1) a. Brutus [$_{VP}$ slept quietly].
b. Sleep quietly, Brutus did.
c. Sleep, Brutus did quietly.

(2) a. Brutus [$_{VP}$ stabbed Caesar].
b. Stab Caesar, Brutus did.
c. * Stab, Brutus did Caesar.

This contrast has often been encoded using structures like those in (3), where the relevant fronting operation can only apply to a node labelled VP. But we may still wonder *why* these phrases, each composed of a head verb and one other word, differ in the way described by (3).

(3)



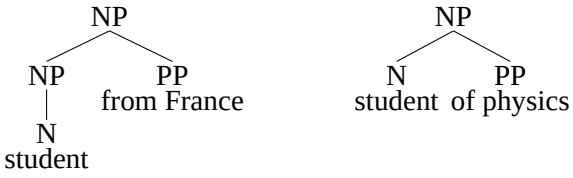
As is well-known, this contrast is a matter of general *constituent structure*, not a fact specific to the particular fronting operation in (1) and (2). The same pattern of judgements can be observed, for example, the context of 'it'-clefts, VP-ellipsis, or 'do so'-substitution.

The contrast under consideration is also not specific to the structure of verb phrases, as we can see by probing the structure of nominal phrases in (4). Along with a requirement that 'one'-substitution targets nodes labelled NP, this pattern can be encoded using the trees in (5).

(4) a. I taught this student from France and you taught that one (from Germany).

b. I taught this student of physics and you taught the one (*of chemistry).

(5)



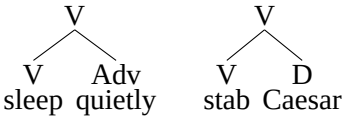
Constituents that are restricted in the way 'Caesar' and 'of physics' are here I will describe as "arguments", to be distinguished from adjuncts which pattern like 'quietly' and 'from France'. We would like to know why non-head parts of phrases come in these two distinct flavours.

Note that I have left aside the distinction between optional and obligatory parts of phrases. Neither of the PPs that attach to 'student' in (4) is obligatory, and yet the distinction which was correlated (and thus, it seems, *confounded*) with the optional/obligatory distinction in the earlier verb phrase examples still appears.

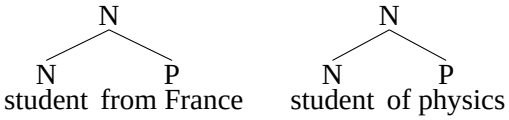
As Hornstein & Nunes (2008) point out, the puzzling nature of these facts comes sharply into focus when we consider the recent trend towards a *relational* view of X′ levels (Speas, 1986; Chomsky, 1995). The "Bare Phrase Structure" (BPS) approach does not permit non-branching nodes, and the label chosen for a newly-formed constituent records only the two-way choice of which daughter contributes the head. With this spare machinery we cannot make the distinctions encoded in (3) and (5): we have no choice but the structures in (6) and (7).[1]

[1] I assume that 'sleep quietly' and 'stab Caesar' are projections of 'sleep' and 'stab' respectively; in contrast to the "functional specifier" approach to adjunction (Cinque, 1999), according to which 'quietly' would not be part

(6)



(7)



These representations give us no way to distinguish between the two kinds of structures: the facts indicate that given two constituents α and β, there are two distinct ways in which α and β can combine with α as their head, and BPS only has room for one.[2] But whether one generally subscribes to BPS or not, the labelling distinction in (3) and (5) remains a stipulation to be eliminated if possible. Considering BPS is just one way to bring this clearly to our attention.

2.2 Counter-cyclic attachment

The distinction discussed in the previous subsection correlates with other apparently unrelated contrasts. One is that adjuncts seem to be able to avoid reconstruction in a way that arguments are not (Freidin, 1986; Lebeaux, 1988; Landau, 2007), as illustrated in (8) and (9). The bracketed phrases are arguments in (8a) and (9a), but adjuncts in (8b) and (9b).

(8) a. * Which claim [that Mary$_i$ was a thief] was she$_i$ willing to discuss?

b. Which claim [that Mary$_i$ had made] was she$_i$ willing to discuss?

(9) a. * Paint [pictures of Mary$_i$], she$_i$ knows I wouldn't

b. Paint [in Mary$_i$'s house], she$_i$ knows I wouldn't

The degraded status of (8a) and (9a) is straightforwardly accounted for via Condition C, since 'Mary', in its base position, is c-commanded (and thus bound) by 'she'. The question then arises of why an analogous Condition C violation does not appear in (8b) and (9b).

---

of the projection of 'sleep', but rather a specifier of a distinct functional projection dominating that of the verb. On this view one must supplement the basic headedness of phrase structure with some notion of an "extended projection" for 'sleep' and 'stab' to be identified as the heads of these two structures.

[2]The account of adjuncts in the original proposal of BPS is problematic: Chomsky (1995) suggests a system where the node at the root of 'sleep quietly' would have the ordered pair ⟨V,V⟩ as its label, identifying the V 'sleep' as the head without rendering 'sleep' non-maximal by sharing a label with it. But with labels such as ⟨V,V⟩, one could re-formulate the X-bar system of the GB era by using ⟨X,X⟩ in place of X′, ⟨X,X,X⟩ in place of X″, and so on. Whatever the force of the Inclusiveness Condition that motivates BPS, it is generally thought to rule out such "intrinsic bar levels", in which case it should also rule out Chomsky's account of adjunction.

Lebeaux's (1988) influential analysis of the pattern in (8) (easily adaptable to (9)) supposes that the adjunct 'that Mary had made' is added to the structure *after* the fronting operation has taken place — "counter-cyclically" — so there is no point in the derivation where 'Mary' is c-commanded by 'she', and no Condition C violation arises. This requires, of course, an independent reason to believe that the argument 'that Mary was a thief' (and similarly 'pictures of Mary' in (9)) cannot likewise be added counter-cyclically. The suggestion was that this is not possible for arguments because, unlike adjuncts, they must be present at d-structure (and thus, throughout the derivation) in order to satisfy thematic/subcategorisation requirements.[3]

My proposal will be based along roughly these lines: the notion of assigning thematic roles (or something like them) plays a central part in my analysis. For this idea to be meaningfully cashed out, however, we require some independent notion of what we mean by "thematic role assignment". For this I will draw on neo-Davidsonian semantics (Parsons, 1990; Schein, 1993; Pietroski, 2005). Furthermore, even given an independently justified conception of thematic role assignment, we might still hope for a better understanding of why *receiving* a thematic role correlates with more *constrained* syntactic behaviour (e.g. inability to attach counter-cyclically, rendering sisters inaccessible) and *not receiving* one correlates with more *flexible* syntactic behaviour, rather than the other way around. These are the questions that I aim to shed light on.

2.3 Adjunct islands and freezing effects

Another property of adjuncts, not obviously related to those discussed so far, is that they are generally islands for extraction (Cattell, 1976; Huang, 1982). A representative contrast between licit extraction from an argument clause and illicit extraction from an adjunct clause is illustrated in (10).[4]

(10) a. Who do you think [that John saw ____]?

b. * Who do you worry [because John saw ____]?

[3] Adjuncts' ability to attach counter-cyclically cannot be tied to their "optionality" in a general distributional sense, because the argument clause of 'claim' is equally optional ('Which claim was she willing to discuss?').

[4] A number of apparent exceptions, where extraction from adjuncts yields a surprisingly acceptable result, have been identified and discussed (e.g. Truswell, 2007). I will put this ongoing debate aside and assume that the basic generalisation is correct.

Proposals aiming to derive the prohibition on extraction from adjuncts have been made by Uriagereka (1999) and Stepanov (2001). Uriagereka's account is independently motivated by assumptions about constraints on the PF interface adopted from Kayne (1994), but makes no connection between the fact that adjuncts are islands and the other generalisations noted in the previous subsections. Stepanov links the fact that adjuncts are islands to Lebeaux's (1988) idea of counter-cyclic attachment: he suggests that adjuncts not only *can* attach counter-cyclically, late in the derivation, but that they *must* attach counter-cyclically, and in particular they must attach after all non-adjunction operations have occurred. Therefore it is impossible to derive (10b), for example, because the wh-movement of 'who', which is not an adjunction operation, becomes impossible once 'because John saw (who)' is adjoined. Stepanov's attempt to unify the island behaviour of adjuncts with the contrast discussed above in §2.2 is appealing, but nonetheless requires a further stipulation beyond Lebeaux's proposal: while subcategorisation properties arguably justify the assumption that adjuncts *don't need* to be present "early" (at d-structure, or equivalent), there is no obvious reason why they *can't* be.

In the system that I will propose for the contrasts in the previous two subsections, it becomes possible to impose a single, more general constraint that derives both the islandhood of adjuncts and a distinct, independently observed generalisation: the prohibition on extraction from moved constituents (Wexler & Culicover, 1981; Corver, 2005). More specifically, it will emerge from my proposal that adjoined constituents and moved constituents have, in a precise sense, the same status at a certain point in the derivation. This makes it possible to impose a single constraint on the system which will rule out both the "adjunct island" violation in (10), and the "freezing effect" or "derived island" violations in (11) and (12).

(11) a. Who did you send [a big heavy picture of ____] to London?

b. * Who did you send to London [a big heavy picture of ____]?

(12) a. Who did you buy [a picture of ____]?

b. * Who was [a picture of ____] bought (by you)?

These contrasts indicate that a phrase which otherwise permits extraction, as illustrated by the acceptable examples, becomes an island when it is moved away from its base position (either via heavy NP shift, or via passivisation/A-movement).[5] In the framework I will propose, these

effects will follow from the very same constraint that rules out (10b).

Proposals have been made which do invoke movement out of a moved constituent (sometimes called "smuggling" (Collins, 2005)). These are clearly inconsistent with the constraint I assume, as Collins (2005:97) mentions, and I do not have any suggestions to offer about how this incompatibility might be best resolved.

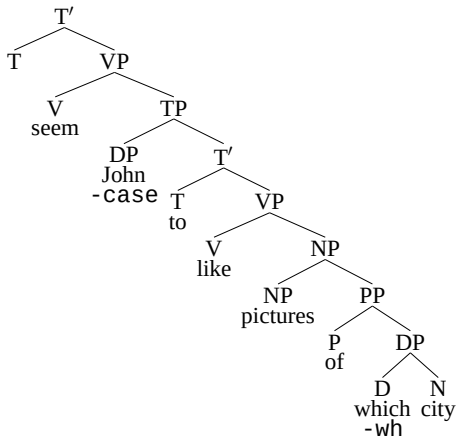
3. An overview of the proposal

This section provides a basic sketch of my proposal regarding adjunction and its relationship to merge and move operations.[6] This outline will be simplified in certain respects in order to clearly convey the relevant intuitions. Further details, of both the proposed technology (§4 and §5) and the explanations it provides for the empirical puzzles presented above (§6), will follow.

To begin, consider the point in the derivation of (13) conventionally illustrated as in (14).

(13) Which city does John seem to like pictures of?

(14)



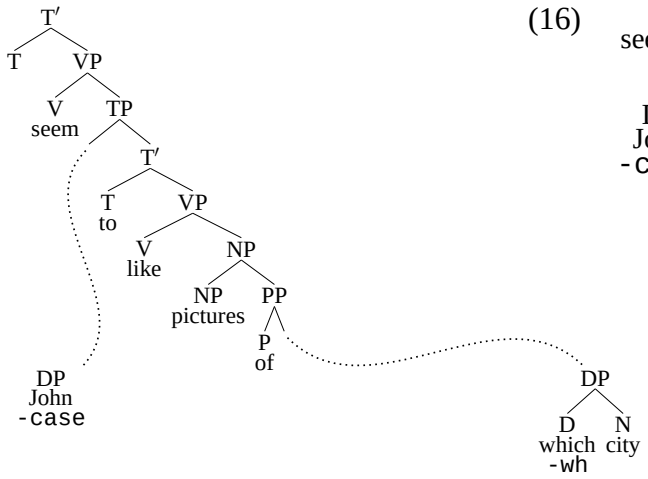
Note that two pieces of this tree, namely 'which city' and 'John', must move from their current positions in any successful continuation of this derivation. I assume that this information is present at this point in the derivation, in the form of certain unchecked features or equivalent. It is represented in (14) by the `-case` and `-wh` features on 'John' and 'which' respectively. (I also assume, as usual, that the checking of the `-wh` feature will require movement of the entire

[5] I say nothing about whether subject islands should be entirely reduced to freezing effects (Stepanov, 2001).

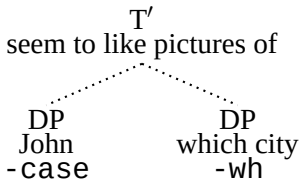
[6] What I present as an implementation of overt movement is intended, to the degree possible, as a mechanism that might equally underly other varieties of long-distance dependency that are often analysed in terms of, say, covert movement or agree; cf. Hornstein (2009:ch. 6) and also Hornstein (2001) on binding and control.

DP that is headed by its bearer, 'which'.) To put the same observation differently, we can say that 'which city' and 'John' do not occupy the positions where they appear in the tree in (14) in quite the same sense that the other pieces (e.g. 'seem', 'pictures') do. With this in mind, we can consider a notational variant that makes this distinction more visually salient, shown in (15).

(15)



(16)



This sort of notation provides a novel perspective on minimalist derivations which, I will argue, opens our eyes to certain ways that adjunction might be accommodated. The crucial intuition to appreciate is that (15) is, first and foremost, a formal object comprising *three* pieces: [$_{DP}$ John], [$_{DP}$ which city] and [$_{T'}$ seem to like pictures of] (Stabler, 1997, 2001; Michaelis, 2001). In fact, in the system I will propose, these three pieces and the relationships among them are, in a sense, all that matters. Accordingly, the system will work instead with a compressed version of (15), as shown in (16), where each of the three pieces just identified has been assigned PF and LF interpretations (although I continue to omit semantic values). Note that the root node of this tree has no internal structure to which syntax is sensitive. Such nodes have much in common with lexical items, being the pairing of a phonological string (as shown here) and a semantic interpretation (not shown). The shift from (14) to (16) is a claim about the formal nature of syntactic objects, not simply a notational abbreviation. Syntactic operations applying to (16) will not be able to "see" the internal relations among the words 'seem', 'like', etc. any more than syntactic operations applying to a conventional object like (14) are usually able to "see" the internal relations among the phonemes that make up a word. When I say "PF (or LF) interpretation", I refer to the process that produces such derived but unstructured objects.

By moving from (14)/(15) to (16) we have discarded certain information about hierarchical

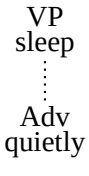
structure. Specifically, we have discarded the hierarchical relations that were encoded by solid lines in (15). Of course, this information was there in the conventional representations for a reason: it is significant that 'pictures' is (the head of) the *object* of the verb 'like' rather than the subject, for example. But when we say that this is significant, and that it is important that (14)/(15) encode it, what we mean — to a good first approximation — is that it is important to ensure that the eventual PF representation linearises 'pictures' to the right of 'likes' rather than to the left, and that the eventual LF representation involves pictures as the theme/patient of liking rather than as the agent/experiencer. The PF and LF representations assigned to the T′ piece in (16) simply encode such information directly, as facts about linear order and semantic interpretation, although only linearisation, not semantic interpretation, is shown. (Of course, it is also significant that the various lexical items have had their selectional requirements fulfilled, but this is encoded, in effect, by the T′ annotation in (16).[7]) This basic idea has some parallels with "direct compositionality" (Barker & Jacobson, 2007). But to repeat, my main aim for now is not to demonstrate all the details, only to foreshadow the possibilities that would arise from reconstruing (14)/(15) as shown in (16) *if* such a move could be shown to be feasible.

The new perspective that emerges from considering the manipulation of structures like (16) is helpful in two respects: it makes available (i) a concrete sense in which "movement" can be analysed as re-merge, and (ii) a certain hypothesis about the nature of adjunction configurations and how they differ from complement and specifier configurations.

First: emphasising the sense in which 'John' and 'which city' are *dis*connected from the main clausal structure in (15) and (16) will play a part in formulating a system where merge and move are genuinely reduced to a single operation (Epstein et al., 1998; Chomsky, 2004). What usually distinguishes move from merge, intuitively, is that one of the constituents it affects is a subconstituent of the other. To the extent that we can adopt a mindset (and notation) that deemphasises the sense in which 'John' and 'which city' are parts of the T′ constituent in (15) — and brings out the sense in which 'John' and 'which city' are *external* to this T′ constituent — doing so will facilitate the unification of merge and move. Details will come in §4.

[7] If the projecting matrix T head's requirement for a VP complement were not fulfilled, this T′ annotation at the root of (16) would have been a mere T instead; and had the requirement that 'seem' have a TP complement not been fulfilled, there would have been no VP with which to satisfy the previous requirement, etc.

Second: as regards adjunction, I will suggest, roughly, that the "loose" connections among these sorts of pieces, indicated by dotted lines in (15) and (16), are essentially the kind of connection that an adjunct has to its host. This is to be contrasted with the "tighter" connections within these pieces, indicated by solid lines in (15), which characterise complements and specifiers. This idea that adjuncts bear, in some sense, a looser or less intricate relation to their hosts than arguments do has been put forward by Chametzky (1996) and Hornstein & Nunes (2008). Given a system that deals with structures like (16), I will suggest that at a certain point in the derivation of 'John slept quietly' we will have a structure like (17). For comparison, at the equivalent points in the derivations of 'John saw Mary' and 'Who did John see', we will have — by the same logic that relates (15) to (16) — the two structures in (18) and (19).

(17)


(18)


(19)


Details aside, the comparison of (17), (18) and (19) is intended just to foreshadow the basic idea that the two "non-canonical" variants of (what is generally thought to be) the more canonical verb-object construction in (18) — a movement variant and an adjunction variant — will both make use, in slightly different ways, of the same basic grammatical machinery.

The claim will not be that adjunction is parasitic on movement, which might mean that 'quietly' is "moving out of" the VP in (17), in something like the way that 'who' is "moving out of" the VP in (19); nor that movement is parasitic on adjunction, which might mean that 'who' is adjoined to the VP, in some sense, in (19). Rather, I will suggest a new set of basic operations which build structures like these, and that certain patterns of these operations yield what are known as adjunction phenomena, and that other patterns yield what are known as movement phenomena. The conception of movement that these basic operations bring is very much consistent with — arguably even implied by — the existing suggestion that movement be thought of as re-merge. So one way to state the central claim is that we find a natural home for adjunction phenomena if we adopt this conception of basic operations and structures, which is already implicit in many existing discussions of the relationship between merge and move.

Broadly speaking, the picture will be as follows. There will be three basic operations: insert, merge and spellout. Intuitively, the insert operation introduces an element into the derivational workspace without establishing any grammatical dependencies, following Stabler (2006). A constituent typically thought of as occupying just a single position (say, 'Mary' in 'John saw Mary', under simple assumptions) will undergo insert, then merge, then spellout. A constituent typically thought of as undergoing one movement step (e.g. 'John' in 'John was arrested') will undergo insert, then merge, then merge, then spellout. A constituent typically thought of as adjoined (e.g. 'quietly' in 'John slept quietly') will undergo only insert and spellout.

The fact that adjuncts do not undergo merge does not imply that they are not composed, with respect to both LF and PF interpretation, with their hosts. In this sense my use of the term "merge" differs from that of Chomsky (1995), who uses this term as a kind of catch-all for any operation that composes two syntactic objects: when Chomsky (2004:117–119) discusses the possibility of two distinct operations, one for adjunction structures and one for non-adjunction structures, they are both treated roughly as sub-cases of "Merge", named "pair Merge" and "set Merge". My choice of usage is guided by the fact that the operation I call "merge" implements the kind of syntactic composition that the usual minimalist assumptions about (bare) phrase structure *do* handle relatively well, namely that which produces complements and specifiers (recall §2.1). The challenge, then, is to find out what other operation(s) implement adjunction.

One might worry, given the claim that adjuncts do not undergo merge, that I will be forced to add, by stipulation, a distinct operation to combine adjuncts with their hosts. It is true that I will be proposing mechanisms, distinct from (what I call) merge, by which adjuncts combine. But to repeat from above, I will argue that these distinct mechanisms are, in effect, independently required in order to implement the common "move as re-merge" intuition.

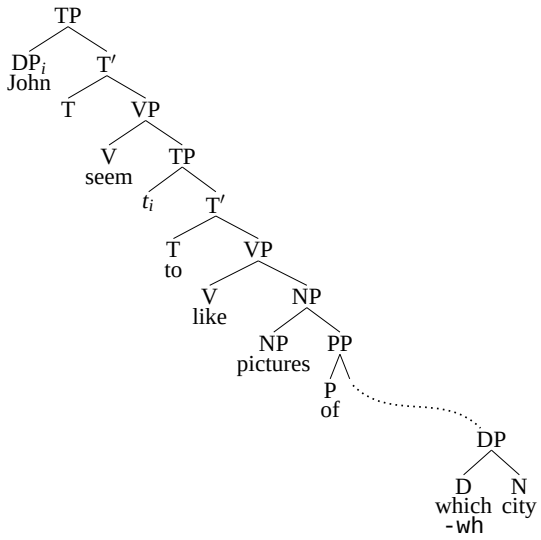
The details of the proposal outlined here will be presented in the next two sections. First, §4 gives a proper presentation of the proposed reconceptualisation of merge and move operations, manipulating representations defined entirely by "dotted line structure" like (16), (17), (18), and (19). Then in §5 I will return to "solid line structure" — the structure internal to pieces that are linked by dotted lines. For the purposes of the overview in this section, and also for the purposes of §4, it is convenient to assume that this sort of structure can be completely ignored,

but this is a slight simplification: in a precise sense that will be made clear, a *small amount* of this "solid line structure" (though much less than is visible in (15)) will be reintroduced. With the details of both long-distance, dotted-line structure and local, solid-line structure laid out, a natural treatment of adjunction becomes possible, which will be presented in §6.

4. A new conception of merge and move

Consider now the next step of the derivation of (13). The embedded subject 'John' will move to the matrix subject position. Using the modified notation from (15), the resulting syntactic object has the structure shown in (20).

(20)

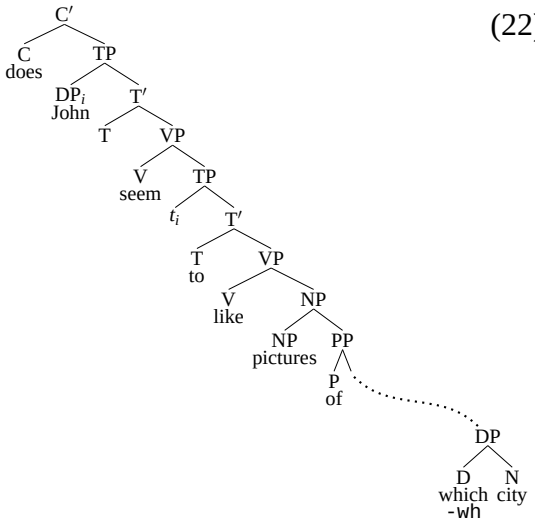


The question of exactly what we understand to be in the embedded subject position — a trace, a copy, nothing, or whatever — is not important for our purposes; I will use trace-like notation to minimise clutter. What is important is that the new position of 'John', the matrix subject position, is its final position. It will not move anywhere else in the future of this derivation. In this respect its status is now analogous to that of 'seem' and 'to' and 'like', which are already in their final positions; it is no longer "disconnected" in the way that 'which city' is. As a result, in (20) we now have two pieces of structure, in the way that in (15) we had three: just [$_{\text{DP}}$ which city] and [$_{\text{TP}}$ John seem to like pictures of].

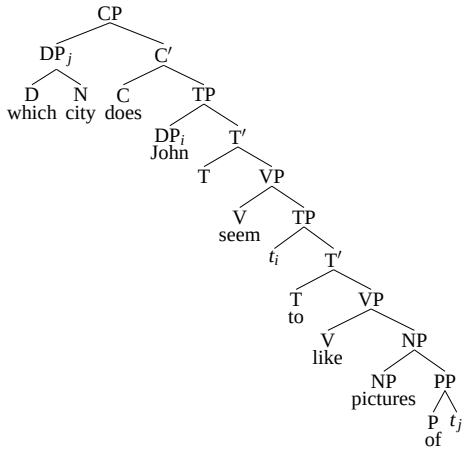
After adding the C head — for now I leave aside the details of such merge operations, and keep attention focused on movement — we will have the (two-piece) structure in (21). Finally, 'which city' will move to the specifier position of CP, producing the final (one-piece) structure in (22). Like 'John' in (20), 'which city' has now settled into its final position. The progression

from (21) to (22) is, of course, analogous to the progression from (15) to (20).

(21)



(22)



Now consider again the relationship between the two representations of the one derivational stage that we began with, namely (14) and (15). It is useful to ask precisely what purpose the structure assigned to the T′ constituent in the conventional (14) serves: why would it *not* be sufficient to take the syntactic object derived at that point of the derivation to be simply a string, without internal structure, marked with a label that determines its combinatory potential?

(23) T′
seem John to like pictures of which city

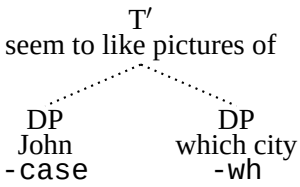
The reason, of course — at least one very salient reason, and the one that will be most useful to consider first — is that pieces of this string need to be rearranged at future points in the derivation, and the structure given in (14) encodes important information about which particular pieces can and/or must be rearranged. This information is not present in (23).

Which parts of the structure in (14), however, are really doing the crucial work? Some of the information encoded there is crucial, but much of it is not. What is crucial, in the relevant sense, is the structure encoded by the dotted lines in (15). It is important that the three pieces [$_{DP}$ John], [$_{DP}$ which city] and [$_{T'}$ seem to like pictures of] are identifiable as such. But note that the internal structure of these pieces is *not* relevant to any of the movement operations that complete the derivation. The three strings ‘John’, ‘which city’ and ‘seem to like pictures of’ are moved around *relative to each other*, but the internal structure of these three pieces remains unaffected. We know this because the only features that can trigger future movement operations are the `-case` feature on ‘John’ and the `-wh` feature on ‘which’. Importantly, this is not

an empirical claim about the possibility or impossibility of certain movement configurations: rather, the point is just that given a conventional structure like (14), it follows as a matter of logic, from the positions of the unchecked movement-triggering features (plus the assumption that all movement checks features), which aspects of the structure are relevant in this sense.

Following this line of thought, it is safe to ignore the relationships indicated by solid lines in (15), and consider only those indicated by dotted lines. We can use (16) to represent this.

(16)



This can be understood as follows: we have a T′ constituent with PF interpretation ‘seem to like pictures of’, out of which will move (i) a DP constituent with PF interpretation ‘John’, in order to check a Case feature, and (ii) a DP constituent with PF interpretation ‘which city’, in order to check a wh feature. Another way to put the basic observation is that while it is not sufficient to take the structure derived at this point to consist of a single string as in (23), it is also not *necessary* to take it to consist of eight distinct strings as in (14). Three is enough.

This perspective essentially originates from Michaelis (2001), following Stabler (1997). It has facilitated insights into the relationships between minimalist syntax and other grammatical frameworks, for example tree-adjoining grammars (Joshi, 1987) and combinatory categorial grammars (Steedman, 2000); see Stabler (2011) for review. In this paper I am suggesting that the same construal of merge and move operations that provided the clearest picture of the mathematical properties of minimalist syntax can also pave the way for theoretical developments of the sort that mainstream syntacticians regularly deal with.

Note that (16) is an unordered tree: the left-to-right order in which nodes appear on the page is not significant. The geometry of the tree does not distinguish between the status of ‘John’ and ‘which city’, although these are, of course, distinguished by their different feature-checking requirements. Since we have discarded the internal structure of the T′ constituent, this means that there is no remaining *syntactic* record of where these two constituents are “coming from”, i.e. that ‘John’ is coming from the embedded subject position and that ‘which city’ is coming from the embedded object position. There are two concerns that this might raise: first, that we

might lose information that is relevant for semantic interpretation, and second, that we might lose information that is relevant for computing compliance with locality or island constraints. These are valid and relevant concerns, and I return to them below in §4.4.

4.1 Recasting merge and move

Adopting the new notation in (16), how will the continuation of the derivation look? The next step is the raising of 'John'. Recall from (20) that this produces a structure that has two relevant pieces, 'which city' and 'John seem to like pictures of'. The number of relevant pieces is reduced because 'John' has settled into its final position (since it has checked all its features), so no future operations will separate it from 'seem to like pictures of'. In the new notation we therefore have the simple structure in (24), which is straightforwardly related to (20). The effect of a movement step, described in terms of these new structures, is therefore roughly to take a non-root node (here, the node corresponding to 'John') and incorporate it into the root node (here, 'seem to like pictures of'). For the moment I leave open the question of how the relative order of the two strings being combined here is determined; questions like this will play a crucial role below in §5. Similarly, the effect of the wh-movement of 'which city' is to transform (25) into (26). These two structures correspond to (21) and (22) in the expected way.

(24)
TP
John seem to like pictures of
⋮
DP
which city
-wh

(25)
C′
does John seem to like pictures of
⋮
DP
which city
-wh

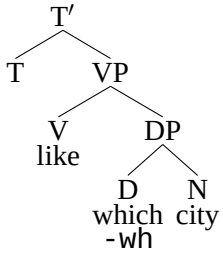
(26)
CP
which city does John seem to like pictures of

With this reconception of movement in mind, let us turn to (external) merge operations. We will see that the proposed conception of movement makes it possible to unify merge and move (or external merge and internal merge) to a degree that would otherwise be difficult. To illustrate this, consider (27). For ease of exposition I will continue to ignore VP-internal subject positions; therefore 'John' is placed directly into the specifier position of TP, and does not move from this position. Just *before* this occurs, a T′ constituent has been constructed out of which a wh-phrase will move. This stage of the derivation is illustrated in (29); for convenience, the

stage under discussion is shown in (28) with conventional notation.

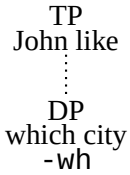
(27) Which city does John like?

(28)



(29)

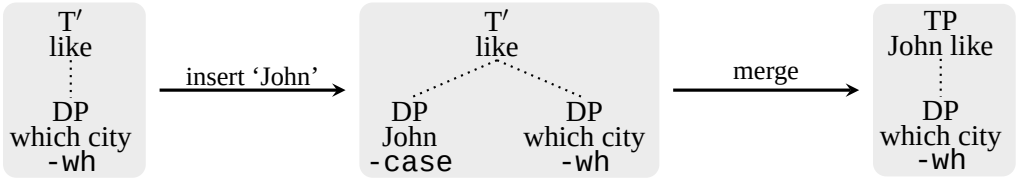
(30)



Now consider the structure that we need to end up with immediately *after* placing 'John' in the subject position. What we would expect to have at this point is a TP constituent, with PF interpretation 'John like', out of which a wh-phrase will move. This is shown in (30). So whatever mechanisms we end up proposing in order to implement this first merging of 'John', their effect should be to transform (29) into (30). Roughly speaking, the desired effect is to add to the piece at the root of the structure (here, augmenting it from a T′ 'like' to a TP 'John like'), and leave other pieces (here 'which city') unchanged.

Now, the crucial observation that facilitates the "move as re-merge" perspective is as follows. We *already have available* a mechanism that adds to the piece at the root of a structure: 'John' was added to the root of the structure in the transition from (16) to (24), as was 'which city' in the transition from (25) to (26). We would like the same operation that carried out these derivational steps seen earlier — whether we call this operation "move" or "merge" — to be at least part of what does the work of transforming (29) into (30). This operation takes a daughter piece and incorporates it into the root piece. If this is going to be part of what transforms (29) into (30), then we need a way to put 'John' into the structure as a daughter piece.

A new operation, called "insert" (following Stabler, 2006), does exactly this: adds a new element to a structure as a daughter to the root. The effect of inserting 'John' into the structure in (29) is shown in (31). This is then followed by an application of the existing operation that incorporates a daughter piece into the root. The result of this two-step process is to transform (29) into (30), as desired.

(31) 

Note that the operation “merge” here is the very same operation that put ‘John’ into the subject position to transform (16) into (24) (and put ‘which city’ into its left-edge position to transform (25) into (26)). The current simplification of ignoring VP-internal subject positions helps to bring out this parallel: in each case ‘John’ is going into a specifier position of TP, whether this is its base position or a derived position. (For ease of exposition I suppose that the “external merge” of ‘John’ here is licensed by a `-case` feature. Such details will be clarified shortly.) So external merge and internal merge are related, in that one has the other as a part; but rather than analysing internal merge as (say) copy plus external merge, we analyse external merge as insert plus internal merge. See Hunter (2011a) for discussion.

That we have been led to posit this insert operation as the way to introduce new elements into the derivation is significant. Recall from (17) that the eventual claim about adjuncts will be, essentially, that the loose relation I have been indicating with dotted lines is the *only* connection they bear to their hosts. The insert operation that we have just introduced is what places them there. The question of how something that is “only inserted” can be linearised and semantically interpreted as a part of the sentence will be answered in §6. But it should be clear that the insert operation introduced here will have a natural role to play in fleshing out the proposal that adjuncts only ever occupy these “loosely attached” positions in syntactic structures.

4.2 Notational book-keeping

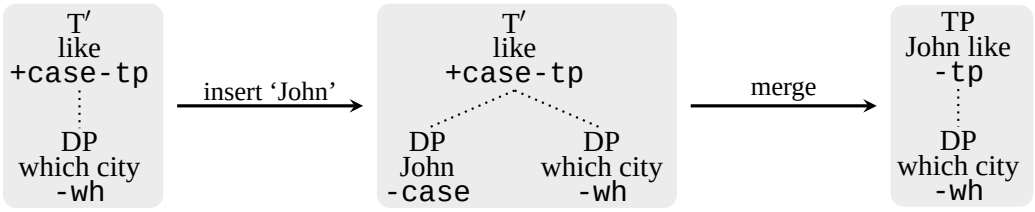
Annotations such as `-wh` and `-case` were introduced to make clear the distinction between constituents that have reached their final positions and constituents that have not. Accordingly, all and only non-root nodes in the new notation I have adopted have such annotations.

Of course, there are also facts about what the pieces that appear at the root nodes can and cannot combine with. The T′ node at the root of the first two trees in (31) is something that can combine with a `-case`-annotated thing, but the TP node at the root of the last tree in (31) is not. It would be meaningless to say that ‘John’ needs to satisfy a certain requirement, indicated by writing `-case`, without specifying where and how this can happen. And if this T′ node

were the kind of thing that discharged `-wh` requirements rather than `-case` requirements, the merge step in (31) would have manipulated 'which city', not 'John'.

Up to this point I have taken these distinctions to be understood implicitly from labels such as T′ and TP, but it will be useful to record them more explicitly. To encode the sense in which the T′ node in (31) can act as the relevant "match" for a bearer of `-case`, I will annotate it with `+case`. Thus (31) will be written as in (31′).

(31′)

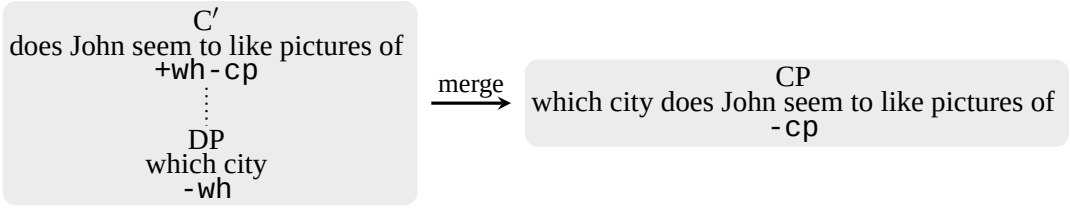


I will use the term "feature" for all these annotations, and when they are satisfied I will say that they are checked or deleted, but readers can substitute their own terminology if desired. Likewise, the names I use for these features have no particular significance: if `-case` and `+case` were replaced with, say, `-epp` and `+epp`, or with `-xyz` and `+xyz`, nothing would change. All that matters is which features "match" which other features.

Notice also the `-tp` feature on the root node of each tree in (31′). I will adopt the simple assumption that features drive not only "movement" steps, but also "external merge" steps (Stabler, 1997; Hornstein, 2001; Adger, 2003; Müller, 2010). Hence selectional relationships, such as selection of a TP complement by a C head, will be implemented by feature-checking: just as the appropriate position(s) for 'which city' to go into are dictated by its `-wh` feature (in combination with appropriately-placed `+wh` features), the appropriate position(s) for 'John like' are dictated by its `-tp` feature (in combination with appropriately-placed `+tp` features).

The derivational step illustrated earlier in (25) and (26) will now be written to likewise indicate explicitly that the C′ constituent can check a `-wh` feature.

(25′,26′)



As in Stabler (1997), features in a sequence are checked in a left-to-right order, so just as the sequence `+case-tp` on 'like' at the beginning of (31′) encodes that it will "check Case" *and*

then "act as a TP", the sequence `+wh-cp` here encodes the fact that 'does John seem to like pictures of' will "check wh" *and then* "act as a CP". It may be helpful to think of a feature sequence of the form `+a-b` as analogous to a category of the form B/A in categorial grammar (ignoring directionality of slashes), where a feature sequence of simply `-a` is analogous to the category of A. In categorial grammar, something of category B/A can combine with something of category A to produce something of category B. Similarly, something bearing `+wh-cp` (analogous to CP/WH) can combine with something bearing `-wh` (analogous to WH) to produce something bearing `-cp`.

From this point onwards, the features I write in syntactic structures are the only thing that dictates the combinatoric possibilities of their bearers.[8] I will generally continue to write conventional labels (e.g. C′, VP) as well, but these are only there as an aid to the reader.

4.3 A more complete example

Having looked closely at how specific "external merge" and "internal merge" parts of derivations proceed, we can now step through the bigger picture of a derivation. It will be useful to consider, first, a case where a moved element makes a phonological contribution in each of the positions it occupies; for this purpose, let us adopt the simple hypothesis that resumptive pronouns are an instance of such a phenomenon, i.e. resumptive pronouns are phonologically non-null residues of wh-movement (Boeckx, 2003). For illustration I will use the relative clause in (32). While resumptive pronouns are rarely considered acceptable in English, we can take (32) as a model of equivalent sentences in some other language with more promiscuous resumption.

(32) I visited the city which John likes it

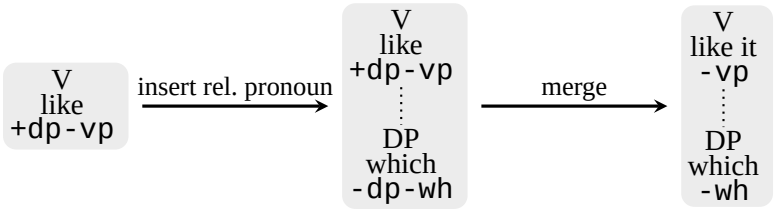
As we have seen, the approach I am advocating here will suggest that for much of the derivation, the moving relative pronoun 'which' is "not in" the direct object position, in a certain somewhat unconventional sense. One might therefore worry that this system might not be able to deal with any phonological and/or semantic effects that the base-position occurrence of such a moving element is usually thought to be responsible for. The purpose of the resumptive

[8] There is a subtle exception to this in the treatment of counter-cyclic adjunction below, although I will gloss over it for the purposes of this paper; see Hunter (2010§2.6) for details.

pronoun in this example is to address this worry. I will take the resumptive pronoun as a sort of place-holder for whatever one might want the underlying occurrence of 'which' in the direct object position to trigger: if not a resumptive pronoun, then perhaps object agreement on the verb, or (almost certainly) a semantic variable in the appropriate thematic position.

To begin, the relative pronoun must be merged into the direct object position. The two steps that achieve this are not dissimilar to the two steps that merge 'John' into the subject position in (31). They are shown in (33). Like (31), this shows a new element (the relative pronoun) being inserted and thus placed as a daughter to the root node ('like'). Note that insertion is not symmetric: inserting α into β establishes α as a daughter of the root of β, while inserting β into α does the reverse. So 'John' was inserted into the leftmost structure in (31), and the relative pronoun is inserted into the leftmost structure in (33).[9]

(33)



Now consider the second step of (33), the merge step, and compare it with the merge step in (31). In each case a "positive" feature on the root node is checked against a corresponding "negative" feature on one of its daughters and the PF interpretations of these two nodes are composed (concatenated). The `+dp` feature on 'like' encodes subcategorisation requirements; one could notate it with something like '[__ DP]' if one preferred. Alternatively, it may be helpful to think of a VP/DP and a DP combining to form a VP in a categorial grammar. But whatever the notation and terminology ("subcategorisation", "selection", "features", "probes"), the point is that 'like' comes with an unsaturated syntactic slot (represented here by `+dp`) and that the relative pronoun is something that can saturate this slot (represented here by `-dp`).

Where (33) differs from (31) is in the fact that when the relative pronoun's `-dp` feature is discharged by merge, other features remain unchecked on it (specifically, `-wh`). This is why we

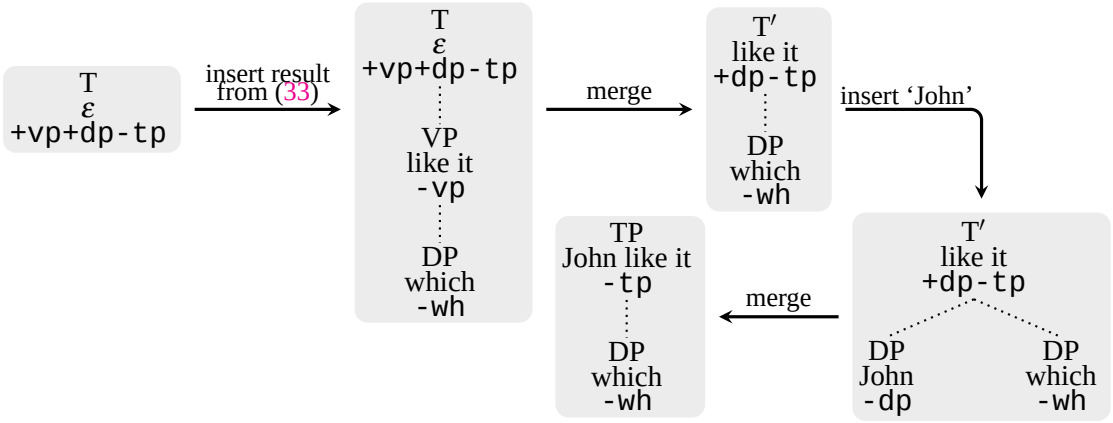
[9]This asymmetry between inserting A into B and inserting B into A does not create a "choice" of how to proceed with derivations that was not already present on conventional minimalist views. There is no escaping the fact that generally, when combining two things, one of them projects and the other doesn't. Combining A and B such that A projects over B is achieved by inserting B into A and then applying merge to the result; combining them such that B projects over A is achieved by inserting A into B and then applying merge to the result.

are left with a two-piece structure even after merge applies, with 'which' waiting to (re-)merge at some future point. Notice that this two-piece structure is exactly what we would expect to have at this point in the derivation (compare with (21) and (25)): it is a VP constituent with PF interpretation 'like it', out of which will move a constituent with PF interpretation 'which'.

I have not provided any explanation for *why* the relative pronoun contributes to PF interpretation as 'it' in its base position and as 'which' in its higher position. (Note that the choice to write 'which', rather than 'it', say, on the daughter node of the pre-merge structure in (33) is arbitrary.) One might ask *why* 'it', rather than 'which' or anything else, is added to the PF interpretation of the root node. This is a valid question but it is not specific to the novel conception of merge and move that I propose. On a view where a complete tree structure is built, with the relative pronoun associated with both the object position and the operator position (whether via copies, or indexed traces, or multidominance, etc.) and then subsequently PF-interpreted, the same question arises, namely why the PF-effect of the relative pronoun's association with the object position is 'it', while that of its association with the operator position is 'which'.

Continuing now with the derivation of the relative clause in (32), the VP that has been derived must be merged as the complement of the T head, and then the subject 'John' as the specifier of it. This is achieved as shown in (34).

(34)

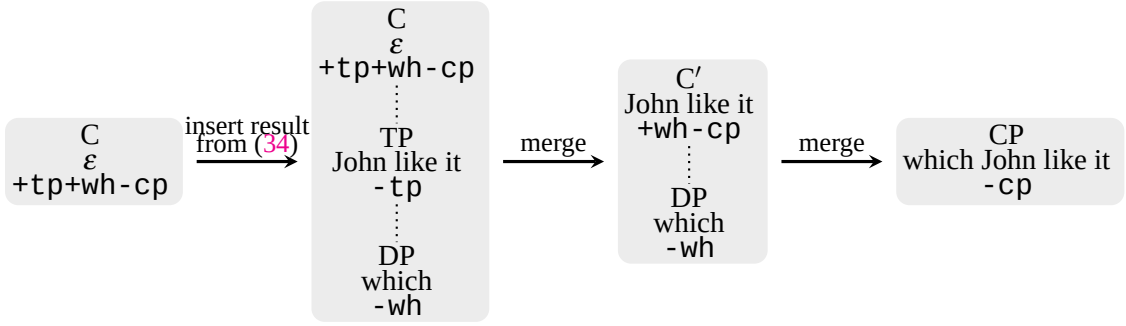


The first two steps are similar to those shown in (33), but differ in that the expression being inserted (i.e. the VP derived in (33)), comes with a moving sub-part of its own, the relative pronoun 'which'. When 'like it' is incorporated into the PF interpretation of the root node upon merge, this larger compound "inherits" the waiting wh-phrase. (I assume here that the PF interpretation of the T head is the empty string ε, so the PF interpretation of the T′ unit that results from combining this with 'like it' is 'like it'.) In the final two steps of (34), the subject

'John' is inserted and then merged, just as in (31) above. The wh-phrase waits patiently in the wings. The final structure is what we would expect: it is a TP constituent with PF interpretation 'John like it', out of which a wh-phrase 'which' will move.

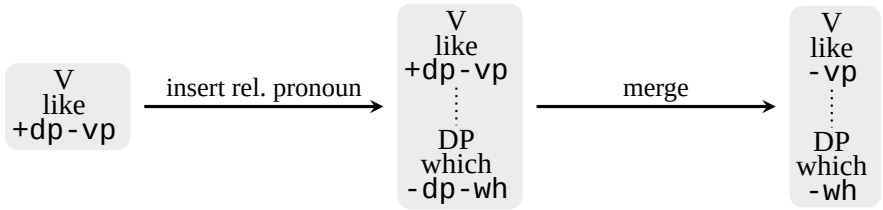
To complete the derivation of the relative clause, this TP must be merged as the complement of the C head and the wh-phrase raised to its specifier position; see (35). The third structure shown in (35) is what 'which' has been waiting for all this time: the feature on the root node that is now ready for checking is a `+wh`. Thus merge applies, checking this with the `-wh` feature of 'which'. Note that there is no need to treat this (internal) merge step in any way differently from the previous (external) merge step that checked the `+tp`/`-tp` features.

(35)



This completes the derivation of the relative clause, with resumption, in (32). Now, for comparison, consider the variant of (32) where the moving wh-phrase does *not* make a phonologically overt contribution in its base position (i.e. 'the city which John likes'). The derivation is entirely analogous except that when merge applies to check the `+dp` and `-dp` features corresponding to the object position of the verb 'likes', the PF interpretation of the root node is left unchanged (or, it is concatenated with the empty string ε, rather than with 'it'). This is shown in (36); compare with (33).

(36)



As noted earlier, the choice between (33) and (36) is directly analogous to the choice that one must make, in more conventional systems, between "pronouncing" and "not pronouncing" a trace or a lower copy/occurrence. The difference is simply that in the current framework it confronts us relatively early in the derivation, in amongst all the syntactic structure-building

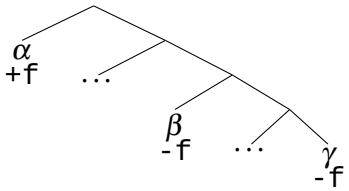
operations, rather than at a distinct post-syntactic linearisation stage.[10]

4.4 Two potential objections

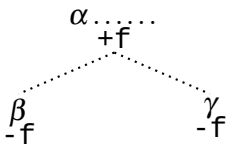
I now return to the two concerns noted in passing above: now we no longer have the entire tree structure available throughout the derivation, one might wonder (i) how we can enforce certain constraints on movement, and (ii) how we can carry out proper semantic interpretation.

First: to illustrate the challenges posed by movement constraints, consider the generic "minimality" configuration in (37). In the framework I have presented, this would be represented by the structure in (38): a constituent headed (let us assume) by α, out of which β and γ are both destined to move.

(37)



(38)



The problem arises because β and γ, each of which is "competing" to re-merge and check its `-f` feature, have exactly the same status in (38). Under conventional constraints along the lines of Relativized Minimality (Rizzi, 1990) or the Minimal Link Condition (Chomsky, 1995), the fact that β intervenes between α and γ would entail that β "wins" and so γ is not permitted to move in this situation.[11] But this fact is not recorded in (38).

In other work where structures like (38) have been used to model minimalist derivations (e.g. Stabler, 2001; Kobele, 2006), a particularly strong minimality-like constraint is typically

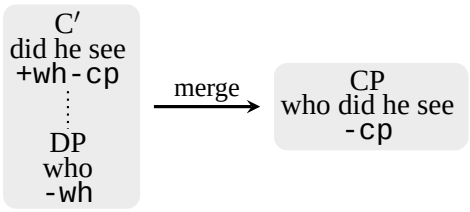
[10]The only analyses that could not be straightforwardly carried into the current framework are those where the PF realisation in a certain position depends on something that occurs later in the derivation. For example, one analysis of English wh-phrases would have it that the pronunciation of a wh-phrase that merges into an object position actually depends on whether there is another wh-phrase in subject position (or elsewhere): if there is, then the object wh-phrase should be pronounced in situ, but if there is not, it should not. One approach to extending the current system to handle this kind of analysis is to suppose that both options are derivational possibilities and that the undesired one is subsequently filtered out; see Kobele (2006:178–181) for a detailed version of this.

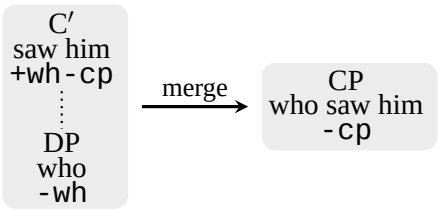
[11]The situation is not significantly altered if we assume that "defective intervention" exists: if it turns out that it is not only candidate movers that can act as intervenors, then the relevant non-moving intervenors (however these are identified) will also need to remain disconnected or "held out" in the way that β and γ are in (38).

adopted. This "Shortest Move Constraint" (SMC) dictates that when such competition for features exists, *neither* of the would-be movers is permitted to move. Thus in (37) neither β nor γ is permitted to move. Whenever it arises that two constituents are simultaneously looking to move to check the same type of feature, the derivation is doomed. Note that in order to state this stronger constraint, the structure in (38) is sufficient. What (38) fails to encode, problematically for Rizzi-style minimality restrictions, is the relative positions of β and γ; but if the very existence of competition between β and γ rules out the movement of either, then the relative positions do not matter. Exactly how well the SMC fares empirically, compared to Rizzi-style restrictions, remains largely an open question. For a response to one of the obvious potential problems, multiple wh-fronting, see Gärtner & Michaelis (2010).

Finally, note that this problem only arises for constraints which make reference to the *relative* positions of various constituents. The choice not to maintain entire structures does not interfere with our ability to enforce absolute constraints on movement — for example, prohibiting movement out of adjuncts, or out of some other kind of domain we take to be impermeable. Since we have adopted the point of view where constituents that are destined to move at some future point in the derivation are explicitly identified, such absolute constraints can be imposed at the point where the impermeable domain is established. Essentially, we can prohibit extraction from adjuncts, for example, by prohibiting *adjunction of* constituents that have moving sub-parts; see Frey & Gärtner (2002) for a simple instance of this strategy. The account I propose below for adjunct islands is also of essentially this form.

I turn now to the second potential objection, concerning semantic interpretation. It may appear that with the internal structure of derived expressions no longer maintained throughout the derivation, we will encounter problems when trying to compute semantic values. For example, one might worry that having no record of where a wh-phrase is "coming from" will prevent its binding of the appropriate variable: if, when wh-movement applies at the end of the derivation of 'Who did he see?', there is no sense in which 'who' is moving *from* the object position, how can we ensure the object variable is the one bound? How can this situation, shown in (39), be distinguished from the analogous subject-question case in (40)?

(39) 

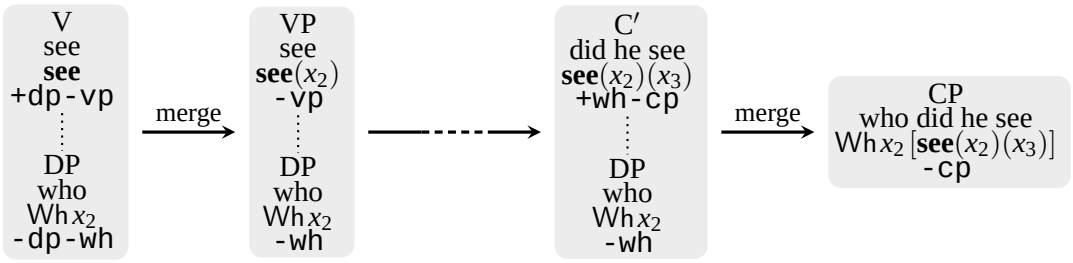
(40) 

These diagrams make it clear that these two wh-movement steps are structurally indistinguishable. An account based on the idea that the semantic effect of re-merging a wh-phrase is roughly to (i) place a variable in the source position and then (ii) bind it from the target position, will not be compatible with this.

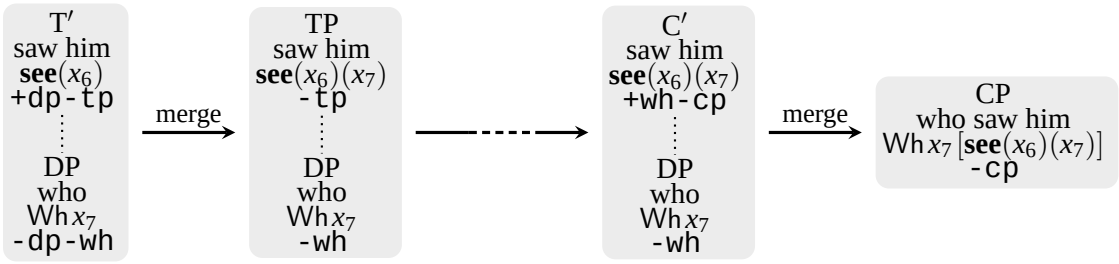
But if we carry over, from discussion in earler sections, the slightly modified understanding of how derivational operations and interface interpretation interract — i.e. make the same adjustment to our understanding of LF interpretation as we have already made for PF interpretation — this apparent problem disappears. The idea, in effect, was the following: while one *can* understand the PF effects of wh-movement to be to (i) remove a wh-phrase from some position, (ii) replace it with a gap or resumptive pronoun, and (iii) put the wh-phrase in a new position, one *can also* understand the gap or resumptive pronoun to be entirely a product of the earlier interaction with the base position; recall the discussion of (33) and (36). Accordingly, just as the phonological gap is "already there" in the structures to which merge applies in (39) and (40), the variable is "already there" too; the effect is very reminiscent of Cooper Storage (Cooper, 1983).[12] Leaving aside technicalities (see chapter 4 of Hunter (2010) for formal details), this is intuitively illustrated in (41) and (42), which show both the base position and operator position merging of the wh-phrase, with semantic values as the third line at each node. In each case, the first merge step places a variable in a thematic position, and the (re-)merge step incorporates the binder of that variable. The choice of indices is arbitrary. What is important is that, although (39) and (40) obscure the fact, there is a point in the derivation at which the wh-phrase is able to place "its own variable", however distinguished, in the object position in (41) and in the subject position in (42). This is analogous to the way the wh-phrase is able to place (the PF realisation of) a resumptive pronoun in (33).

[12] Thinking of traces as semantic variables, one could picture 't_i' and 't_j' at the top of the dotted lines in (15).

(41)



(42)



In short, the suspicion that these reduced structures will not permit proper semantic interpretation is founded on the traditional assumption that all semantic interpretation should occur "after" all syntactic structure-building.[13] The assumption that PF and LF interpretation must take place after all syntactic operations has, I think, made it unnecessarily difficult to flesh out the intuition of move as re-merge. What usually distinguishes move from merge, intuitively, is that in the case of move one of the affected elements is drawn from a position somewhere within the derived tree structure; but typically this position is not the position where (all of) the element's interpretation will be realised, hence the need to move it (e.g. the operator component of a wh-phrase will not be interpreted in object position). This apparent difference between where merge and move draw their operands from can be eliminated if we consider interpretive consequences more incrementally, and realise that it is only the variable component of a wh-phrase that ever needs to go into object position.

5. Complements and specifiers and local interpretation

The discussion of semantic interpretation just above focussed on how we can correctly implement the desired semantic relationships between the pre-movement and post-movement positions of particular constituents. I argued that certain questions raised by the reconception of movement introduced in §4 can be straightforwardly answered, essentially by treating LF interpretation in the way I had already suggested for PF interpretation. In this section I will consider

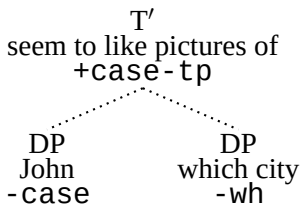
[13] This assumption is perhaps in some sense no longer generally accepted, following Uriagereka (1999) and Chomsky (2001). But the connection remains unclear between the sense in which pieces of structure are "interpreted" in such discussions of phases/spellout, and the task of compositionally determining semantic values.

"local" semantic interpretation: the semantic composition that occurs at individual merge steps, putting aside issues of whether the constituents involved have moved or will move in the future.

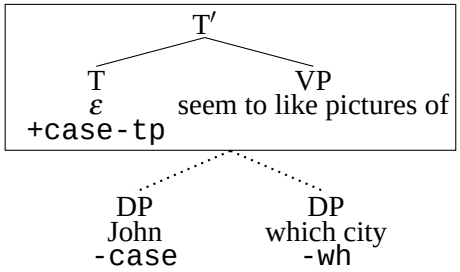
The key idea will be as follows. The reduced representations adopted in §4 *are* insufficiently structured to permit proper semantic interpretation, but *not* because proper semantic interpretation must be able to "see" where moving constituents originated; this kind of concern was addressed above. Rather, they are insufficient because proper semantic interpretation, according to independently-motivated assumptions, must be able to see certain *local grammatical relations*, which are likewise obscured. Consequently, we should back off slightly from the very structurally impoverished representations adopted in §4, and take a small step back in the direction of the fully articulated conventional tree structures. Because the semantically-relevant grammatical relations are very local, only a small window of very-recently-built structure needs to be maintained. Thus what we end up with will look much more like the reduced structures in §4, which provide no window to previously-built structure at all, than like the original conventional tree structures, which essentially provide an unboundedly large window.

How big, exactly, will this small window be? I will argue that independently-motivated hypotheses concerning interpretation suggest that the window should show only the internal structure of the current maximal projection. Consider the tree in (14), which was earlier reconstrued so as to have "only" the structure in (16′). I will argue that this went slightly too far; instead, the internal structure of the TP that is currently being built should be represented, as shown in (43). Although the notation looks a bit unwieldy, the relevant concepts are familiar. Where (43) differs from (16′) is that it provides some information about the internal structure of the T′ node that is atomic in (16′); this is shown in the boxed part of (43).

(16′)



(43)

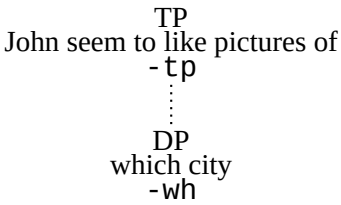


Note that the rest of what the two representations encode is identical: the T′ constituent, whether structured as in (43) or not as in (16′), has two sub-parts waiting to (re-)merge in the

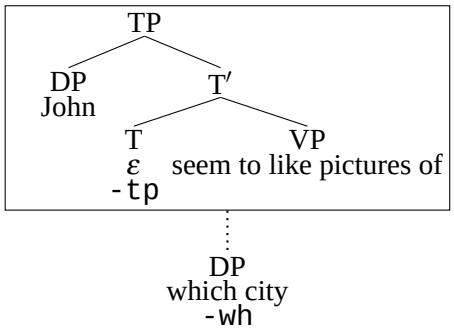
familiar way, namely ‘John’ and ‘which city’. Note also that as mentioned above, the boxed part of (43) does not encode *all* of the internal structure of the T′. It records only the structure of the current maximal projection, by which I mean only the structure of the TP that is in the midst of being built: we can read off the fact that the T head has taken a VP as its complement, that the T head is phonologically null and that the VP has PF interpretation ‘seem to like pictures of’ (and the LF interpretations of these two pieces too), but we know nothing about the internal structure of the VP. This is what was meant by the idea that the “sliding window” onto internal structure lets us see only a short history of very local structure. This internal structure is represented by solid lines, in accordance with the distinction between the two kinds of links noted in discussion of (15). Finally, note that the only features that appear inside the boxed part of the new representations like (43) will be on the head: any other pieces with remaining unchecked features will necessarily be outside the box because they will re-merge later to check these features.

For further illustration of the idea behind (43), the result of applying merge to this structure to complete the TP is illustrated in (44), alongside its less-structured counterpart from earlier, (24′). Details of how derivations proceed with these structures will be explained shortly.

(24′)



(44)



Now, *why* am I suggesting that it is important to maintain this small window of structure? The reason is that the complement and specifier relationships that it makes visible play an important role in determining exactly how the PF- and LF-interpretations of the relevant pieces are composed. To illustrate the intuition, I will begin by considering the more straightforward case of PF interpretation; then I will make an analogous argument with respect to LF interpretation, which is more subtle but ultimately a better fit for the bigger picture I am proposing.

Let us consider more carefully the effect of applying merge to the structure in (16′), to see

what goes wrong with this simpler structure and how (43) solves the problem. As a result of checking the `-case` feature on 'John', 'John' should be combined with the string 'seem to like pictures of'. But in what order: 'John seem to like pictures of' or 'seem to like pictures of John'? While all the derivations I have presented up to this point "do the right thing", there has been no consistent rule for the order in which two strings are combined: in this particular case, the desired result is that the piece checking a `-f` feature, here 'John', appears on the left; but there have been other cases where the piece checking a `-f` feature appears on the right. The crucial additional information we need, of course, is whether this merge step is establishing a complement or a specifier. This information is missing in (16′), but can be easily read off (43).

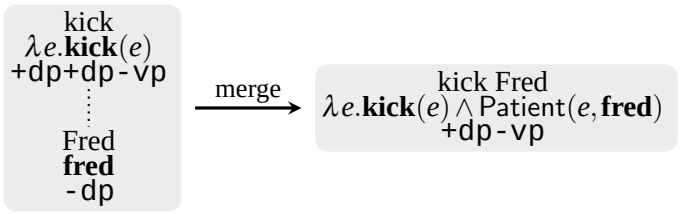
With this idea, grounded in questions of PF interpretation, in mind, we can now turn to the analogous argument for LF interpretation. I adopt neo-Davidsonian event semantics (Parsons, 1990; Schein, 1993), such that (at the VP level, taking subjects to be merged VP-internally now[14]) the semantic interpretation of (45a) is (45b).

(45) a. John kicked Fred

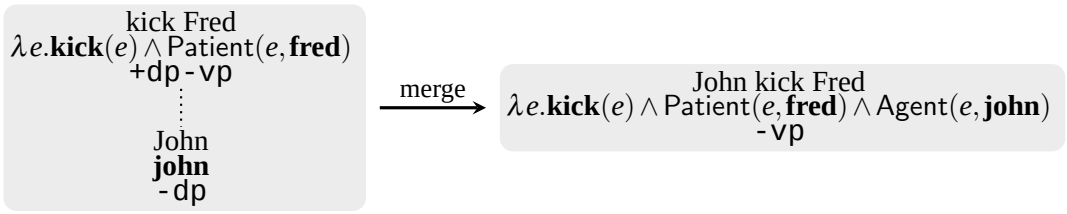
b. $\lambda e.\textbf{kick}(e) \wedge \mathsf{Agent}(e, \textbf{john}) \wedge \mathsf{Patient}(e, \textbf{fred})$

Suppose that the lexical meanings of 'John' and 'Fred' are simply the individuals **john** and **fred** respectively, and that the lexical meaning of 'kick' is the event predicate $\lambda e.\textbf{kick}(e)$. Then we expect that the semantic value in (45b) should be pieced together roughly as in (46). But if the root nodes have no syntactic structure, then this presents a problem analogous to the one just discussed for linearisation. When combining a verbal event predicate with an argument α, the desired effect is sometimes to add the conjunct $\mathsf{Patient}(e, \alpha)$ and sometimes to add the conjunct $\mathsf{Agent}(e, \alpha)$; just as the desired PF effect is sometimes to linearise the argument on the right and sometimes to linearise it on the left.

(46) a.



[14] I ignore any additional *v*P layer for ease of exposition, but nothing crucial hinges on this.

b. 

To resolve this we can suppose that the *structural relationship* between a verb and one of its subcategorised arguments plays a role in determining which thematic relation is used. To a first approximation, complements are patients or themes and specifiers are agents (Baker, 1988). While the details of these correlations are notoriously hard to pin down (Dowty, 1991; Schein, 2002), I only need to adopt the relatively weak assumption that structural position plays *some* important role in thematic role assignment, whatever the correct details may be. In other words, correctly composing (45b) from the meanings of the VP's parts requires knowledge of the structural relationships among these parts, just as composing the PF interpretations in the correct left-to-right order does.

More generally, following Carlson (1984) and Pietroski (2005) among others, I adopt the idea that semantic composition is basically about *conjoining* certain predicates, and that the predicate with which a verb's meaning is conjoined when it combines with an argument α is of the form $\lambda e.R(e,\alpha)$ for some thematic relation R, i.e. the predicate satisfied by events to which α bears relation R. For reasons of space, I can only illustrate with the case of simple VP interpretation here; see in particular Pietroski (2005) for many more details. The important idea is that semantic composition proceeds by combining one-place predicates, with certain details of these predicates (e.g.. their linking thematic relation) determined by local structural relations; analogously, PF composition proceeds by concatenating strings, with certain details of this concatenation (i.e. choices of left and right) determined by local structural relations.

One might ask whether the interpretive work of choosing thematic relations *must* be allocated to structural relations in the way I have suggested. One could instead hypothesise that the lexical meaning of 'kick' is the function in (47), and that when the verb combines with its subcategorised arguments this function is applied to **fred** and then to **john**. This way the desired allocation of "agent-hood" and "patient-hood" is bundled into the lexical semantics of the verb, and as a result semantic composition could take place at each merge step *without* knowledge of local structural configurations.

(47) $\lambda y \lambda x \lambda e.\mathbf{kick}(e) \wedge \mathsf{Patient}(e,y) \wedge \mathsf{Agent}(e,x)$

Note that one *could* use a similar mechanism to enforce the desired allocation of "left-ness" and "right-ness" in PF interpretation: one could hypothesise (as categorial grammar does, in effect) that the lexical phonology of the verb is the function $\lambda s \lambda t.\ t$ kick s, and that when the verb combines with its subcategorised arguments this function is applied to the string 'Fred' and then to the string 'John'. But defining such a function for each transitive verb individually would arguably miss certain generalisations about the relative orderings of subjects, verbs and objects, so one might prefer to suppose that the three lexical items here are phonologically specified simply as 'kick', 'Fred' and 'John', and that these strings are composed in accordance with certain structural relations. Similarly, (47) arguably misses certain generalisations about the assignment of thematic roles to objects and subjects. So we need not assume that this fully-fledged compositional recipe is "obviously" the correct lexical verb meaning.[15]

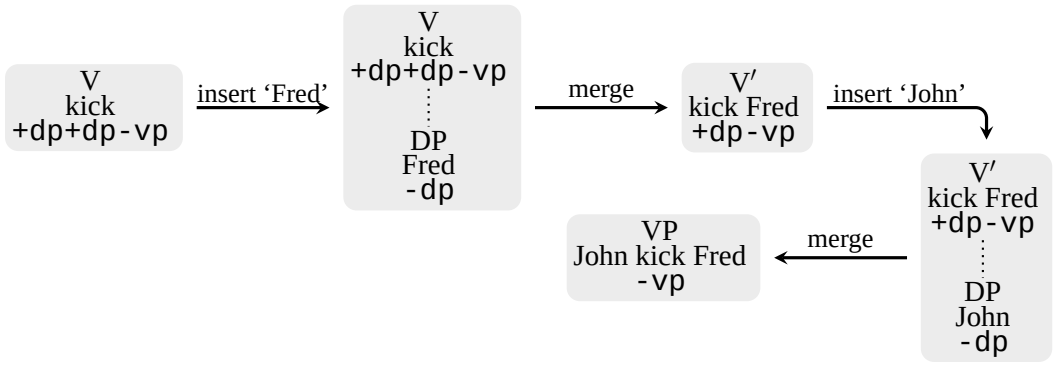
If we suppose that in the course of a derivation a small amount of local structure remains visible, as in (43) and (44), then this kind of interpretive strategy will be naturally accommodated. What this means is that we must dissociate two kinds of work that the merge operation was responsible for in §4: the first is the checking of features, and the second is the formation of a new unit, with phonological and semantic values composed from those of the units whose features were checked. These two duties will now be performed by two distinct operations. I will retain the term "merge" for the the operation which checks features and thereby establishes head-argument relationships, but does not itself perform any semantic or phonological composition. The other operation, which semantically and phonologically composes units among which merge has established head-argument relationships, I will call "spellout"; following Uriagereka (1999), the intuition will be that a sequence of merge steps builds some tree-like structure which is then "flattened" into a word-like object, and is manipulated only as such for the rest of the derivation.

Thus the derivation of the VP considered above proceeds not as in (48), as suggested in §4,

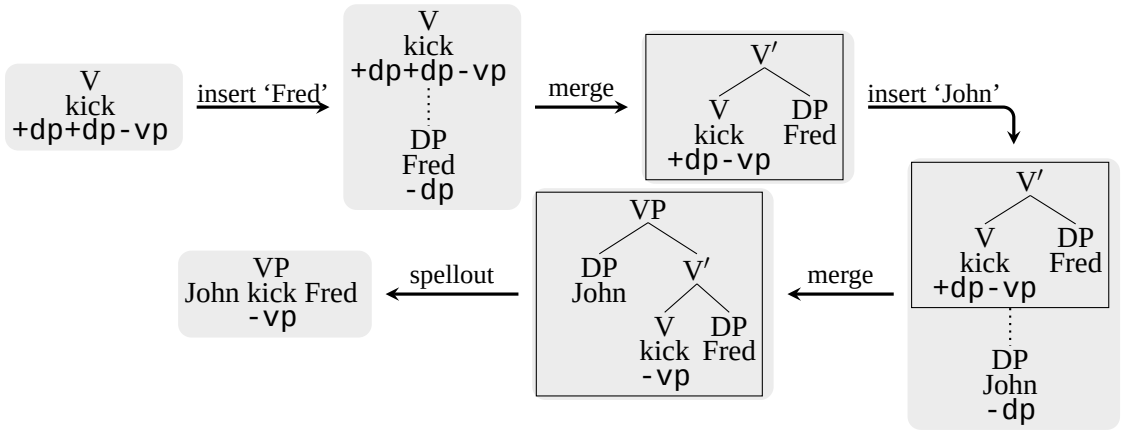
[15] Besides possibly missing such generalisations, there are phenomena which turn out to be difficult or impossible to analyse if we take the lexical meaning of 'kick' to be (47) rather than $\lambda e.\mathbf{kick}(e)$. I can not properly address this subtle issue here, but see Schein (1993), Carlson (1984) and Williams (2008) for arguments to this effect.

but rather with all phonological and semantic composition taking place at once when spellout applies, as shown in (49).

(48)



(49)

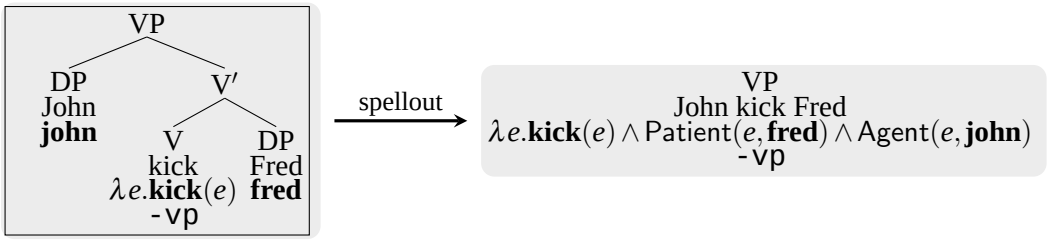


The pattern of insertion and feature-checking steps is unchanged. What is importantly different about (49), however, is the fact that all interpretive *composition* — both phonological and semantic composition, although (49) only illustrates the phonological side — is "delayed" until spellout applies. Up until the point where spellout applies, no new phonological or semantic values have been constructed; the ingredients 'kick', 'Fred' and 'John' are just arranged in a certain hierarchical structure.[16] When spellout applies, the phonological and semantic values of these three ingredients are composed, *in accordance with the structural relations among them*, to produce a single new unit. Because of the particular hierarchical structure that has been established through merge operations, spellout can "see" that 'John' should be treated as a complement (linearised on the right, marked as a patient) and that 'Fred' should be treated as a specifier (linearised on the left, marked as an agent).

While semantic values are omitted in (49) for readability, (50) shows the "interpret-and-flatten" effect of the ending application of spellout for both PF and LF interpretation.

[16] These trees should be thought of as unordered in the same way that syntactic phrase markers usually are, with linearisation determined only when spellout applies.

(50)



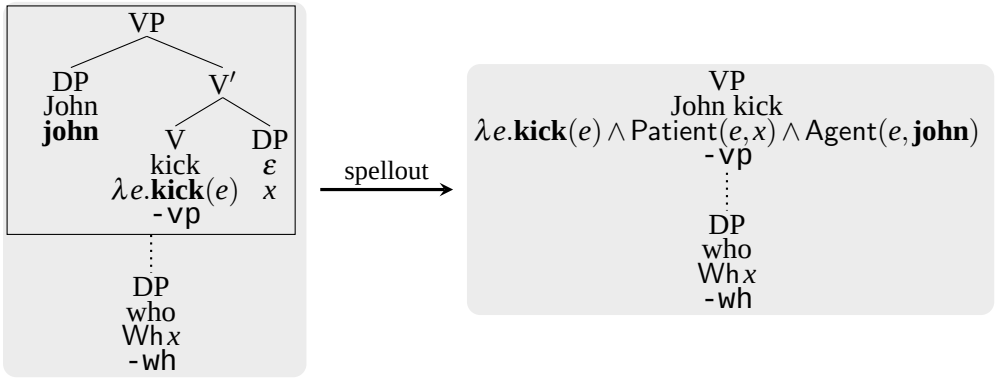
Note that the object that is produced by spellout is not different in kind from those with which the system already works; we end up with exactly the same object in (48) as we do in (49). Once it has been flattened into an unstructured unit, this VP can be inserted and merged into the next maximal projection up (say, TP), just as the DP unit 'Fred', for example, was inserted and merged into the VP. Only flattened objects, of the sort already dealt with in §4, can be inserted,[17] so one can think of spellout as preparing an XP for insertion into a subsequent projection. But spellout is not different in kind from the existing operations of insert and merge; it is simply another derivational operation that can be applied in certain circumstances. (There is no relevant sense in which, as a result of applying spellout, any semantic or phonological material "goes anywhere" or is "sent away".) However one might like to think of what spellout does, it was already a part of what merge did in (48).

One final aspect of these small trees that are constructed and then flattened is important to note here. Recall that when something merges into a non-final position (e.g. when a wh-phrase merges into a thematic position), its phonological and semantic contributions are split into those that will be interpreted at the current position (e.g. a resumptive pronoun, a phonetically null element, a semantic variable) and those that will be interpreted at the position where it later re-merges (e.g. a phonologically overt wh-phrase, a semantic operator). Now that the semantic and phonological content relevant to the current merging position is not immediately composed, this splitting is illustrated somewhat more clearly. A variant of (50) where the object will later undergo wh-movement is shown in (51); and a variant where the object will later undergo a semantically vacuous fronting operation (e.g. 'Fred, John kicked') is shown in (52). I write '`-f`' for the feature that triggers the fronting operation in the latter. I write null semantic values

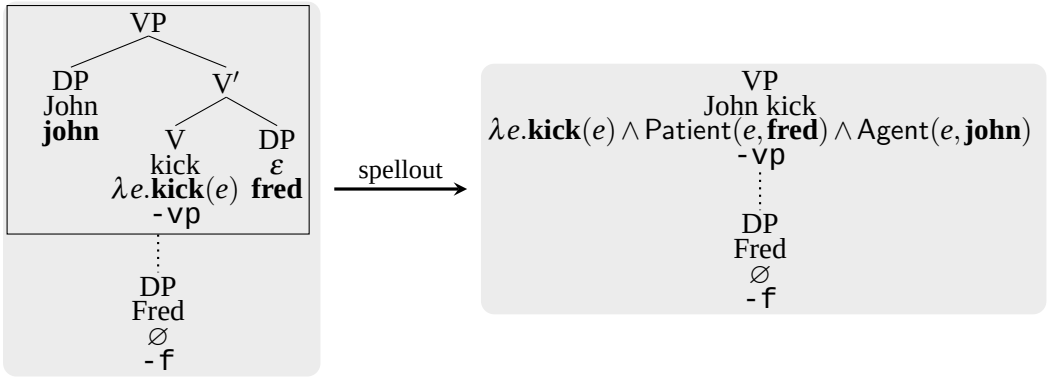
[17] This follows from the assumption that we should enrich the representations used up to the end of §4 only by the amount that is empirically necessary to allow proper PF and LF composition. It turns out that using a list to store the structure of the current projection does allow proper PF and LF composition but does not allow un-flattened structures to be inserted, so I adopt this restrictive hypothesis. See Hunter (2010) for details.

as '∅'.

(51)



(52)



Notice that when 'who' was merged in the course of producing the structure to which spellout is applied in (51), its "locally-relevant" PF and LF content — namely the empty string and a variable — are placed in the complement position to be interpreted as appropriate, while "the rest" of its PF and LF content remains disconnected. This is exactly analogous to the way we saw a wh-phrase insert a variable earlier in (41), the only difference being that this empty string and variable are not composed with the rest of the VP until spellout applies. The interpretable content of the object in (52) has likewise been split. As in the wh-movement case, all the PF-relevant content of this object 'Fred' goes to the disconnected piece that will re-merge later in the derivation; but here, since I assume the fronting is semantically vacuous, all the LF-relevant content goes into the complement position. Thus the semantic product of spellout in (52) is not different from that in (50).

The derivational mechanisms that we have arrived at now leave room for adjunction to be naturally accommodated.

6. Adjuncts find a home

As foreshadowed in §3, I will propose that the phenomenon of "being adjoined to XP" can be analysed as being *inserted*, and then composed with the rest of XP when *spellout* applies, without ever undergoing merge. The division of labour between merge and spellout introduced in the previous section, with merge establishing head-argument relations and spellout performing interpretation, makes it possible for adjuncts to be introduced (i.e. inserted) and then taken into account by interpretive operations (i.e. spellout) without interfering with the more richly-articulated structure established between heads, complements and specifiers.

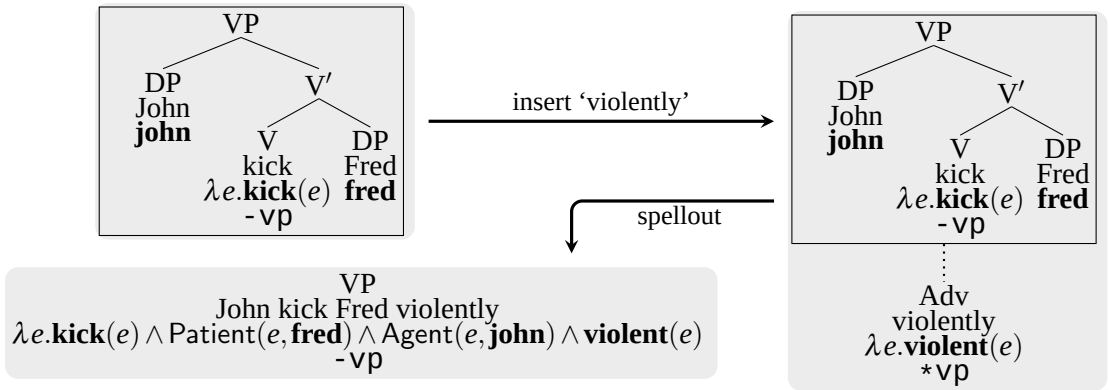
More specifically, consider (the verbal domain of) the sentence in (53) with a VP-adjunct.

(53) a. John kicked Fred violently

b. $\lambda e.\mathbf{kick}(e) \wedge \mathsf{Agent}(e, \mathbf{john}) \wedge \mathsf{Patient}(e, \mathbf{fred}) \wedge \mathbf{violent}(e)$

The derivation of this VP begins in the same way as the similar one without the adjunct in (49): the two arguments are inserted and merged. The continuation of the derivation from this point onwards is shown in (54): this consists of inserting 'violently' and then immediately applying spellout. (This will actually turn out to be one of two ways 'violently' can be introduced, but we will focus on this one alone for now.)

(54)



Note that 'violently' has a new kind of feature, written with '*', which encodes that it is an adjunct. In particular, since it has the feature *vp, it is a VP-adjunct. Recall that it is not my aim to eliminate adjunction, in the sense of reducing it to something else entirely, and consequently it is to be expected that certain lexical items are specified as adjuncts (i.e. things that adjoin, whatever that may turn out to mean; this is the question under investigation).[18] My aim is to present an interesting account of the derivational processes that *carry out* this

"adjunction", from which the various distinctive properties discussed in §2 emerge. The kind of stipulation I am striving to avoid is the stipulation of an ad hoc *adjunction operation* with the desired tailor-made properties.

What I am proposing in (54) is that when spellout flattens a particular XP, any units further down in the tree bearing the appropriate '`*`' feature are incorporated into the newly-produced unit. Here, (the head of) the XP being flattened bears `-vp`, and so 'violently', bearing `*vp`, is incorporated as part of the new VP unit. The point in the derivation at which 'violently' is inserted here, relative to the insertion and merging of 'Fred' and 'John', does not matter: there are a number of different orders in which the three insert steps and two merge steps can produce the structure to which spellout applies in (54).

This raises two obvious questions. First: how exactly is 'violently' incorporated into the (PF and LF interpretations of) the newly-formed unit? And second: why is this insertion-without-merge idea a sensible treatment of adjuncts — as opposed to say, the reverse idea that adjuncts are *inserted and merged* whereas *arguments* are *only inserted*?

The answers to these two questions are closely related. Recall that we have separated structure-building (now done by merge) from interpretive composition (now done by spellout), and that the reason for postulating the purely structure-building merge operation was to establish certain relationships which identify exactly how certain pieces should be composed. I have adopted the position that semantic interpretation is essentially about *conjoining predicates*, and that building up VP-internal structure via merge establishes the structural relations that determine exactly *which* predicate should be conjoined when a verb is composed with an argument (e.g. $\lambda e.\mathsf{Agent}(e,\mathbf{john})$ or $\lambda e.\mathsf{Patient}(e,\mathbf{john})$). The crucial idea behind the proposal to treat adjuncts as in (54) is that the proper interpretation of adjuncts *does not* depend on any such structure of the sort built by merge: the composition of the predicate $\lambda e.\mathbf{violent}(e)$ with the meaning of a verb (or verb phrase) does not need any extra disambiguating help of the sort

[18] It follows from this that not only will adverbs like 'violently' be lexically marked with adjunction features, but also the heads of any other phrases that adjoin. For example, a PP 'in Boston' that adjoins to a VP will likewise bear a `*vp` feature, which in turn means that this phrase will be most straightforwardly analysed as being composed from the preposition 'in' with feature sequence `+np*vp`. This is directly parallel to the way many categorial grammars would assign 'violently' the category VP\VP, and assign 'in' the category (VP\VP)/NP.

that is necessary to compose **john**, for example, with the meaning of a verb (or verb phrase) (Hornstein & Nunes, 2008). So the answer to the question of exactly how 'violently' is interpreted as part of the VP is, roughly, that it "just is"; in contrast to the way 'Fred' is interpreted *as a complement* and the way 'John' is interpreted *as a specifier*. Accordingly, the idea is that merely inserting 'violently' is all that needs to be done before it is interpreted.

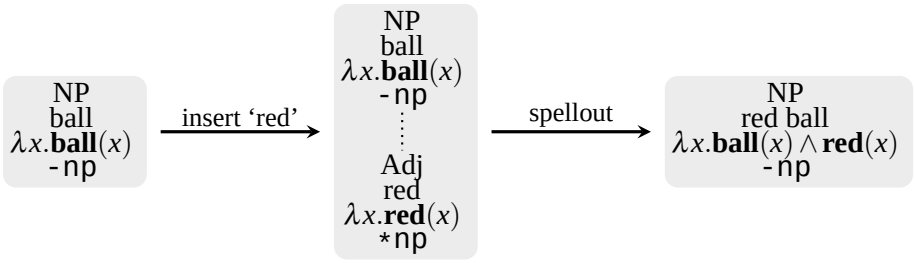
A similar argument can be made for PF interpretation (linearisation). Recall that the analog of identifying which particular thematic relation should be used to link an argument was identifying whether an argument should be linearised on the left or the right. While there are many unanswered questions concerning the linearisation of adjuncts, it is fair to say that their linear positioning is less strict than that of complements and specifiers. Adjuncts can sometimes appear on either side of the phrase they modify, as illustrated by (55) (where t_i indicates a trace left by subject movement out of the VP, which I have abstracted away from above).

(55) a. John$_i$ [t_i kicked Fred] violently b. John$_i$ violently [t_i kicked Fred]

Not all adjuncts exhibit this flexibility: adjectives generally precede their hosts while relative clauses follow them (in English), for example. Whatever ultimately underlies these patterns, being an adjunct clearly does not dictate a particular linear ordering relative to the host phrase, in the same way that being a complement dictates linearisation on the right and being a specifier dictates linearisation on the left. There is clearly no possibility of linearising an adjunct "neither on the left nor on the right", in the way that the meaning of 'violently' is linked neither as an agent nor as a theme in (54); but it does seem to be true that adjuncts are linearised neither *necessarily on the left* nor *necessarily on the right*, which is perhaps the best we could expect.

As mentioned just above, typical examples of adjuncts to nominal expressions include adjectives and relative clauses. Like adverbials, these generally have a conjunctive interpretation, and so the treatment of the adverb in (54) can be extended naturally to these cases too. An example is given in (56).

(56)

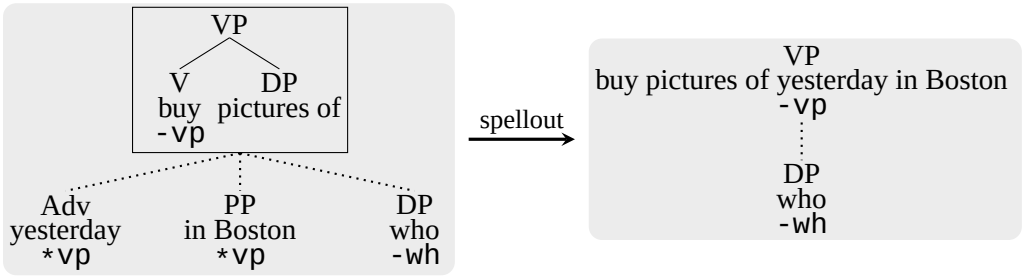


I leave open the details of to what extent nouns take arguments in the way that the verb 'kick' does in (54). There are of course many cases where it seems clear that a noun phrase should be treated as having a structure along the lines of a verb with arguments (e.g. 'John's kicking (of) Fred'). This raises questions such as why the obligatory/optional distinction that often distinguishes arguments from adjuncts in verb phrases does not carry over straightforwardly, and why some additional morphological trimmings appear in the noun phrase case. But these are beyond the scope of this paper.

This analysis of adjunction naturally permits iterability of adjuncts and fits comfortably alongside the existing use of the insert and merge operations to implement movement. When spellout applies, any units marked with the relevant adjunction feature are interpreted as such, and any others are left alone. For example, the application of spellout that interprets the VP in the sentence in (57) is shown in (58), with two adjuncts being interpreted and one moving constituent left waiting to re-merge later (ignoring VP-internal subject positions for clarity).

(57) Who did John buy pictures of yesterday in Boston?

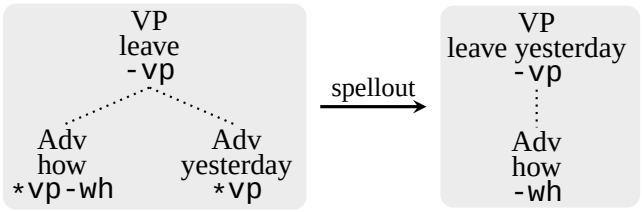
(58)



Combinations of adjunction and movement by a single constituent are straightforwardly handled by combining the relevant features. Just as the feature sequence `-dp-wh` on 'who' means that it will merge once as a DP and then again as a wh-phrase, we can use `*vp-wh` for a word like 'how' which adjoins to a VP (without making a phonological contribution there[19]) and then merges as a wh-phrase. In a derivation of 'How did John leave yesterday?', for example, the relevant application of spellout will proceed as shown in (59). Because of the

`-wh` feature that remains after its `*vp` feature is used at spellout, 'how' remains disconnected, waiting to re-merge later (just as 'which' does after its `-dp` feature has been checked in (36)).

(59)



6.1 Adjuncts and constituent structure

This analysis of adjunction provides a natural explanation for the constituency puzzles discussed in §2.1. The pattern to be explained is that when a particular XP is targeted for movement, an adjunct to XP can be either included in or excluded from the targeted constituent. Arguments do not have this same flexibility. The relevant examples, based on the sentence used in (54), are given in (60). The result of this fronting/clefting is acceptable just when 'Fred' is fronted along with the verb, whether or not 'violently' is included as well.

(60) a. Kick Fred violently, John did.
b. Kick Fred, John did violently.
c. * Kick, John did Fred violently.
d. * Kick violently, John did Fred.

The basic idea behind the account of this pattern will be that the underlying non-fronting sentence 'John kicked Fred violently' has two *semantically equivalent* derivations, differing slightly in the way 'violently' is introduced into the derivation. The first of these two semantically equivalent derivations is the one illustrated earlier in (54), and the way in which 'violently' is introduced there interacts with the fronting transformation to produce (60a); the way in which 'violently' is introduced in the second derivation interacts with the fronting transformation to produce (60b). There is no analogous freedom to choose between two ways of introducing the object 'Fred' into the derivation, hence no flexibility in the way it interacts with fronting.

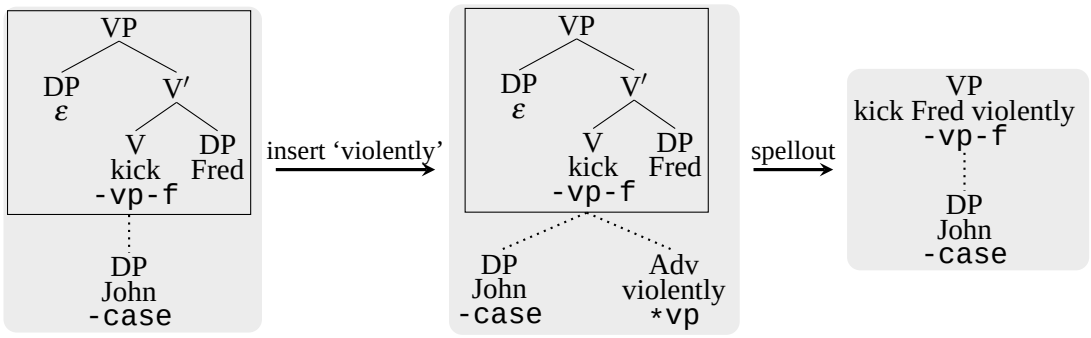
To illustrate the explanation I will assume that the fronting operation in (60) is driven by a `-f` feature on the verb 'kick', which is checked by some functional head bearing the matching `+f` feature; I will call this function head Z. I will also assume for concreteness that the head that

[19] It does, of course, make a semantic contribution. Roughly, it leaves a manner variable, say by adding an event conjunct along the lines of "and event *e* was done in manner *m*", to be later bound by an operator "for which *m*".

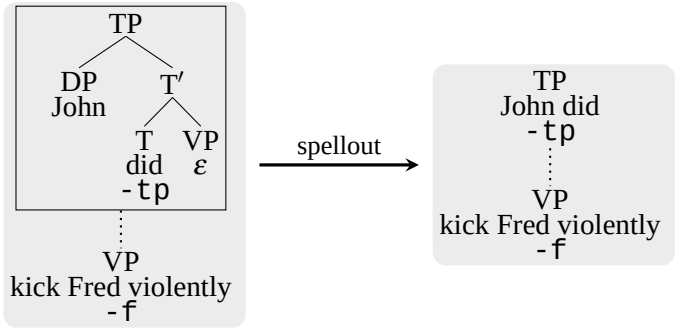
takes VP as its complement is T. I will ignore any semantic effects of the fronting operation, and will not make any claims about the semantics of the TP projection. I will, however, need some notation for the result of semantically composing the T head with its VP complement, whatever the details of this may be: for this I will write simply '$\boldsymbol{F}\left[\phi_{\mathrm{T}}\right]\left[\phi_{\mathrm{VP}}\right]$', where a subscripted '$\phi$' indicates a placeholder semantic value. It will not be important *what* the semantic value of T is, or *how* it is composed with that of the VP; what is important is just that we have some way to notate the composition of VP with its immediate surroundings.

Deriving (60a) is straightforward. The VP is built as shown in (61a). This is essentially the same as (54) earlier, with two small differences. First, I assume that the subject 'John' will subsequently move for Case, so it is held out of the VP projection; this is necessary for VP-fronting to produce the desired results but is not relevant to the central issue, namely adjunction. Second, and more importantly, we have the extra `-f` feature on the verb which is projected to the VP. When the VP is merged as the complement of T, its `-f` feature remains unchecked and so the phonological content of the VP remains detached, as shown in (61b), and waits to remerge when the `+f` feature of the Z head becomes available later; see (61c).

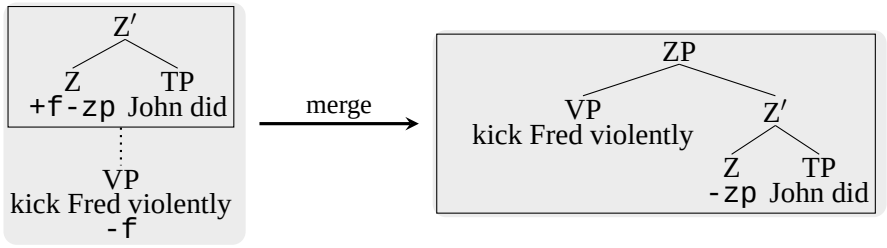
(61) a.



b.

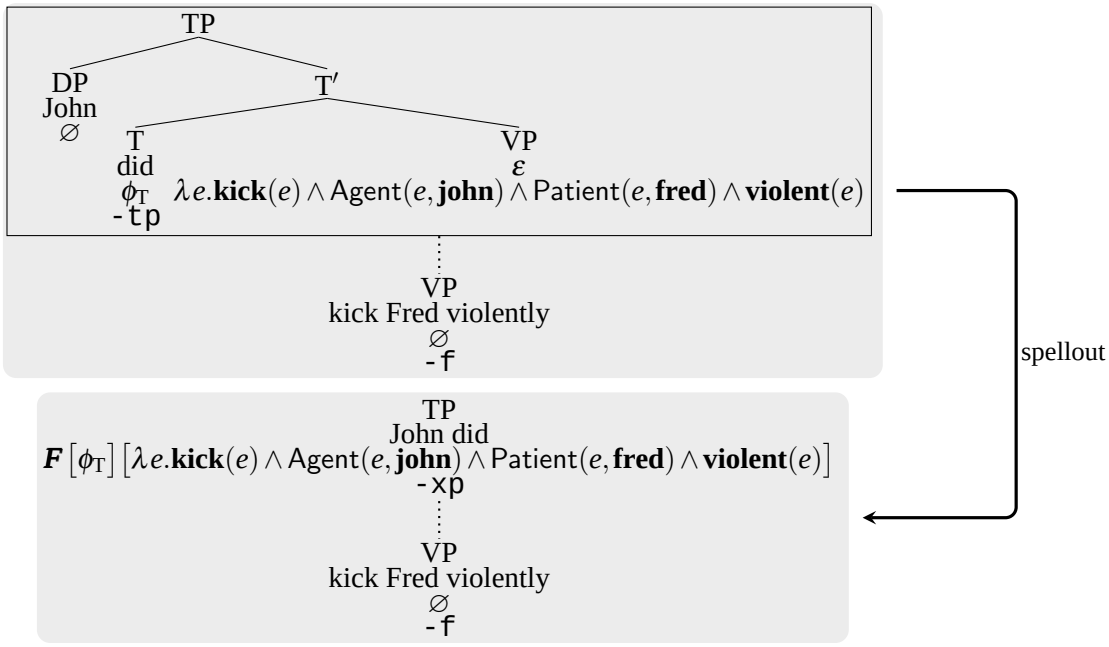


c.



Although I have not included semantic values in (61), note that while the VP's *phonological* realisation in its position as the complement of T is null, its *semantic* realisation here is exactly what it would be in a derivation where no fronting is taking place, since I take fronting to be semantically vacuous; recall the analogous effect of merging 'Fred' into the object position in (52). Thus in (61b) spellout composes the semantic value of the VP, itself computed as in (54), with that of the T head, as shown in more detail in (61′). (I also take the semantic realisation of the subject in its Case position to be null.) The disconnected 'kick Fred violently' piece is semantically null but phonologically non-null; it is the "other half" of the complement of T which is phonologically null but semantically non-null.

(61′)

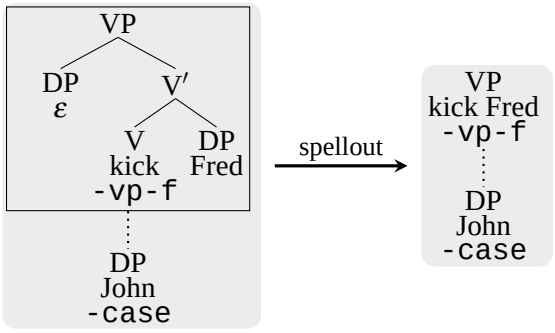


Notice also that since the `-f` feature in (61) originates on the lexical item 'kick', it ends up being carried by the VP unit produced when spellout composes this verb with its arguments. Since 'violently' is part of that unit, it is henceforth inseparable from 'kick Fred', and is included in the fragment of the sentence that is eventually fronted, producing (60a). The crucial idea behind my analysis of the alternative in (60b), to which I now turn, is that there is a way for 'violently' to semantically modify 'kick Fred' just as it does in (61), *without* being part of

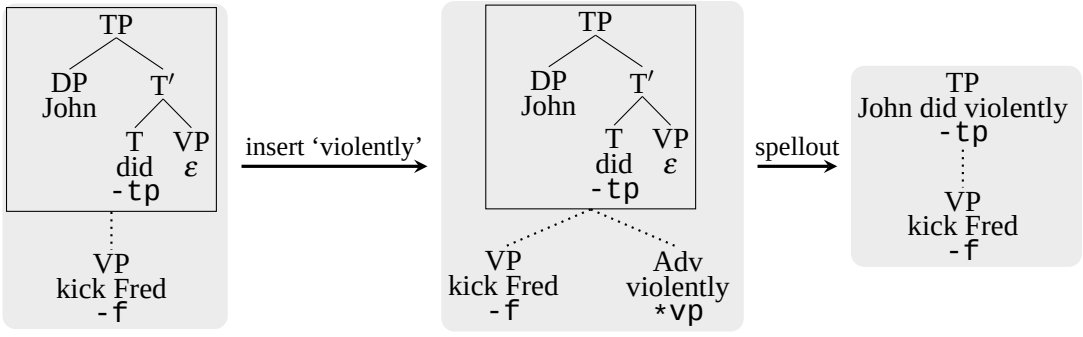
the unit produced when spellout composes the verb with its arguments, and therefore without being part of the fronted constituent.

Recall the intuition underlying the proposal that adjuncts are only inserted, and not merged: an adverb such as 'violently' is "more easily accommodated" than an object such as 'Fred' is, because its phonological and semantic composition does not require any disambiguating help (e.g. identifying thematic relations) of the sort that syntactic configurations established by merge provide. Adjunction features such as `*vp` encode only this ability to compose "directly", as 'violently' does with a verb, in contrast to composing "indirectly", as 'Fred' must with a verb. One way, naturally, to compose directly with the verb 'kick' is to be inserted during the construction of the VP it heads, as 'violently' was in (61). But there is another way to achieve the same result: 'violently' can be inserted during the construction of the TP, in which the to-be-modified VP is, crucially, also participating (namely, as a complement). This is how we derive the alternative fronting possibility in (60b) where the adjunct is "stranded". The relevant parts of this derivation are shown in (62); compare with (61), which produced (60a).

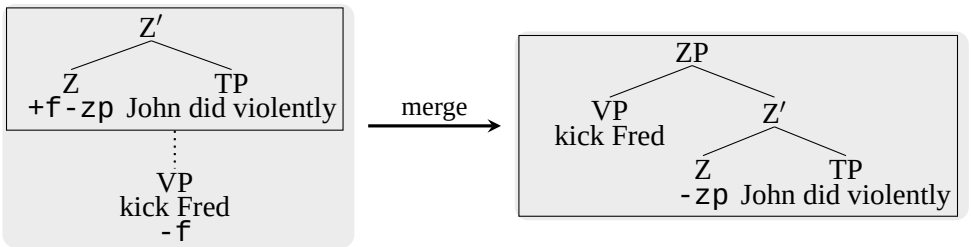
(62) a.



b.



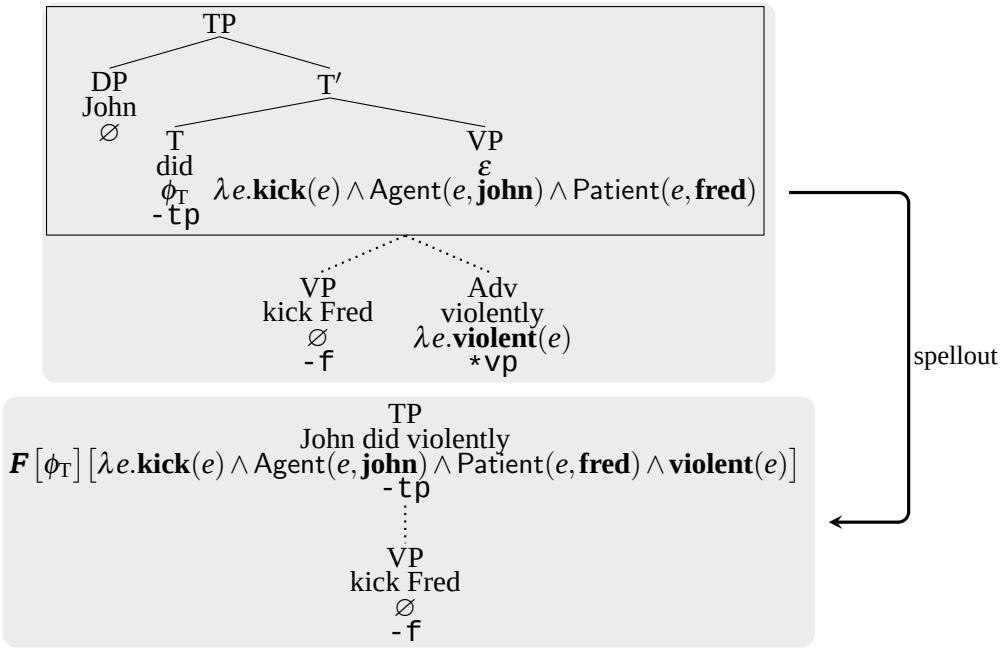
c.



The distinctive part of this derivation is that 'violently' is inserted in (62b), during construction of the TP, rather than during construction of the VP as before. As a result, the disconnected piece waiting to be fronted (and have its `-f` feature checked) is only 'kick Fred', *without* the adjunct. As observed earlier, it follows from the implementation of movement I have adopted that when the verb enters the derivation with a `-f` indicating that fronting will occur, the fragment that will be left disconnected and subsequently fronted is precisely the unit produced by the application of spellout that composes the verb with its arguments. Since 'violently' can be inserted either during construction of the VP (as in (61a)) or during construction of the TP (as in (62b)), it can be either included in or excluded from the fronted fragment.

Why is it independently plausible that 'violently' (but not 'Fred') has both of these options open to it? The reason is that because the composition of 'violently' with (the rest of) the VP does not rely on any specifics of the *internal structure* of the VP, this composition can take place at any application of spellout where this VP is "visible". The application of spellout in (62b) that flattens the TP is one such candidate. The finer details of this specific step are shown in (62′); compare with (61′).

(62′)



The way in which 'violently' is semantically composed in (62′) is not all that different from what was illustrated earlier in (54) (and assumed but left implicit in (61a)), even though now it is occurring at a point where a TP projection is being flattened rather than a VP projection. To repeat, the semantic destiny of an adjunct like 'violently' is to conjoin, unencumbered by such

complications as thematic relations, with (a projection of) a verb. In (54)/(61a), $\lambda e.\textbf{violent}(e)$ was conjoined with $\lambda e.\textbf{kick}(e)$ as a result of its $*\mathsf{vp}$ feature "matching" the V which headed the maximal projection under construction. Here in (62′), $\lambda e.\textbf{violent}(e)$ is conjoined with the VP meaning which is the complement of the maximal projection under construction. These two possibilities can be represented abstractly as in (61″) and (62″). In the first case, (61″), composition of ϕ_{TP} is straightforward (whatever the details of '$\boldsymbol{F}\left[\ldots\right]\left[\ldots\right]$' are). The relevant ϕ_{VP} happens to already include $\lambda e.\textbf{violent}(e)$, but this isn't of any particular consequence for how ϕ_{T} composes with ϕ_{VP}. The second case, in (62″), is different. Here ϕ_{VP}, which was computed in (62a), does not include $\lambda e.\textbf{violent}(e)$. In this case the effect of the spellout step shown in (62′) is to compose ϕ_{T} with a complement *which is itself now modified by* $\lambda e.\textbf{violent}(e)$.

(61″)
$$\phi_{\text{VP}} = \lambda e.\textbf{kick}(e) \wedge \mathsf{Agent}(e, \textbf{john}) \wedge \mathsf{Patient}(e, \textbf{fred}) \wedge \textbf{violent}(e)$$
$$\begin{aligned}\phi_{\text{TP}} &= \boldsymbol{F}\left[\phi_{\text{T}}\right]\left[\phi_{\text{VP}}\right] \\ &= \boldsymbol{F}\left[\phi_{\text{T}}\right]\left[\lambda e.\textbf{kick}(e) \wedge \mathsf{Agent}(e, \textbf{john}) \wedge \mathsf{Patient}(e, \textbf{fred}) \wedge \textbf{violent}(e)\right]\end{aligned}$$

(62″)
$$\phi_{\text{VP}} = \lambda e.\textbf{kick}(e) \wedge \mathsf{Agent}(e, \textbf{john}) \wedge \mathsf{Patient}(e, \textbf{fred})$$
$$\begin{aligned}\phi_{\text{TP}} &= \boldsymbol{F}\left[\phi_{\text{T}}\right]\left[\lambda e.\phi_{\text{VP}}(e) \wedge \textbf{violent}(e)\right] \\ &= \boldsymbol{F}\left[\phi_{\text{T}}\right]\left[\lambda e.\textbf{kick}(e) \wedge \mathsf{Agent}(e, \textbf{john}) \wedge \mathsf{Patient}(e, \textbf{fred}) \wedge \textbf{violent}(e)\right]\end{aligned}$$

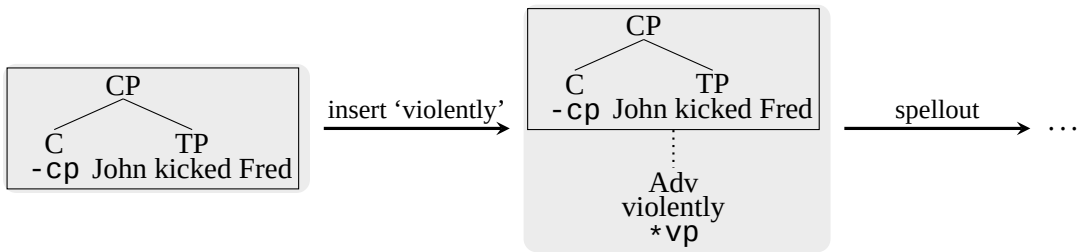
Two crucial points cannot be overemphasised. The first is that there is *no* stipulation here along the lines of: "Elements that are merely inserted, and remain disconnected until interpreted by spellout, can be included in or excluded from a moved constituent, whereas elements that are inserted and then merged must be included in the moved constituent." This would be just a restatement of the generalisation noted in §2.1. The desired pattern emerges, without any such stipulation, from the interaction of (i) the implementation of movement I have adopted, from which it follows that the unit manipulated by VP-fronting is precisely the unit produced when the verb is composed with its arguments, and (ii) the implementation of adjunction I have proposed, from which it follows that an adverb like 'violently' can be semantically composed either when a verb composes with its arguments or when some other head composes with the entire VP. The proposed implementation of adjunction makes use of the very same "dotted-

line structure" that arises from the conception of movement I adopt, and so once we allow an adjunct like 'violently' to be inserted either before or after spellout composes the verb with its arguments, the way it will interact with VP-fronting is determined.

The second crucial point is that I am not proposing the following: 'violently' can modify either the VP or the TP, and when it is included in the fronted fragment it is modifying the VP, and when it is not included in the fronted fragment it is modifying the TP. I am proposing that the non-fronting sentence 'John kicked Fred violently' can be derived in two different, but semantically equivalent, ways. The final values for ϕ_{TP} are identical in (61″) and (62″): in both cases, 'violently' modifies the VP. In an alternative where an adjunct modified the TP, this adjunct would bear a `*tp` feature and ϕ_{TP} would be of the form $\phi_{\text{TP}} = \boldsymbol{F}\left[\phi_{\text{T}}\right]\left[\phi_{\text{VP}}\right] \wedge \phi_{\text{adjunct}}$. This is not what happens to the adjunct in (62″). There, its semantic value occurs directly conjoined with ϕ_{VP}, *inside* '$\boldsymbol{F}\left[\ldots\right]\left[\ldots\right]$'; the crucial idea is simply that it is placed there at the application of spellout that interprets the structure of the TP, in (62b). Neither the internal structure of the VP, to which the application of spellout in (62a) has sole access, nor the internal structure of the TP, to which the application of spellout in (62b) has sole access, is of any importance for the composition of 'violently' with the VP it modifies. Notice that this account does not rely on any specific properties of the TP projection, or of the relationship between TP and VP: whatever the next projection above VP is, the same account will apply.

The two applications of spellout just mentioned, however, do have the important property that they both involve, in one way or another, the VP. Therefore they are both candidates for performing the composition of 'violently' with its target. Not just *any* later application of spellout will do. Consider the "next projection up", where TP is merged as a complement of a C head. Suppose that 'violently' is not inserted during construction of the VP, nor during construction of the TP, and we try inserting it during construction of this CP. This is illustrated (for simplicity, in a case where no fronting is involved) in (63).

(63)



In the structure to which spellout applies here, the VP which 'violently' is to modify is not visible. It is a part of the TP which is visible, of course, but the internal structure of the TP is not visible here and neither, therefore, is the VP which is its complement. In this respect, the application of spellout here that flattens the CP is crucially different from the other two discussed above.[20]

Crucially, the flexibility discussed here for the attachment of 'violently' is not available for arguments. Therefore there are *not* two derivations of 'John kicked Fred', one of which makes available 'kick Fred' for fronting and one of which makes available 'kick' alone. This is a consequence of the fact that 'Fred' requires a more specific syntactic configuration than 'violently' does: in order for its precise interpretation to be properly identifiable, 'Fred' must establish itself as a complement of 'kick'. In other words, it must interact with 'kick' in a certain way that is *only* possible during construction of the VP, not during construction of the TP. For comparison with (61″) and (62″): there is no possibility of adding a *thematic* conjunct to ϕ_{VP} at the point where spellout applies to flatten the TP as shown in (64), because the relevant structural relations are not visible at that point (i.e. the choice for R would not be defined). Therefore 'Fred', unlike 'violently', has no option of "later insertion", and is necessarily included in the fronted fragment.

(64) $\phi_{\mathrm{TP}} = \boldsymbol{F}\left[\phi_{\mathrm{T}}\right]\left[\lambda e.\phi_{\mathrm{VP}}(e) \wedge R(e, \mathbf{fred})\right]$ (not possible)

6.2 Counter-cyclic attachment

The key idea in the previous subsection was essentially that in order to modify a certain XP, an adjunct can be inserted during the construction of that XP, *or* during the construction of a higher projection into which that XP has merged; the latter permits the "stranding" of adjuncts in fronting constructions, for example. Since in the framework I am proposing there is no formal distinction between (first) merge and re-merge, the very same mechanisms will also allow an adjunct to modify a certain XP by being inserted during the construction of a higher projection into which that XP has *re*-merged. This possibility which has naturally emerged produces exactly the effects of Lebeaux's (1988) "counter-cyclic adjunction" analysis of the

[20] There is nothing preventing spellout from *applying* here, producing a flattened CP, but 'violently' will not be seen as relevant to the semantic or phonological composition performed when this happens.

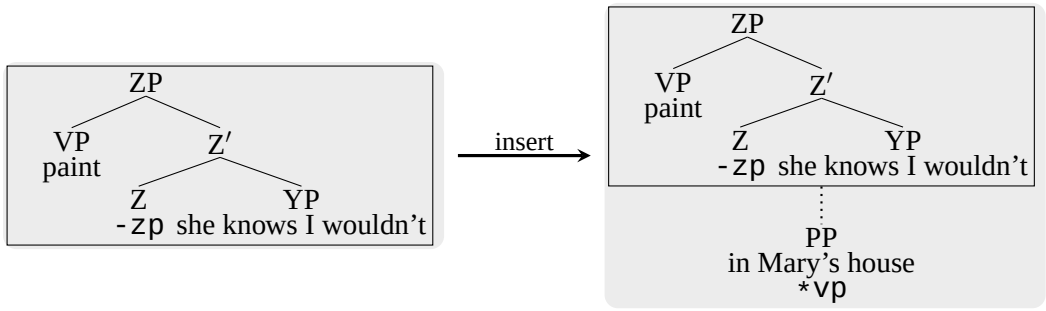
contrasts in (8) and (9) from §2.2. I will illustrate with the VP-fronting cases in (9), repeated here, since these are more directly analogous to the derivations in the previous subsection.

(9) a. * Paint [pictures of Mary$_i$], she$_i$ knows I wouldn't

b. Paint [in Mary$_i$'s house], she$_i$ knows I wouldn't

Consider a derivation that builds a VP consisting only of the intransitive 'paint', merges this as appropriate in the embedded clause, and then re-merges it into the fronted position (say, specifier of ZP, maintaining terminology from above), as if the sentence to be derived were simply 'Paint, she knows I wouldn't'. At this point, crucially, the VP 'paint' is visible, in the relevant sense: it is visible as the specifier of the ZP under construction, just as the VP was visible as the complement of the TP under construction in (62b). Then inserting the adverbial phrase 'at Mary's party' during construction of the ZP will yield modification of 'paint', as desired, just as inserting 'violently' during construction of the TP permitted modification of VP; see (65).

(65)



When spellout applies, 'in Mary's house' will be attached to 'paint' to produce (9b). Just as in Lebeaux's proposal, 'Mary' is never c-commanded by 'she', and there is no Condition C effect. The object 'pictures of Mary' in (9a) does not have the flexibility to be introduced during construction of the ZP like this, just as the object 'Fred' did not have the flexibility to be introduced during construction of the TP as 'violently' was in (62).

The classic wh-fronting examples in (8) can be treated similarly, with one additional assumption. Since the fronted phrase is a DP 'which claim', it is this phrase which is visible — and hence eligible for modification — in the way that the VP 'paint' is in (65), and so I must suppose that the relative clause attaches at the DP level. This is an unusual assumption, since the most obvious approach to the relevant semantics would suppose that the relative clause is conjoined with the NP before the D head combines with the resulting complex predicate.

One way to resolve this tension is to suppose that the quantificational semantics of the question ('for which *x*, . . . ') is not contributed by the D head 'which', but rather is a consequence of the DP appearing in a certain syntactic position. The idea is that in much the same way that an argument is interpreted as a patient/theme in virtue of appearing in the complement-of-VP position, a DP (which denotes simply a property) is interpreted as the restrictor of a question-forming operator in virtue of appearing in the specifier-of-CP position. Put differently: just as (50) illustrates the composition of an event predicate from 'Fred'-as-patient and 'John'-as-agent, an analogous rule will compose a question at the CP level from one property marked as restrictor and another property marked as nuclear scope. This approach has some similarities with that of Cable (2008), who likewise dissociates question semantics from wh-words. The major difference is that for Cable it is a functional "Q-particle" that takes up the slack, rather than a structural *position* itself; this contrast mirrors the choice between whether to assume that thematic relations are assigned by functional heads (e.g. Borer, 2005) or on the basis of structural configurations (as I have assumed, following e.g. Carlson (1984) and Pietroski (2005)). Details remain to be worked out, but I leave this for future work.[21]

The fact that we are forced to this conclusion about the position of relative clauses might lead one to think that the current system, with its requirement that counter-cyclic adjunction can only target the root of the re-merged constituent, is on the wrong track. But there is evidence that this restriction on counter-cyclic attachment is in fact correct: Landau (2007) notes that an adjunct to the object of a fronted VP, as in (66a), does not escape a Condition C violation in the way that an adjunct at the root of the fronted constituent itself does in (66b) and (66c).

(66) a. * Eat food that Mary$_i$ cooks, she$_i$ knows I never would

b. Food that Mary$_i$ cooks, she$_i$ knows I would never eat

c. Eat food at Mary$_i$'s party, she$_i$ knows I never would

The same pattern can be observed in the wh-fronting cases that Lebeaux discussed. The Condition C effect cannot be avoided in (67a), despite the fact that 'Mary' is contained within an adjunct ('at Mary's party'). This follows if the only adjuncts that can attach after wh-movement

[21] Bach & Cooper (1978) also attach relative clauses the DP level, but with different semantic justification.

are those that modify the fronted wh-phrase itself, such as the relative clause in (67b).

(67) a. * Which argument that John danced at Mary's party was she willing to discuss?

b. Which argument that surprised Mary at John's trial was she willing to discuss?

An account that allowed late adjunction anywhere inside a fronted constituent, such as Lebeaux's, would fail to predict Condition C violations in (66a) and (67a). This provides further reason to entertain the hypothesis that relative clauses attach at the DP level, rather than necessarily maintaining the assumption that they attach inside the fronted DP at the NP level: even in the absence of any theory of how late adjunction comes about, this hypothesis would be required in order to bring (66) and (67) together under a single descriptive generalisation.[22]

To restate the general idea: an adjunct that modifies XP can be inserted either (i) during construction of XP itself (i.e. where the X head combines with its arguments), or (ii) during the construction of any other phrase where XP is present, either because XP has (first-)merged there as in (62b) or because XP has re-merged there as in (65). The option of inserting the adjunct where XP is first-merged produces the "stranding" of adjuncts (in cases where XP moves away from this position), and the option of inserting the adjunct where XP is re-merged produces counter-cyclic adjunction of the sort proposed by Lebeaux, but with further empirically supported predictions about restrictions on this counter-cyclic adjunction.

6.3 Adjunct islands and freezing effects

I take the examples in (10) as representatives of licit extraction from an argument and illicit extraction from an adjunct. This contrast also receives a neat analysis in the proposed framework.

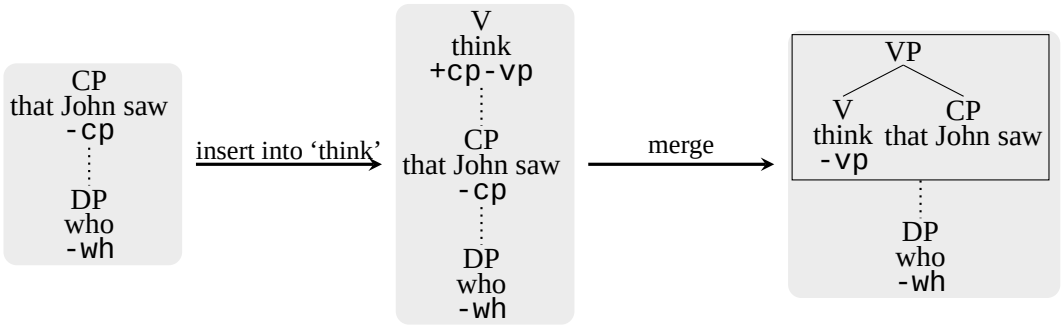
(10) a. Who do you think [that John saw ____]?

b. * Who do you worry [because John saw ____]?

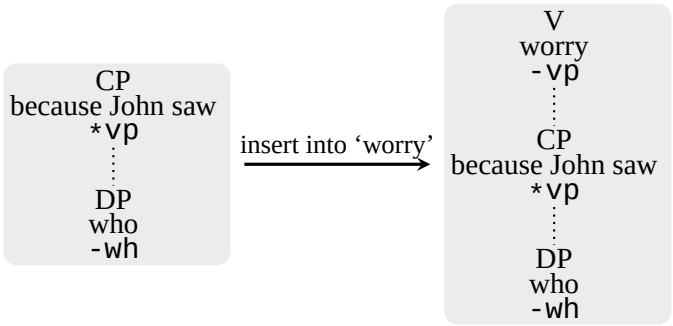
The crucial difference between the two sentences is in the construction of their respective matrix VPs. The relevant derivational steps are shown in (68).

[22] Although another alternative would be to suppose that attachment to the relative clause in (67b) is not *at* the root, but rather just *sufficiently close to* the root; perhaps making use of the notion of extended projections.

(68) a.



b.



The two structures that these partial derivations begin with represent the subconstituents bracketed in (10). In (68a), we have a complement clause ‘that John saw’ with a disconnected ‘who’. In (68b), we have a VP-adjunct ‘because John saw’, likewise with a disconnected ‘who’. Each is inserted into a structure headed by the matrix verb: ‘think’, which requires a CP complement, in (68a), and ‘worry’, with no such requirement, in (68b).

After this first step the two cases crucially diverge. For (68a) a merge step is required, to check the `+cp` and `-cp` features, establishing the embedded clause as an argument of ‘think’. For (68b), however, no such merge step is required: the adjunct is ready to be interpreted as part of the VP having been only inserted. This distinction is the hook on which the prohibition on extraction from adjuncts will hang. In order to encode the adjunct island constraint, we need the possibility of extracting ‘who’ to be contingent upon the merging of its parent node (here, ‘that John saw’) into an argument position. We can do this by stipulating that spellout does not apply to structures where there are disconnected pieces that are not directly connected to the root; in terms of the diagrams used here, spellout does not apply to structures with more than two “layers”. The derivation in (68a) can therefore continue, with an application of spellout that will construct a new VP unit ‘think that John saw’. But the derivation in (68b) cannot, because we are left with a structure which is too deep: it has three layers, such that ‘who’ is not directly connected to the root. Therefore we can derive (10a), but not (10b).

I should emphasise that I have not said anything insightful so far about the nature of adjunct island effects. This constraint on the structures to which spellout can apply is simply a restatement of the fact that extraction from adjuncts is prohibited. The attraction of it is that in combination with the implementation of movement I have adopted, the very same constraint simultaneously prohibits extraction from moved constituents.

The general idea is as follows. The adjunct island constraint amounts to a prohibition on applying spellout to a structure in which a disconnected piece α, waiting to re-merge, is the daughter of (or is "moving out of") another piece β, which is an adjunct and therefore *is itself* disconnected from the root node. But the general constraint also prohibits applying spellout to configurations in which a disconnected piece α is the daughter of another piece β which *has not yet reached its final position*, and is thus also disconnected from the root node, as it would be if it were an adjunct. This second kind of configuration is exactly the one that characterises freezing effects. Therefore the crucial distinction we have just observed in (68) will be the same as the one underlying the contrast illustrated in (12), repeated here, between extraction from an in-situ DP and extraction from a moved DP.

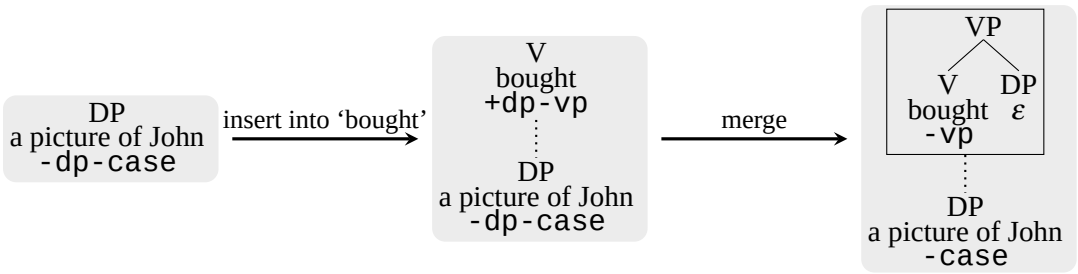
(12) a. Who did you buy [a picture of ____]?

b. * Who was [a picture of ____] bought (by you)?

First, to fix relevant background assumptions, let us consider the derivation of a passivised sentence without wh-movement. The relevant part of the derivation of (69) is shown in (70).

(69) A picture of John was bought

(70)

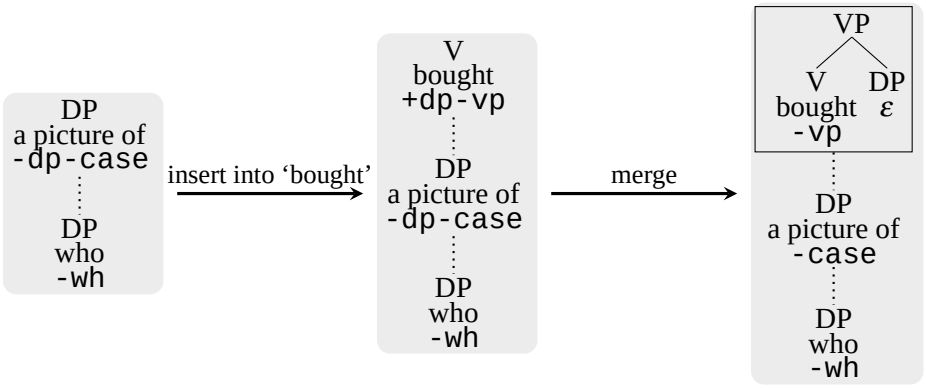


The steps in (70) begin at the point where the (surface) subject 'a picture of John' has been fully constructed. Besides its `-dp` feature, this unit bears an additional `-case` feature which will drive its subsequent movement to the surface subject position. Because this feature remains unchecked, we are left with a disconnected piece 'a picture of John' after the merge step in

(70). This causes no problem in the derivation of (69), where no movement out of the subject is required: spellout can apply to the last structure shown in (70).

Now consider (71), showing the attempted derivation of (12b).

(71)



Here the subject itself has a disconnected piece 'who' waiting to re-merge. After the two steps shown here the `+dp` requirement of 'bought' has been satisfied, and there is nothing further to be done to complete the construction of the VP; for the derivation to continue, spellout should now apply. However, as in (70), 'a picture of' remains disconnected from the root of the structure, and 'who' itself remains disconnected from 'a picture of'. Therefore we find ourselves again with a "three-layer" structure, so the constraint introduced above prohibits application of spellout.

To see the similarity between adjunct island configurations and freezing configurations, note the similarity between the attempted extraction from an adjunct in (68b), and the attempted extraction from a moving subject in (71). In each case the constituent out of which 'who' needs to move remains disconnected from the root — in the first case, because 'because John saw' is an adjunct, and in the second case, because 'a picture of' has not reached its final position. Contrast these with the successful extraction from a (non-moving) complement in (68a), where the constituent 'that John saw' out of which 'who' needs to move does not remain disconnected from the root. The proposed constraint therefore makes a natural cut between (i) adjoined constituents and moving constituents, out of which movement is disallowed, and (ii) non-moving argument constituents, out of which movement is allowed.[23]

[23]This constraint also restricts remnant-movement configurations. Hunter (2012) explores the consequences of this restriction. One instance of remnant movement that *is* allowed is the VP-fronting derived in (61).

7. Conclusion

I have proposed a novel account of the syntactic phenomena of movement and adjunction, and the relationship between them. Two independently-appealing ideas that provided points of departure were a syntactic framework in which merging and moving are unified into a single operation, and a distinctive, restrictive view of semantic composition which forces us to assign interpretive significance to certain structural relations.

The modified view of merge and move operations leaves room for the possibility of a constituent being only loosely associated with a phrase: in particular, since the unification of merge and move required an insertion operation that does not check features, it is possible to hypothesise that to be adjoined is to be (only) inserted. As a result adjuncts have a degree of syntactic freedom that arguments lack, from which their otherwise puzzling abilities to be "either inside or outside" a phrase and to combine counter-cyclically emerge naturally.[24] A further consequence that emerges is that adjuncts and moving constituents have, in a certain sense, a common status; the proposal is therefore strengthened by any evidence that these two classes of constituents behave alike in certain contexts. I argued that existing generalisations concerning conditions on extraction domains provide exactly this sort of evidence: extraction out of both moved constituents and adjoined constituents is generally disallowed, a combination of otherwise unrelated facts that can be unified as one in the framework I have presented.

References

Adger, D. 2003. *Core Syntax: A Minimalist Approach*. Oxford: Oxford University Press.

Bach, E. & R. Cooper. 1978. The NP-S analysis of relative clauses and compositional semantics. *Linguistics and Philosophy* 2:145–150.

Baker, M. 1988. *Incorporation: A Theory of Grammatical Function Changing*. Chicago: University of Chicago Press.

Barker, C. & P. Jacobson, eds. 2007. *Direct Compositionality*. Oxford: Oxford University Press.

Boeckx, C. 2003. *Islands and Chains*. Amsterdam: John Benjamins.

[24] Hunter & Frank (to appear) build on this part of the proposal to provide an account of adjunct extraposition.

Borer, H. 2005. *The Normal Course of Events*. Oxford: Oxford University Press.

Cable, S. 2008. Q-Particles and the Nature of Wh-Fronting. In *Quantification: A Cross-Linguistic Perspective*, ed. L. Matthewson. Bingley, UK: Emerald.

Carlson, G. 1984. Thematic roles and their role in semantic interpretation. *Linguistics* 22:259–279.

Cattell, R. 1976. Constraints on movement rules. *Language* 52:18–50.

Chametzky, R. A. 1996. *A Theory of Phrase Markers and the Extended Base*. Albany, NY: State University of New York Press.

Chametzky, R. A. 2000. *Phrase Structure: From GB to Minimalism*. Malden, MA: Blackwell.

Chomsky, N. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.

Chomsky, N. 2001. Derivation by phase. In *Ken Hale: A Life in language*, ed. M. J. Kenstowicz. Cambridge, MA: MIT Press.

Chomsky, N. 2004. Beyond explanatory adequacy. In *Structures and Beyond*, ed. A. Belletti. Oxford: Oxford University Press.

Cinque, G. 1999. *Adverbs and Functional Heads: A Cross-Linguistic Perspective*. New York: Oxford University Press.

Collins, C. 2005. A smuggling approach to the passive in English. *Syntax* 8:81–120.

Cooper, R. 1983. *Quantification and Syntactic Theory*. Dordrecht: Reidel.

Corver, N. 2005. Freezing effects. In *The Blackwell Companion to Syntax*, eds. M. Everaert & H. van Riemsdijk, 383–406. Malden, MA: Wiley-Blackwell.

Dowty, D. 1991. Thematic proto-roles and argument selection. *Language* 67:547–619.

Epstein, S. D., E. Groat, R. Kawashima & H. Kitahara. 1998. *A Derivational Approach to Syntactic Relations*. Oxford: Oxford University Press.

Freidin, R. 1986. Fundamental issues in the theory of binding. In *Studies in the acquisition of anaphora*, ed. B. Lust, vol. 1, 151–188. Dordrecht: Reidel.

Frey, W. & H.-M. Gärtner. 2002. On the treatment of scrambling and adjunction in minimalist grammars. In *Proceedings of Formal Grammar 2002*, eds. G. Jäger, P. Monachesi, G. Penn & S. Wintner, 41–52.

Gärtner, H.-M. & J. Michaelis. 2010. On the Treatment of Multiple-Wh Interrogatives in Min-

imalist Grammars. In *Language and Logos*, eds. T. Hanneforth & G. Fanselow, 339–366. Berlin: Akademie Verlag.

Hornstein, N. 2001. *Move! A minimalist theory of construal*. Oxford: Blackwell.

Hornstein, N. 2009. *A Theory of Syntax: Minimal Operations and Universal Grammar*. New York: Cambridge University Press.

Hornstein, N. & J. Nunes. 2008. Adjunction, labeling, and bare phrase structure. *Biolinguistics* 2:57–86.

Huang, C. T. J. 1982. Logical relations in chinese and the theory of grammar. Ph.D. thesis, MIT.

Hunter, T. 2010. Relating movement and adjunction in syntax and semantics. Ph.D. thesis, University of Maryland.

Hunter, T. 2011a. Insertion Minimalist Grammars: Eliminating redundancies between merge and move. In *The Mathematics of Language (MOL 12 Proceedings)*, eds. M. Kanazawa, A. Kornai, M. Kracht & H. Seki, vol. 6878 of *LNCS*, 90–107. Berlin Heidelberg: Springer.

Hunter, T. 2011b. *Syntactic Consequences of Conjunctivist Semantics: Unifying Movement and Adjunction*. Amsterdam: John Benjamins.

Hunter, T. 2012. A Constraint on Remnant Movement. In *Towards a Biolinguistic Understanding of Grammar: Essays on Interfaces*, ed. A. M. di Sciullo, 31–56. Amsterdam: John Benjamins.

Hunter, T. & R. Frank. to appear. Eliminating Rightward Movement: Extraposition as flexible linearization of adjuncts. *Linguistic Inquiry* .

Joshi, A. K. 1987. An Introduction to Tree Adjoining Grammars. In *Mathematics of Language*, ed. A. Manaster-Ramer, 87–115. Amsterdam: John Benjamins.

Kayne, R. 1994. *The antisymmetry of syntax*. Cambridge, MA: MIT Press.

Kobele, G. M. 2006. Generating copies: An investigation into structural identity in language and grammar. Ph.D. thesis, UCLA.

Landau, I. 2007. Constraints on Partial VP-fronting. *Syntax* 10:127–164.

Lebeaux, D. 1988. Language acquisition and the form of the grammar. Ph.D. thesis, University of Massachussetts, Amherst.

Michaelis, J. 2001. Derivational minimalism is mildly context-sensitive. In *Logical Aspects of Computational Linguistics*, ed. M. Moortgat, vol. 2014 of *LNCS*, 179–198. Berlin Heidelberg: Springer.

Müller, G. 2010. On deriving CED effects from the PIC. *Linguistic Inquiry* 41:35–82.

Parsons, T. 1990. *Events in the semantics of English*. Cambridge, MA: MIT Press.

Pietroski, P. M. 2005. *Events and Semantic Architecture*. Oxford: Oxford University Press.

Rizzi, L. 1990. *Relativized Minimality*. Cambridge, MA: MIT Press.

Schein, B. 1993. *Plurals and Events*. Cambridge, MA: MIT Press.

Schein, B. 2002. Events and the semantic content of thematic relations. In *Logical Form and Language*, eds. G. Preyer & G. Peter, 263–344. Oxford: Oxford University Press.

Speas, M. 1986. Adjunctions and projections in syntax. Ph.D. thesis, MIT.

Stabler, E. P. 1997. Derivational minimalism. In *Logical Aspects of Computational Linguistics*, ed. C. Retoré, vol. 1328 of *LNCS*, 68–95. Berlin Heidelberg: Springer.

Stabler, E. P. 2001. Recognizing head movement. In *Logical Aspects of Computational Linguistics*, eds. P. de Groote, G. Morrill & C. Retoré, vol. 2099 of *LNCS*, 254–260. Berlin Heidelberg: Springer.

Stabler, E. P. 2006. Sidewards without copying. In *Proceedings of The 11th Conference on Formal Grammar*, ed. S. Wintner, 157–170. Stanford, CA: CSLI Publications.

Stabler, E. P. 2011. Computational perspectives on minimalism. In *The Oxford Handbook of Linguistic Minimalism*, ed. C. Boeckx. Oxford: Oxford University Press.

Steedman, M. 2000. *The Syntactic Process*. Cambridge, MA: MIT Press.

Stepanov, A. 2001. Cyclic domains in syntactic theory. Ph.D. thesis, University of Connecticut.

Truswell, R. 2007. Extraction from adjuncts and the structure of events. *Lingua* 117:1355–1377.

Uriagereka, J. 1999. Multiple spell-out. In *Working Minimalism*, eds. S. D. Epstein & N. Hornstein, 251–282. Cambridge, MA: MIT Press.

Wexler, K. & P. Culicover. 1981. *Formal Principles of Language Acquisition*. Cambridge, MA: MIT Press.

Williams, A. 2008. Patients in Igbo and Mandarin. In *Event structures in linguistic form and*

interpretation, eds. J. Doelling, T. Heyde-Zybatow & M. Schaefer, 3–30. Berlin: Mouton de Gruyter.