Unconventional Mergers

Mark de Vries
University of Groningen
– final draft, dec. 2011 –

Abstract. This article discusses the nature of the operation Merge, and the possibility of unconventional mergers, with some attention to the issue of representation. Regular Merge has three possible effects, depending on the input. Next to 'first-time' merge, there are both 'internal' and 'external' *re*merge. The last creates a temporary doubly-rooted structure, which potentially results in a configuration with superficially non-local properties. Furthermore, it is argued that a fundamentally different kind of operation is needed in order to account for the general behavior of parentheses: Parenthetical Merge, which has the effect of blocking scopal relationships, among other things. Finally, a family of construction types is studied in which both phenomena seem to come together: sentence amalgamation.

Keywords: Merge, remerge, multidominance, parenthesis, amalgams, c-command, non-locality

1. Introduction

An essential aspect of a generative grammar is its structure building capacity.[*] In the Minimalist Program, rewriting rules have been replaced with the simpler and more general operation called Merge. Merge combines syntactic objects into larger syntactic objects, and it can do so recursively, meaning that the output of Merge can be used as input again. This leads to a hierarchical structure, as we will discuss in detail.

Almost from the beginning, the exact definition and workings of Merge have been disputed, which is remarkable at first sight, since the operation was presented in Chomsky (1995) as a conceptual necessity, and one would perhaps not expect much discussion on such basic issues. Nevertheless, the generative core of grammar has been an important focus of attention since *Syntactic Structures* (Chomsky 1957), as it determines which kind of representations can be derived and which cannot. And this puts potential analyses of more concrete constructions and grammatical phenomena within certain boundaries. With respect to Merge, we can distinguish at least five issues of discussion:

(i) *Labels.* To which extent are the insights of X-bar theory to be incorporated in Merge? How 'bare' is phrase structure, and do we need labels to begin with? Are the categorial status and the projection status of phrases predictable? For relevant discussion, see Epstein, Thraínsson & Zwart (1996), Collins (2002), Di Sciullo & Isac (2008), and Boeckx (2008), among others.
(ii) *The environment.* Which boundary conditions play a role for Merge? How is the syntactic workspace organized? Is there a so-called numeration? Which objects are accessible as input for Merge, and where are the results stored? How many objects can be Merged at the

[*] I would like to thank the anonymous reviewers for useful comments and questions. I am also grateful to the Editors for kindly allowing me to submit a paper to this specialized volume. This research was financially supported by the Netherlands Organisation for Scientific Research (NWO).

same time? Is Merge invoked in cycles/phases? See also Bobaljik (1995), De Vries (2005a), and Chomsky (2005), for instance.

(iii) *The nature of the objects handled and created by Merge.* According to Chomsky, a phrase is a simple set (or superset if it contains other phrases). This leads to syntactic representations that involve hierarchy but not order. However, it has also been argued that Merge creates ordered pairs instead. This would mean that the relationship between sisters is basic as well, which in turn affects the discussion on linearization. See, e.g., Langendoen (2003), Zwart (2006), and Fortuny (2008).

(iv) *Movement.* Is movement simply Merge applied to the same element again (remerge)? Is it possible to move an item from one structure to another? What does that mean for the representation of syntactic structure? Do we need copies, chains, multidominance, or all of them? See Starke (2001), Gärtner (2002), Nunes (2004), De Vries (2009b), and the references therein.

(v) *Uniformity.* Is there just one kind of Merge, or could there be more? Chomsky himself introduced both set-Merge and pair-Merge (for adjuncts); see Chomsky (2004) for some discussion. Additionally, special devices may be needed for parentheses and the like; see also Ackema & Neeleman (2004) and De Vries (2007).

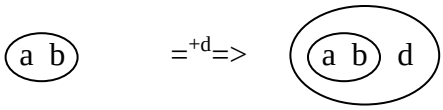
Needless to say, these issues are interrelated in several ways.

Merge, at any rate, is a tool fit for creating standard syntactic hierarchies. In this article, we will discuss the possibility of certain non-canonical mergers. Section 2 is about remerge. It is argued that remerge follows from standard boundary conditions (or rather the lack thereof) on the selection of input elements for Merge. Depending on the exact configuration, there are two options: *internal remerge*, which corresponds to regular movement, and *external remerge*, which corresponds to the idea of 'sideward movement' or 'sharing'. Section 3 is about Parenthetical Merge. This is a specialized operation necessary for the connection of parenthetical material with the matrix. Section 4 tentatively explores the consequences of the application of both external remerge and Parenthetical Merge within one sentence. It will turn out that this correctly predicts a number of remarkable properties of so-called *amalgams*: partial 'invisibility' and apparent non-locality.

2. Internal and external remerge

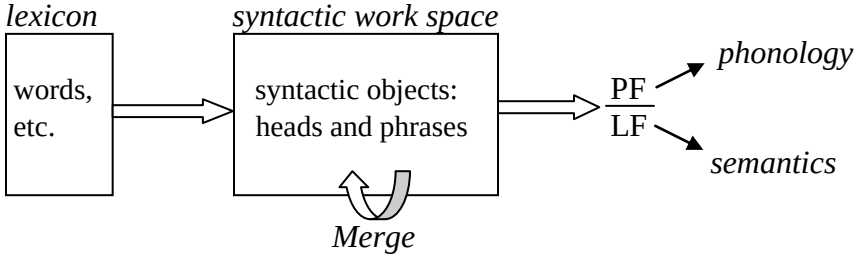
The operation Merge can be viewed as a (mathematical) function with input variables and one output, applied to a linguistic domain: Merge (a, b) → c, such that a and b are syntactic objects, and c equals the combination of a and b, in familiar bracket notation $[_c$ a b], which, by definition, is also a syntactic object. By definition, since it is not a logical necessity: generally, properties of a whole can be emergent, and not every property of the parts is necessarily preserved in the whole. For instance, the combination of two singulars leads to a plural, like connected atoms form a molecule. Merge, however, preserves syntactic objecthood. Of course, this is the default assumption, and it is intuitively plausible. Crucially, Merge does not tamper with the input. Thus, Merge ($[_c$ a b], d) results in $[_e\, [_c$ a b] d], but not in [a b d], which would in fact imply that the group c is changed or destroyed. Therefore, the iterative application of Merge automatically yields a hierarchy (1b) and not a flat concatenation (1a): a syntactic structure is not a word string.

(1) a. a—b $=^{+d}=>$ a—b—d

b. 

In other words, if a group is Merged with another syntactic object, the participants of the group do not play a role in the outcome: the original group stays intact. If only for practical purposes, groups deserve a label. In the above example, the inner and outer circle can be called c and e, respectively. An immediate question that arises is the following: are the elements a and b themselves still accessible for Merge once they have been merged together? The usual answer is positive, as we will discuss below. But notice right away that this leads to representational complications.

First, we need to separate the core properties of Merge from the boundary conditions.[1] Essentially, Merge relates syntactic objects to each other, and therefore it is structure-building: it combines the input objects into a group. As an immediate consequence, it forms a hierarchy: the original input objects are directly included in the output object. It is usually assumed that the number of input objects must be two, which leads to binary branching. Furthermore, both the input and output may be subject to certain constraints. This in particular is where theories converge. Consider the simple model in (2):

(2) 

I will assume that material from the lexicon can be transferred to the syntactic work space, which may then contain several independent syntactic objects. Merge operates on objects within this work space. If syntax is done, the resulting phrase structure is passed on to the interfaces. Possibly, syntax works in cycles/phases; this could mean that periodically, embedded material is pushed out of the syntactic work space and will be kept on stack. Much more can be said about all of this, but let us concentrate on what is happening inside the syntactic work space.

Each phrase, word, morpheme, or simply feature (bundle) transferred into the work space from the lexicon is recognized as a syntactic object, and can be used as input for Merge. Notice that it is also a structural root before merger, that is, the 'top node', often trivially so for simplex items. Now, two potential constraints on the input come to mind.

First, Merge could be restricted to roots by a simple and general rule. An object embedded inside a larger object (due to an earlier instance of Merge), a 'term', is no longer accessible. As a consequence, movement is impossible. Though such a limitation clearly has its advantages (mainly conceptual – see Koster 2007 for interesting comments), it is also clear that alternative

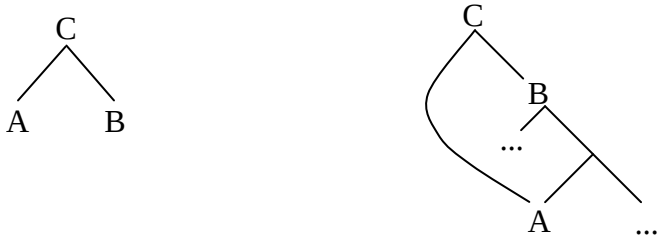
[1] In handbooks, these are sometimes treated on a par, and even combined into one complex definition. I think this is unfortunate, also from a didactical perspective.

explanations for (apparent) displacement phenomena then need to be found. Here, I prefer to adopt the standard view that movement, or rather, *remerge*, exists, and hence that terms of syntactic objects can be used as input for Merge again, at least within the same cycle.

Second, perhaps naively, it could be that the work space has only one 'center of attention', the root of the main structure. Each instance of Merge involves this root, and something else: any other root (normally, an unused lexical item) or a term to be moved. The root of the resulting structure is then the newborn center of attention. Although this has some initial plausibility, the strictest version of it must be discarded, considering that we need to be able to generate structures containing sisters that are both composed phrases. For example, the potential subject *the old man* and the predicate *ate a sandwich* are both phrases that must be derived before they can be combined into one clause. Therefore, more than one complex structure can be present in the work space, and in order to derive these and the combination thereof, the center of attention has to shift at least once from one structure to the other.[2]

Summarizing so far, Merge is sensibly restricted to syntactic objects figuring inside the syntactic work space, but at first sight there are no further pressing limitations on the input. Thus, an input object can be a root (simplex or complex) or a term of an existing phrase. Now, suppose that A and B are merged at a certain point of the derivation, yielding C. Depending on the initial status of A and B, three possible configurations ensue.[3] The first two are familiar; see (3):

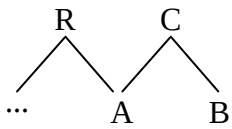
(3) a. *first-time merge* b. *internal remerge*



In (3a), A and B are unrelated roots before merger, each possibly complex (and hence the result of previous mergers). In (3b), A is originally a constituent of B, which implies that it has been input to merger before, in a previous step of the derivation. We can call this situation *internal remerge*, which corresponds with traditional movement. I will represent it using multidominance, thereby discarding artifacts such as traces or copies (cf. Gärtner 2002, among many others).

The third possibility, *external remerge*, is perhaps more surprising. It arises if a term is remerged with an external root. The result is depicted in (4):

(4) *external remerge*



[2] An alternative perspective in terms of layered derivations involving a complex numeration/resource is sketched in Zwart (2011).

[3] If the requirement is dropped that the center of attention is always a root (and hence that one of the input elements for Merge is a root), then two more possibilities follow, which can be called 'quirky' internal and external remerge; see De Vries (2009b) for discussion.

This structure can be derived by two simple sequences of Merge. Either Merge (..., A) → R is followed by Merge (A, B) → C, or the other way around. This process has also been called *grafting*, *parallel merge*, *sharing*, or (abstracting away from the issue of traces/copies/chains) *interarborial* or *sideward movement* (see, for example, Van Riemsdijk 2006a, Citko 2005, Guimarães 2004, Bobaljik & Brown 1997, Nunes 2001), and dates back to earlier ideas by Williams (1978), McCawley (1982), Goodall (1987), and others. According to most authors, the doubly-rooted structure in (4) can only exist as an intermediate stage in the derivation. Eventually, the two roots need to be connected. For further references, see De Vries (2009b), which article also discusses the linearization of structures involving multidominance – an important issue that I will ignore here for reasons of space.

A more formal characterization of the possible effects of Merge is provided in (5):

(5) Merge (α, β) → γ constitutes
   a. *first-time merge* iff α and β are independent roots before merger;
   b. *internal remerge* iff β is a root and α is included in β (or the other way around) before merger;
   c. *external remerge* iff β is included in some root δ, and α is an independent root (or the other way around) before merger.

Notice that Chomsky's original dichotomy 'external merge' versus 'internal merge' corresponds to (5a) and (5b).

At the cost of a complicated boundary condition on the output of Merge, we could exclude (5c) from the grammar, whilst maintaining (5b):

(6) *Exclude external remerge*
The output of Merge must be an independent syntactic object.

where *independence* can defined as follows:
A syntactic object α is independent iff α is not included in any other syntactic object, and no term of α is included in an object that is not also a term of α.

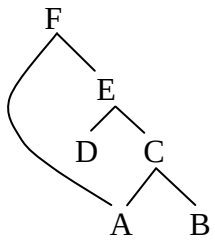
However, let us put off such a condition, in line with the authors cited, and investigate to which extent external remerge is a useful concept.

Before we go on, let me add two remarks. First, it is hard to decide which grammar is more minimal: one with a more limited expressive power, or one with less – or less complicated – rules or constraints. Ideally, the two criteria do not interact, or they even enforce each other, but as I just indicated, the situation is not (always) like this. Second, there is a tradition of defining syntactic structure in terms of tree representations (see Carnie 2008 for an overview and references). One could then call upon the simple 'single mother condition' in order to prevent the effect of external remerge: syntactic structures are trees and not graphs. However, from the present minimalist perspective, regular movement would also lead to a violation of the single mother condition – see (3b). Moreover, one needs to be careful in not confusing representations with the underlying theory. Consider the derivation of (7), where A is internally remerged (moved).

(7)

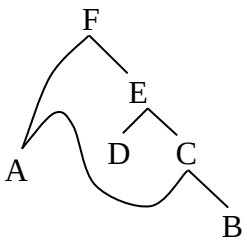
a.	b.
Merge (A, B) → C	*A is the merge-mate of B* *C directly includes A and B*
Merge (D, C) → E	*D is the merge-mate of C* *E directly includes D and C*
Merge (A, E) → F	A *is the merge-mate of E* *F directly includes A and E*

c.

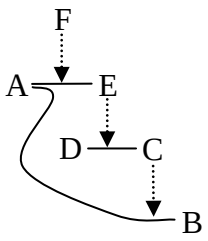


Merge does not create trees, it creates (a list of) strictly local relationships. Sisterhood correlates with being merge-mates; dominance correlates with inclusion. Non-local relationships all depend on the transitivity of dominance. Of course the whole point of merger is to relate syntactic objects to each other, which leads to semantic composition, and which facilitates all kinds of direct and indirect licensing. Since an object can be involved in more than one direct relationship, it makes sense to assume that it can be remerged. In example (7), A is the merge-mate of both B and E. A rather straightforward (and for most human beings more insightful) *representation* of the list of basic relationships created by Merge is the graph provided in (7c). As in any representation, there are some arbitrary choices. Lines represent inclusion directionally, for instance. Traditionally, the merge-mate relationship is not directly represented by some kind of symbol – but it can be inferred. Furthermore, the position of remerged elements is arbitrary. In this example, we could also picture A in its remerged position, or in fact anywhere else on the paper, as is illustrated in (8a). It is probably best to put it in the eventual spell-out position. In (8b), I suggest a different type of notation: here, the essential merge-mate relationship *is* directly represented, namely by solid lines between objects. The inclusion relationship is indicated by dotted arrows, from parental projections to the lines connecting a directly included pair of merge-mates.

(8) a.



b.



The representations in (7c) and (8) are intended as completely equivalent. Which one is to be preferred depends on personal preferences, traditions, and communicative purposes. The graphs in (7c) and (8a) highlight the multidominance effect of remerge. This might distract the attention from the more essential aspect of it, namely that some object, here A, is involved in two merge-mate relationships; the alternative in (8b) does not have this potential disadvantage.

In Chomsky's view (see also Hinzen 2006 and many others), the Merge operation is tantamount to set-formation. Though standard hierarchies can of course be represented by complex sets, the idea that sets are the foundation of syntax is problematic for several reasons. Importantly, there is no straightforward way of representing displacement by means of a complex set. It is for this reason that Chomsky proposed the copy theory of movement, which, as was already commemorated in the introduction, has been subject to fundamental critique – and rightly so in my opinion. But even apart from displacement, it is not obvious that sets alone could be the essence of syntax. After merger of A and B, there is a set/group {A, B}. This is to

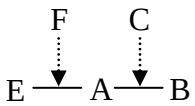
be interpreted as a syntactic constituent. From the notation {A, B} we may also want to infer that A and B are related to each other as syntactic sisters (with consequences for the interpretation, for syntactic licensing, etc.), but strictly speaking, the existence of a mathematically defined set of objects does neither imply nor facilitate any relationship between those very objects. If the idea is correct that upon merger syntactic objects can 'see' each other and interact with each other (and indirectly, each other's components) *because* they are sisters *because* they are merged together, then establishing a sisterhood *relationship* must be an essential part of what Merge does, next to just group formation. Notice, incidentally, that the notation {A, B} – though often referred to as a 'set' – is not essentially different from a tree structure $_{A}^{\wedge}{}_{B}$ or a bracketed structure [A B]: all are representations that can be interpreted as required, according to certain conventions in a certain application domain.

Another issue is the property of inclusion and projection. According to Chomsky, every group in syntax is a projection of one of the members. This does not follow in any way from a simple set, which is why he proposed that Merge creates a complex set, for instance {A, {A, B}} in case A projects. From this object, some kind of asymmetry between A and B can indeed be inferred, but which kind of asymmetrical relationship as well as its direction depend entirely on conventional interpretation.[4]

I conclude that what Merge does, minimally, is to create a group of associated sisters which defines a new, inclusive syntactic object. In other words, it establishes basic relationships between the input and output objects, as indicated in (7a/b), for instance.[5] Such relations ('direct inclusion', 'merge-matehood') can be considered theoretical primitives, for which we then need a convenient representation.

Let us now return to external remerge. An abstract, minimal example similar to (4) is given in (9):

(9)

a.	b.	c.
Merge (A, B) → C	*A is the merge-mate of B* *C directly includes A and B*	
Merge (E, A) → F	*E is the merge-mate of A* *F directly includes E and A*	

Again, A is involved in two mergers, this time with two independent roots, B and E. After Merge, the new roots are C and F, respectively.

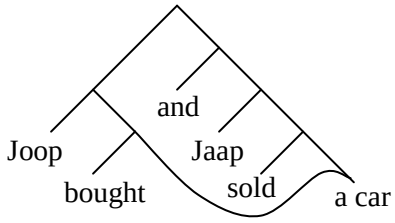
The effect of external remerge has been used to analyze head movement (Bobaljik & Brown 1997), parasitic gaps (Nunes 2004), across-the-board movement (Citko 2005), coordinated *wh* (Gracanin-Yuksek 2007), (transparent) free relatives (Van Riemsdijk 2006b), cleft and *wh*-amalgams (Guimarães 2004), and several other construction types. The most widely discussed is Right Node Raising (Wilder 2008, Chen-Main 2006, Kluck & De Vries to appear,

[4] I am glossing over further complicating issues related to the status of labels and the Wiener-Kuratowsky convention here. Independently of that, it is thinkable that there is more than one fundamental asymmetry, and that these do not coincide – which is also highly problematic for the set-only conception of Merge. Concrete candidates next to the asymmetry of inclusion and projection figuring in the literature are asymmetries between sisters in terms of syntactic dependency, selection, or simply precedence. See De Vries (2009b) for some discussion and further references. In the present article, I will not be concerned with possible asymmetries between sisters, and I will simply talk about merge-mates in order to avoid unnecessary intricacies.

[5] Compare also Koster's (2007) notion of a basic syntactic 'triad'.

among many others – based on earlier ideas by McCawley 1982). The basic idea is simple: in an example like *Joop bought _ and Jaap sold a car*, the verb phrases in the first and second conjunct share the direct object *a car*; thus, this noun phrase is merged as the complement of one verb, and then remerged with the other verb. The two clauses are completed, and finally joined by means of coordination. A schematic representation of the result is the following:

(10)



One interesting aspect of RNR is that it shows apparently non-local behavior. An example is (11), where the ellipsis site is inside a relative clause embedded in the first clausal conjunct:[6]

(11) [Joop talked to [a man who BOUGHT _]], and [Jaap talked to [a man who SOLD *a red car*]].

This possibility turns out to be an emergent property of applying external remerge, which can be used to create a bypass without actually violating locality constraints. For regular internal remerge, this is not the case. Put generally, (12a) can be derived unproblematically, but (12b) cannot.

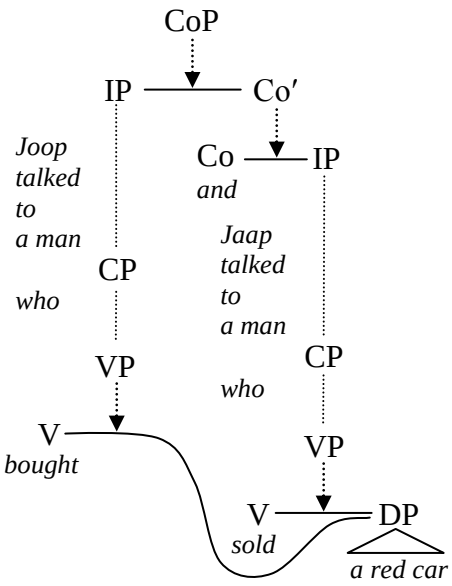
(12) a. ... [$_{CoP}$ [$_{XP}$ ... [$_{\varphi}$ ... α ...]] Co [$_{YP}$ ... [$_{\varphi}$ ... α ...]]] ...
b. * ... [$_{XP}$ ... [α [... [$_{\varphi}$ ... α ...]]]] ...

Here, α is the shared/moved phrase, φ is a locality barrier, and CoP in (12a) is the coordination phrase joining the complex parts XP and YP.

In a derivational grammar, locality rules can be implemented as restrictions on the input for Merge: material embedded in some locality barrier φ is no longer accessible for selection. This view is compatible with the idea of syntactic phases/cycles. As a consequence, α cannot be remerged with a projection dominating φ in (12b) (but note that the story changes if there is an available edge facilitating successive cyclic movement). In (12a), the situation is different because α can be remerged *before* φ is created. In the concrete example (11), the verb *bought* is merged with the phrase *a red car*. No locality boundary is reached yet, and the noun phrase can be externally remerged with the other verb, *sold*. This yields a simple doubly-rooted structure as in (9). Both temporary roots are expanded by Merge, successively leading to relative clauses, complex noun phrases, and matrix clauses. Finally, these are combined by coordination into one complex sentence. The result – strongly abbreviated – is depicted in (13):

[6] Note that RNR is coordination-final, not sentence-final. The entire CoP in (11), for instance, can be inserted as a complex subject: *That Joop talked to a man who BOUGHT _ and (that) Jaap talked to a man who SOLD a red car is quite surprising.*

(13)



The DP *a red car* is locally related to both verbs by Merge. The fact that the relevant VPs are deeply embedded inside the conjoined clauses in the final stage of the derivation creates a sense of non-locality, but if the analysis is correct, this is only apparently so.

In Section 4, we will see that certain other construction types involving external remerge also show signs of non-local behavior.

3. Parenthetical Merge

The possibility of *internal* and *external remerge* depends on the choice of input for Merge. The operation of Merge itself is the same in all cases.[7] Here, I will argue that we nevertheless do need a second type of Merge for another purpose, namely to account for the phenomenon of parenthesis.

Parenthesis is a cover term for a wealth of construction types, including parentheticals, comment clauses, appositions, hedges, and non-restrictive relative clauses (see Dehé & Kavalova 2007 for an overview). The internal structures of these vary greatly, but the important point is that they also have something in common. All behavior that is typical for parentheses, I believe, follows from one essential factor, namely the particular way they are related to the matrix. This general idea is not very controversial, but the opinions differ wildly on how to give it hands and feet. To begin with, it is far from obvious how to define parenthesis either syntactically or phonologically, even though everyone recognizes it intuitively. For this reason I proposed the following working definition: *parenthesis is a grammatical construction type that involves a message that is presented or perceived as secondary with respect to the host*, where *message* covers propositions, modal propositions, questions, metalinguistic comments, and so on. This

[7] Some of the literature on multidominance, and specifically on external remerge, gives another impression about this, which I think is somewhat misleading. Also, note that the phenomenon of remerge is independent of dimensionality. Multidominance (which results from doing the same thing twice or more) and 3D-structure (which involves another kind of basic relationship, as is the case for Parenthetical Merge discussed below) are independent factors.

definition ties in with Potts's (2005) conclusion that appositions and other 'conventional implicatures' involve independent lambda terms that are not 'at issue'. As an example, consider (14), which contains an appositive (i.e., non-restrictive) relative clause (ARC):

(14) You probably know Joop, who is my neighbor.

The primary message is the host clause *you probably know Joop*; the secondary message is *he (=Joop) is my neighbor*. The parenthesis cannot be denied directly. Thus, if the hearer responds *"No, that is incorrect",* this means that he or she does not know John; it does not mean that John is not the speaker's neighbor. Furthermore, the ARC is outside the scope of the modal operator *probably*: what is presented as probable is the hearer knowing John, independently of the issue whether John is the speaker's neighbor. What is more, the semantic composition of the host clause is complete by itself. A parenthesis is always a non-restrictive addition. But of course it can have an important *pragmatic* import. As for the phonology, parentheses normally are prosodically distinguished from the host. How exactly is subject to variation, but a change in pitch and speech rate are common factors. See Dehé (2009) for discussion and a qualification.

A well-known idea is that parentheses are 'orphans', which means that they are not syntactically integrated with the host clause at all (see Haegeman 1991, among others). Given the above, this does not sound entirely unreasonable. However, it also seems like giving up finding an explanation for parenthesis (or at least a formal description of it), and I am more sympathetic to approaches represented by, e.g., Espinal (1991) and Ackema & Neeleman (2004), who define special devices in order to cope with the extraordinary behavior of parentheses.[8] Apart from that, there are several problems with the idea of orphanage. The most straightforward is this. Parentheses, obviously, are both interpreted and pronounced (and thus have an effect on the phonology and the semantics of the utterance as a whole). Therefore, they must be present at the PF and LF interfaces. Given the standard Y-model of grammar (or any variation of (2)), the only way to get there is via the syntax. If a parenthesis were to be added, say, at some unspecified discourse level beyond LF, this would be problematic for the pronunciation (unless, perhaps, a completely different model of grammar is assumed, but that would require elaborate independent justification). In this respect, it is worth stressing that parentheses often interrupt the host clause: they need not be sentence-peripheral. Second, orphanage does not explain the fact that parentheses are secondary information. Third, some parentheses, such as appositions and ARCs, are directly attached to an anchor/antecedent, and they seem to form a constituent with it. In Dutch, they can be topicalized together with the anchor, but stranding in the middle field is impossible. This is illustrated in (15):

(15) a. Ik heb [Joop, *onze buurman*,] gezien.
I have Joop our neighbor seen
'I've seen Joop, our neighbor.'
b. [Joop, *onze buurman*,] heb ik _ gezien. (topicalization)
c. * Joop heb ik [_, *onze buurman*,] gezien. (stranding)

[8] Note that these proposals differ substantially from each other, and also from my own, to be spelled out below.

Such facts suggest that a parenthesis can be added on the constituent level, which implies that syntax plays a role. This is corroborated by the ambiguity of (16), where the comment clause *I think* can be related to the host clause as a whole or to the direct object alone:

(16) Next week, Joop is going to Paris, I think.
  (i) 'I think that Joop is going to Paris next week.'
  (ii) 'I think that it is Paris that Joop is going to next week.'

Fourth, it is remarkable that appositions take over the Case of the anchor in many languages, which mimics the situation in coordination. Example (17) is from Russian (thanks to Evgenia Markovskaya and Herman Heringa). See Heringa (2011) for more examples and discussion.

(17) Oni dali ih Anite, *drugomu pauku*, tože.
they gave them Anita:DAT other spider:DAT too
'They gave it to Anita, the other spider, as well.'

Finally, the simple fact that there can be parenthesis within parenthesis (within parenthesis, etc.) implies recursion – again an indication for the involvement of syntax. An example is (18), where we can distinguish five levels of interpretation:

(18) I still owe Anna – and Anna, who hit Joop, an unpleasant guy, as you know, disappeared last night – 250 dollars.

I conclude that parenthesis needs to be represented in syntax.

How can a parenthetical phrase or clause be connected to its host? Clearly, regular Merge is not fit for the job, since it automatically leads to semantic composition and a dominance hierarchy. Scope, and in fact c-command-based relationships in general, are defined over dominance. But parentheses do not interact with the matrix in such terms. Compare, for instance, the contrast in (19a/b), which tests quantifier binding of a variable embedded in a restrictive and appositive relative clause, respectively. Binding is blocked in the latter. Furthermore, Condition C effects can be lifted in parentheticals; see (20), where coreference is perfectly acceptable.

(19) a. [No climber]$_i$ talked about the mountain he$_i$ conquered last month.
  b. * [No climber]$_i$ talked about the K2, which he$_i$ conquered last month.

(20) He$_i$ said – this is typical for Joop$_i$ – that he$_{(i)}$ didn't like veggie burgers.

Thus, phrases in the matrix do not c-command into parentheses. See De Vries (2007) for a systematic overview and further discussion.[9]

Parentheses are syntactically included in the host sentence, but they are 'invisible' for scopal relationships. How do we solve this dilemma? Traditional grammarians distinguished between two fundamental relationships, hypotaxis (subordination) and parataxis (nonsubordination). Similarly, let us assume that there are two basic types of inclusion: regular

[9] Notice that c-command stretches across phases. Not only is this potentially problematic for phase theory in itself, it also shows that simple solutions for the scopal invisibility of parenthesis in terms of syntactic phase domains are not going to work – whence my turn to a more fundamental solution below.

inclusion (which corresponds to traditional dominance) and parenthetical inclusion. These notions do not follow from anything else: they are simply primitives of the grammar. As a consequence, there are also two kinds of Merge operations: (regular) Merge, and Parenthetical Merge (which I will abbreviate as par-Merge):[10]

(21) a. (regular) Merge (A, B) yields C such that
(i) C directly (regularly) includes A,
(ii) C directly (regularly) includes B, *and*
(iii) A is the merge-mate of B.
b. par-Merge (A, B) yields C such that
(i) C directly par-includes A,
(ii) C directly par-includes B, *and*
(iii) A is the merge-mate of B.

Both types of Merge form a more inclusive object, and hence a constituent. But we want to be able to say that there are two qualitatively different ways of forming a constituent. Notably, both a parenthetical and its host clause are formed internally by regular Merge, but the connection between the two requires an instance of par-Merge.

Before we get to examples, let us discuss the crucial notion of c-command. Put somewhat informally, every node c-commands its sister and everything in it – a well-founded idea which we will maintain. We can now define c-command more precisely over the basic relationships just associated with regular Merge:[11]

(22) a. A c-commands B iff there is an X such that
(i) A and X are merge-mates, *and*
(ii) B = X, or B is regularly included in X.
b. B is regularly included in X iff there is a sequence ('path')
(Y_1, ..., Y_n), where $n \geq 2$, $Y_1 = B$, $Y_n = X$, such that
for all i, $2 \leq i \leq n$: Y_i directly regularly includes Y_{i-1}.

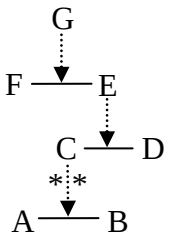
Here, (22b) simply spells out the transitivity of inclusion.

Crucially, a parenthetically merged syntactic object is not regularly included (by definition). Consequently, it will not be c-commanded by material merged later in a bottom-up derivation. Consider the abstract example in (23), where F, A, B, and D can be complex. We now need a representational convention in order to distinguish Parenthetical Merge from regular Merge. Here, I will indicate it by stars next to the vertical inclusion line.

[10] In some previous work (De Vries 2005a/b, 2007), I referred to the latter as b-Merge, with *b* short for *behindance*, thereby alluding to a three-dimensional grammar. Since Carnie (2008: 207, fn.) objects to this neologism (incorrectly attributing it to me, though I took it over from Grootveld 1994), let us maintain the terminology in the main text, which emphasizes the use instead of the (potential) representation. In this respect, it is worth mentioning that in Bosveld-de Smet & De Vries (2008), we tested the usability of different representation types applied to complex coordinate structures (also involving Right Node Raising). The 3D variants did not fare too well.

[11] Clearly, c-command can then be viewed as a direct function of regular Merge (cf. Epstein 1999), which may explain why it is this higher-order relationship that is so important, and not other logically possible ones. This result was also obtained by Kayne (1984) in terms of unambiguous paths.

(23)



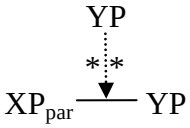
par-Merge (A, B) → C
Merge (C, D) → E
Merge (F, E) → G

In this structure, F c-commands D, which is part of the host structure, but not A and B because those are not regularly included in F's merge-mate E. Put differently, par-Merge breaks the transitive line of regular inclusion. More realistic example configurations will follow.

The above is a technical implementation of the invisibility effects illustrated in (19b) and (20). Of course, it is only the beginning of a complete solution for the phenomenon of parenthesis. Let me sum up what I intended to show so far: (i) parenthesis is fundamentally different from hypotaxis; (ii) nevertheless, parenthesis must be represented in syntax; (iii) regular Merge cannot possibly account for this; (iv) therefore, some basic stipulation is inevitable; (v) once we postulate 'parenthetical inclusion' as a primitive relationship, the required effects follow straightforwardly and parsimoniously.

There is no fixed position for parentheses in the host clause, and in fact there cannot be such a position, since selection does not play a role. Therefore, let us assume that 'independent' parentheses are freely adjoined to the spine of the structure they pragmatically interact with.[12] Suppose we would do this directly, as in (24), where XP_{par} is the parenthesis, and YP some projection of the host.

(24)



par-Merge (XP_{par}, YP) → YP

[12] One may wonder if there is a correspondence between par-Merge and Chomsky's pair-Merge. The answer is probably negative. There are a number of complications in answering this question. First, it is not entirely clear, to me at least, how the properties attributed to adjuncts follow from the formal characterization of pair-Merge (against the background of other assumptions within the Minimalist framework, which is constantly developing). Second, I am not sure that Chomsky's (2004) empirical description of the behavior of adjuncts is correct in all respects. Third, it is unknown to me how Chomsky would explain the differences between regular adjuncts and parentheses. It is undisputed that adjuncts take part in the meaning composition of the sentence ('predicate intersection'). As we have seen, this is different for parentheses. Furthermore, despite the existence of some complex anti-reconstruction cases Chomsky cites, it seems clear to me that adjuncts do not inherently display invisibility effects such as illustrated for parentheses in (19b) and (20). Straightforward examples like (i), where the anaphor inside the adjunct is bound by the subject, or (ii), where a pronoun inside an adjunct is bound by a quantifier, or (iii), where Condition C is violated, show convincingly that adjunction does not shield off c-command:

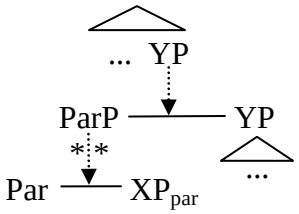
(i) $John_i$ and $Mary_j$ had breakfast [in each $other_{i\text{-}j}$'s house].
(ii) [Every man]$_i$ peeled an orange [with his_i own knife].
(iii) * He_i bought a picture [that $John_i$ liked].

Moreover, adjuncts are phonologically treated as regular constituents of the clause, and as such integrated in the overall intonation contour of the sentence, again, unlike parentheses. It may be, then, that adjuncts have to be integrated into the structure by some operation different from regular Merge (see also Rubin 2003, among others, for comments), but this operation and its consequences must be kept distinct from par-Merge.

This, however, has an unacceptable consequence, namely that the lower part of the host becomes invisible for the higher part.

A solution is to assume that XP_{par} is embedded in an abstract parenthetical phrase ParP. I will argue that this has several advantages. The essential part of the derivation and structure is given in (25):

(25)



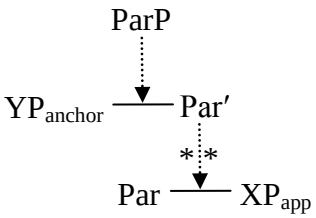
par-Merge (Par, XP_{par}) → ParP
Merge (ParP, YP) → YP

Here, the line of regular inclusion is unaffected in the host YP, but it stops immediately within ParP, with the required effect on c-command.

The head Par, which triggers the application of par-Merge, can be thought of as a specialized discourse connector, a 'parenthetical specifying coordinator' (De Vries 2009a), whose semantic effect can be compared to the 'comma operator' introduced by Potts (2005). Recall that parentheses act as 'independent lambda terms'. As I see it, such semantic effects cannot come out of nowhere: they are evoked by elements in the syntax. I suppose that a similar consideration underlies Potts's introduction of a 'comma feature' in syntax. Just to be clear: there is nothing special about the internal semantics of some parenthesis XP_{par}; it is only the connection with the host that is of a particular nature. We saw that par-inclusion starts a new c-command domain. In semantics, then, this information may trigger the start of a new lambda term. The fact that the parenthetical is syntactically embedded in the host makes sure that it is the parenthetical proposition/message that is perceived as secondary, and not the other way around. Similarly, the information that XP is actively marked as 'parenthetical' in syntax can be used by the phonological component to start a new intonational phrase with somewhat different properties.

Usually, a specifying coordinator is asyndetic, but in some cases it can materialize as an actual conjunction. An example is in (18); see further Blakemore (2005) and Kavalova (2007) for a discussion of *and*-parentheticals. What is more, it is now easier to see the link between independent parentheticals and anchored parentheses like appositions, which show clear signs of structural coordination (cf. Kraak & Klooster 1968, Quirk et al. 1985:1308, Koster 2000). A simple illustration is *Elvis Presley, or The King*. Appositions, then, – and appositive relative clauses alike – are not adjoined to the host, but structurally coordinated to the anchor: [$_{ParP}$ *anchor* [Par *appositive XP*]]; see O'Connor (2008), Cardoso & De Vries (2010), and Heringa (2011) for a more sophisticated discussion of the internal structure. Again, and perhaps more evidently so, we need the parenthetical specifying coordinator Par, as is shown in (26):

(26)



par-Merge (Par, XP_{app}) → Par′
Merge (YP_{anchor}, Par′) → ParP

Here, ParP is structurally like a regular coordination phrase. The anchor is the first conjunct, and it is visible for the host. The apposition is embedded as the parenthetical second conjunct, and hence invisible for c-command.

On a final note, the involvement of the head Par can be used to reduce the amount of overgeneration (that is, nonsensical, probably crashing derivations due to the availability of the additional operation par-Merge) by introducing the general heuristic "the parenthetical specifying coordinator Par, and only Par, triggers the application of par-Merge".

4. Combining Parenthetical Merge and *external remerge* in amalgams

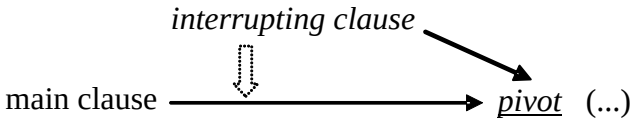
This pre-final section is a brief exploration of the combined effects of Parenthetical Merge and external remerge.

There are various ways of intertwining sentences (see Lakoff 1974, Guimarães 2004, Van Riemsdijk 2006a, Kluck 2011, among others). The illustrations in (27) contain a cleft amalgam ('Horn's case') and a *wh*-amalgam ('Andrews' case'), respectively:

(27) a. Joop got *I think it's* <u>*a didgeridoo*</u> for his birthday.
b. Joop got *you will never guess how many* <u>*instruments*</u> for his birthday.

In both examples there is an interrupting clause (printed in italics) with modal import. Unlike the situation in regular parentheticals, there is a phrase (underlined) that seems to function as the pivot between the host clause and the interrupting clause. This pivot ('callus', 'content kernel') plays a role in both clauses at the same time, which leads to a bracketing paradox. As will be clear from section 2, the application of external remerge is a potential solution to this problem: the pivot can then be shared between two clauses. The general idea is depicted in (28):

(28)

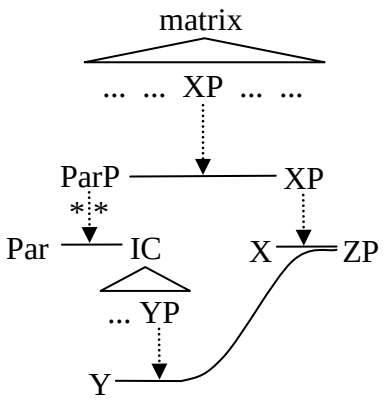


Without claiming that this is ultimately the right analysis of amalgams and related construction types, I would like to adopt it for now, and show that it leads to some interesting predictions.[13]

What then needs to be established is the relationship between the interrupting clause itself and the matrix (strangely, this issue is ignored in some of the literature on the topic). Since there is no selectional relationship whatsoever, and since the interrupting clause intuitively conveys a secondary message, I will treat the interrupting clause as a parenthetical, which, according to the previous section, is an adjunct involving Parenthetical Merge inside a ParP. This ParP must then be adjoined directly above the pivot. Abstracting away from numerous details, the structure of amalgamated sentences is as sketched in (29):

[13] Kluck (2011) objects that a sharing analysis does not adequately reflect the semantics of cleft amalgams. She proposes an interesting alternative in terms of ellipsis (sluicing).

(29)

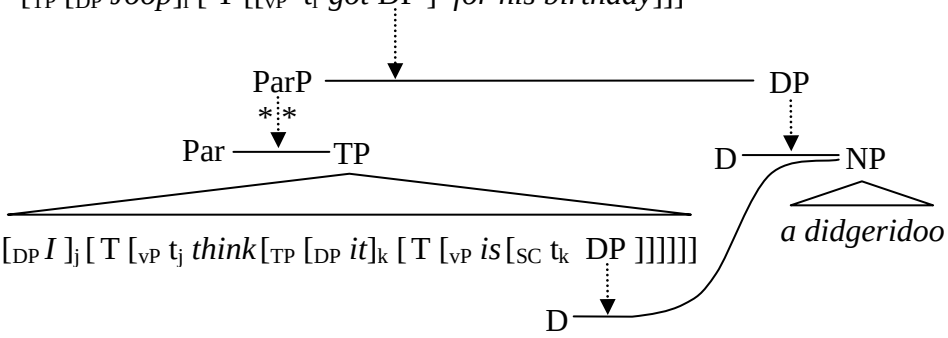


Merge (Y, ZP) → YP
Merge (X, ZP) → XP
Merge (..., YP) → ... → IC
par-Merge (Par, IC) → ParP
Merge (ParP, XP) → XP
Merge (..., XP) → ... → matrix

Here, IC is the interrupting clause, and ZP the pivot, the shared constituent.[14]

More evidence for (29) will follow below. First, let me provide a concrete example derivation of a simple cleft amalgam, as in (27a): *Joop got I think it's a didgeridoo for his birthday*. First we could Merge the noun phrase *a didgeridoo* with a high functional head associated with noun phrases, D. Immediately, *a didgeridoo* can be externally remerged with another D. There are now two DPs, which share the noun phrase *a didgeridoo*. One DP is used as a small clause predicate in a cleft construction, and we generate *it's a didgeridoo* by regular Merge, and consequently derive the more inclusive clause *I think it's a didgeridoo*. This is turned into a parenthetical by par-merging it with the specifying coordinator Par. ParP as a whole can then be adjoined inside the main clause (or rather what is to become the main clause) by merging it with the other DP (which is still a root at that stage of the derivation). The now complex DP is merged as the object argument of the verb *got*. Finally, the subject *Joop* can be added, and the main clause is functionally completed. The result is sketched in (30), in a hybrid notation:

(30) $[_{TP}\ [_{DP}\ \textit{Joop}]_i\ [\ T\ [[_{vP}\ t_i\ \textit{got}\ DP\ ]\ \textit{for his birthday}]]]$



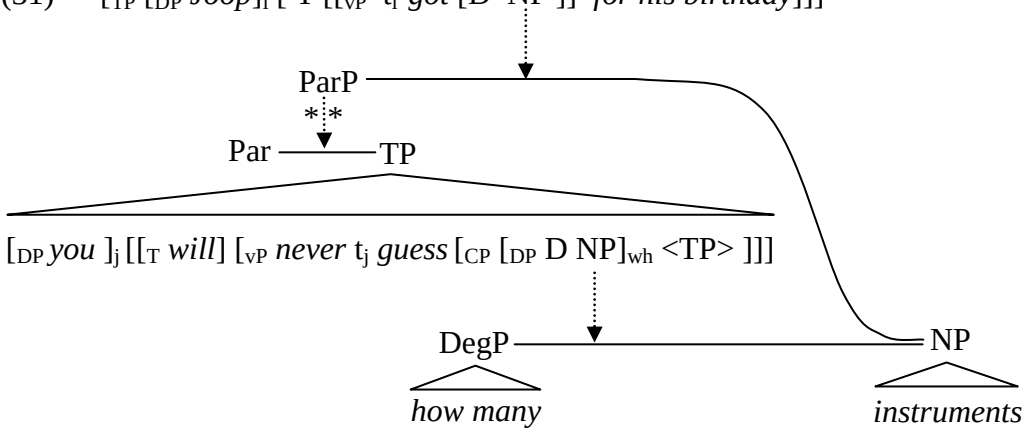
For practical purposes, movement (internal remerge) of the subjects is indicated by simple traces in the bracketed parts of the representation. Needless to say, the details of English clause structure, small clauses and noun phrases do not concern us, here.

To complete the picture, a structural sketch of the *wh*-amalgam in (27b) – *Joop got you will never guess how many instruments for his birthday* – is given in (31). *Wh*-amalgams are

[14] There are indications that what is shared is just a predicate, and that the XP that the ParP is actually adjoined to is a functional projection on top of this predicate, e.g., a strong determiner phrase turning the predicate into an argument in the matrix.

somewhat more complicated than the cleft variant since they contain a sluiced clause; this is indicated by <TP>, in which the displaced *wh*-phrase is base-merged; in this case it stands for ~~*Joop got _ (for his birthday)*~~. Moreover, note that in this example *how many* is a degree phrase related to the pivot NP.

(31) $[_{TP}\ [_{DP}\ \textit{Joop}]_i\ [\ T\ [[_{vP}\ t_i\ \textit{got}\ [D\ \ NP\ ]]\ \ \textit{for his birthday}]]]$



Let us now turn to the consequences of such an analysis.

In the previous sections, we derived some important effects of external remerge and Parenthetical Merge, namely non-local behavior and invisibility for c-command, respectively. If (29) or the concrete examples in (30) and (31) are approximately correct, three predictions ensue; see (32).

(32) (i) *Apparently non-local behavior of amalgams*: the pivot can be deeply embedded inside the interrupting clause.
(ii) *Invisibility of the interrupting clause in amalgams*: material inside the interrupting clause (modulo the pivot) is invisible for c-command-based relationships with phrases in the host structure.
(iii) *Visibility of the pivot in amalgams*: the pivot *is* visible for c-command-based relationships with phrases in the host structure.

Indeed, each of these can be empirically confirmed. First, the examples in (33) and (34) show that the pivot can be embedded in an extended IC:[15]

(33) a. Joop got *I think Jaap claimed it's <u>a didgeridoo</u>* for his birthday.
b. Joop got *I think that it was Jaap who claimed that it's <u>a didgeridoo</u>* for his birthday.
c. Joop got *I guess I have to convince you that it's <u>a didgeridoo</u>* for his birthday.

(34) a. Joop got *I am sure you will never guess how many <u>instruments</u>* for his birthday.
b. Joop got *I guess there's nobody here who can even imagine how many <u>instruments</u>* for his birthday.

[15] In order to avoid processing difficulties, such examples require an increased speech rate within the IC, and emphasis on the pivot. See De Vries (2010) for some examples in Dutch.

As in complex Right Node Raising configurations, the distance between the pivot and the selecting head in the host structure can apparently be large, measured by normal syntactic standards. In (33b), for instance, *a didgeridoo* is deeply embedded inside the interrupting clause: it is part of a cleft construction inside a relative clause inside a subordinate cleft construction. Needless to say, movement (internal remerge) across such a combination of locality domains would be completely impossible.[16] Here, movement is not at stake. Still, *a didgeridoo* is selectionally related to the verb *got* in the main clause, which is at least linearly far away. How is this possible? The answer must be that the main verb does not 'see' the object via a long path through the complex interrupting clause, but directly. The object is syntactically shared between the IC and the host, so there is a bypass. The derivation is roughly as follows. The noun phrase *a didgeridoo* is merged as an internal argument in what is to become the host, and then immediately externally remerged as a small clause predicate in what is to become the IC (or the other way around). There are now two roots. The IC is extended by a (long) series of mergers, thereby creating internal locality boundaries. When it is completed, it can, as a whole, be parenthetically adjoined to the other root (which represents an intermediate stage in the derivation of the host). The host is then completed in a normal fashion. Every step of this derivation is local. The result is like (29), with a complex IC.

Next, let us examine the invisibility of the interrupting clause.[17] The examples in (35) through (38) are taken from Kluck (2008b), enhanced with annotation. In (35) it is shown that quantifier binding of a variable inside the IC is excluded. This can be compared to the situation in (19b).

(35) a. * [No one]$_i$ is going to *he$_i$ thinks it's* *New York* this Sunday.
b. * [Every student]$_i$ bought *he$_i$ didn't realize how many* *books* that day.

By contrast, the examples in (36) show that binding of a variable inside the pivot is acceptable:

(36) a. [Every student]$_i$ sold *you can imagine how many of* *his$_i$ books* that day.
b. [No one]$_i$ is going to *I think it's* *his$_i$ girlfriend* this Sunday.

As for binding Condition C, (37) and (38) show that the IC is opaque, except for the pivot. Thus, binding causes unacceptability in (38), which seems to imply that the pivot is visible. However, the rest of the IC is opaque, so the results in (37) can be compared to those in (20), a regular parenthetical, which may contain a coreferential R-expression.

(37) a. He$_i$ had seen – *Ed$_i$ said it was* *Anna* on tv yesterday.
b. He$_i$ bought – *Ed$_i$ didn't even know (himself) how many* *books*.

[16] Tsubomoto & Whitman (2000) do report an island effect for English cleft amalgams, (not for *wh*-amalgams). However, their examples (such as "?* John is going to [$_{NP}$ e] I got angry because it was Chicago on Sunday", p. 179) are unacceptable for independent reasons. For instance, the highest predicate of the IC cannot be factive. Note that the cited example considerably improves if one more layer of embedding is added: *John is going to – I suspect I/Mary got angry because it was CHICAGO on Sunday*. See also Kluck (2011), who discusses this and related issues in more detail.

[17] The judgments of all the data in this section can be corroborated by similar sentences in Dutch, according to my own intuition (confirmed by some colleagues).

(38) a. * He_i had seen *I think it was* $\underline{Ed_i}$ on tv yesterday.
b. * He_i bought *you can imagine how many* $\underline{books\ about\ Ed_i}$.

Similar contrasts can be obtained with tests for anaphor binding and other scope-related phenomena. Furthermore, Van Maastricht et al. (2010) performed tests with so-called 'transparent free relatives' (another type of amalgamated construction) in Dutch, again with very similar results.

To sum up, the analysis of amalgams in terms of external remerge of the pivot plus Parenthetical Merge of the interrupting clause gives rise to a number of specific predictions, which can be tested. The results are very promising.

Needless to say, there are numerous aspects of these construction types that require further inquiry, both empirically and theoretically. On a final note, let me say something about theta role assignment. Since Chomsky (1981) it is generally assumed that each argument bears exactly one theta role. This might be a problem for a sharing approach to amalgamation, but I think this is not necessarily so. In fact, the Theta Criterion can be seen as support for the proposal above in which external remerge takes place *below* the DP level. Even apart from that, it is the case that in cleft amalgams (and transparent free relatives) the relevant noun phrase is used as a small clause predicate in the interrupting clause, hence not as an argument. In the examples above, it is an argument in the main clause, where it receives its one and only theta role.

5. Conclusions

Reasoning from a Minimalist perspective involving bottom-up derivations, I discussed the nature of the operation Merge and the possibility of certain unconventional mergers. The regular Merge operation has at least three possible effects, depending on the choice of input. Since the difference is relevant, I introduced the rather transparent labels *first-time merge*, *internal remerge*, and *external remerge*. It would be a misconception to think that these stand for different types of operations: Merge itself does the exact same thing in each case, namely it relates two syntactic objects, thereby forming a new hierarchical layer (which is then a new structural root). The possibility of *re*merge depends on the degree of freedom in selecting the input. If it is allowed to select a term of a non-trivial syntactic object, which by itself is also a syntactic object, remerge follows. At the cost of an additional rule, this could be prevented, resulting in a grammar of more restricted expressive power. It is hard to tell which choice is more Minimalist in spirit. I argued for the more liberal stance on other grounds. Internal remerge, then, is equivalent to regular movement. And if there is internal remerge, only a complicated rule could prevent external remerge (also described in terms of 'parallel merge', 'sideward movement', 'grafting', 'sharing', or 'interarboreal movement').

Discarding traces and copies *a priory*, a multidominance graph may be used to represent the effect of remerge generally. However, although such graphs as well as standard trees represent the hierarchical effect of Merge well, they do not directly reflect the core aspect of Merge, namely that it relates the syntactic input objects to each other (leading to semantic composition), thereby fulfilling and/or facilitating all kinds of licensing. For this reason, I put forward an alternative representation type that does emphasize the 'merge-mate' (sisterhood) relationship. Remerge, whether internal or external, simply means that some object can be directly related to more than one other object. To me, this seems an entirely natural assumption.

Multidominance is only a secondary effect. In the past, the aversion of unconventional (complex, clumsy, weird, ...) representations may have lead to overhasty decisions concerning what we now call remerge. But what counts, of course, is the underlying theory, and not the practical problem of convincingly representing the results.

External remerge leads to a doubly-rooted structure. During the syntactic derivation this is harmless. However, from a bird's eye perspective it constitutes a temporary problem that must be resolved before the structure can be processed in the phonological and semantic components. The linearization procedure requires a single-rooted structure (unless, perhaps, complicated additional assumptions are made). As for the semantics, the relationship between the two substructures must be made explicit. In this paper, two such relationships have been discussed: coordination (namely in Right Node Raising constructions) and parenthetical insertion (in amalgams). Furthermore, I showed that derivations involving external remerge potentially give rise to an interesting phenomenon: apparently non-local behavior. By contrast, it follows from the nature of Merge that locality boundaries cannot be bypassed in configurations involving internal remerge.

In section 3, I argued that standard Merge, with its three instantiations depending on the input, is insufficient to account for all relevant language data. The problem is that Merge automatically creates a hypotactical structure (standard embedding). This raises the question what to do with parenthesis, a vast phenomenon encompassing dozens of different construction types, which have a number of related properties in common: they are not selected for by anything in the host structure, they are not embedded in the regular sense, and they are scopally independent. Despite that, there are good arguments to deal with them at the level of syntax: parentheses have effects on both the PF and LF interface, they are linearly integrated with the host, they can apply at the constituent level or the matrix level, they can be recursively added. Therefore, I proposed a second type of merger, Parenthetical Merge. As far as I can see, there is no way of explaining parenthetical inclusion in more basic terms, which is why it needs to be stipulated as a primitive of the grammar. Technically, par-Merge breaks the transitive line of dominance in the host structure, and consequently shields parentheses from c-command-based relationships with material higher up in the host. This type of 'invisibility' can be tested on the basis of various phenomena involving scope, such as variable binding by a quantifier, or Condition C effects. Furthermore, I associated par-Merge with the presence of a functional projection ParP, which is a parenthetical specifying coordination phrase. Concretely, a parenthesis is always the complement of the head Par. For anchored parentheses such as appositions, I assumed that they are structurally coordinated with the anchor. 'Independent' parentheses (parentheticals) can – in principle – be adjoined to any projection of the host, which explains their variable surface position.

Section 4 explored an analysis of sentence amalgamation, in particular cleft amalgams and *wh*-amalgams, as parentheticals of which the pivot constituent is structurally shared with the host. Thus, the derivation of these construction types involves par-Merge (at the top) as well as external remerge (at the bottom). From this, three predictions follow: first, potentially non-local behavior resulting from the application of external remerge, meaning, in this case, that the pivot can be deeply embedded inside the interrupting clause; second, scopal invisibility of the interrupting clause; and third, the exception of the pivot from this. All three were found to be correct on the basis of examples from English (shown here), and have been confirmed by speakers of Dutch.

References

Ackema, Peter & Ad Neeleman (2004). *Beyond Morphology: Interface Conditions on Word Formation*. Oxford: Oxford University Press.

Blakemore, Diane (2005). '*and*-parentheticals', *Journal of Pragmatics* 37: 1165–1181.

Bobaljik, Jonathan (1995). 'In Terms of Merge: Copy and Head Movement', *MIT Working Papers in Linguistics* 27: 41–64.

Bobaljik, Jonathan & Samuel Brown (1997). 'Interarboreal Operations: Head Movement and the Extension Requirement', *Linguistic Inquiry* 28: 345–356.

Boeckx, Cedric (2008). *Bare Syntax.* Oxford: Oxford University Press.

Bosveld-de Smet, Leonie & Mark de Vries (2008). 'Visualizing Non-Subordination and Multidominance in Tree Diagrams: Testing Five Syntax Tree Variants', in G. Stapleton, J. Howse, and J. Lee (eds.), *Diagrams 2008*, LNAI 5223. Berlin: Springer, 308–320.

Cardoso, Adriana & Mark de Vries (2011). 'Internal and External Heads in Appositive Constructions'. Manuscript, University of Lisbon & University of Groningen.

Carnie, Andrew (2008). *Constituent Structure.* Oxford: Oxford University Press.

Chen-Main, Joan (2006). *On the Generation and Linearization of Multi-dominance Structures*. PhD dissertation, John Hopkins University, Baltimore, Maryland.

Chomsky, Noam (1957). *Syntactic Structures*. The Hague: Mouton.

Chomsky, Noam (1981). *Lectures on Government and Binding*. Dordrecht: Foris.

Chomsky, Noam (1995). *The Minimalist Program*. Cambridge, MA: MIT Press.

Chomsky, Noam (2004). 'Beyond explanatory adequacy', in Adriana Belletti (ed.), *The Cartography of Syntactic Structures, Vol. 3, Structures and beyond*. Oxford: Oxford University Press.

Chomsky, Noam (2005). 'On Phases'. Manuscript, MIT.

Citko, Barbara (2005). 'On the Nature of Merge: External Merge, Internal Merge, and Parallel Merge', *Linguistic Inquiry* 36: 475–496.

Collins, Chris (2002). 'Eliminating Labels', in Samuel Epstein & Daniel Seely (eds.), *Derivation and Explanation in the Minimalist Program*. Oxford: Blackwell, 42–64.

Dehé, Nicole (2009). 'Clausal parentheticals, intonational phrasing, and prosodic theory', *Journal of Linguistics* 45: 569–615.

Dehé, Nicole & Yordanka Kavalova, eds. (2007). *Parentheticals*. Amsterdam: John Benjamins.

Di Sciullo, Anna Maria & Daniela Isac (2008). 'The Asymmetry of Merge', *Biolinguistics* 2: 260–290.

Epstein, Samuel (1999). 'Un-Principled Syntax and the Derivation of Syntactic Relations', in Samuel Epstein & Norbert Hornstein (eds.), *Working minimalism*. Cambridge, MA: MIT Press, 317–345.

Epstein, Samuel, Höskuldur Thraínsson & Jan-Wouter Zwart (1996). 'Introduction', in Werner Abraham et al. (eds.), *Minimal Ideas*. Amsterdam: John Benjamins, 1–66.

Espinal, M. Teresa (1991). 'The Representation of Disjunct Constituents', *Language* 67: 726–762.

Fortuny, Jordi (2008). *The Emergence of Order in Syntax*. Amsterdam: John Benjamins.

Gärtner, Hans-Martin (2002). *Generalized Transformations and Beyond: Reflections on Minimalist Syntax*. Berlin: Akademie Verlag.

Goodall, Grant (1987). *Parallel Structures in Syntax: Coordination, Causatives and Restructuring.* Cambridge: Cambridge University Press.
Gracanin-Yuksek, Martina (2007). *About Sharing*. PhD dissertation, MIT.
Grootveld, Marjan (1994). *Parsing coordination generatively*. PhD dissertation, Leiden University.
Guimarães, Maximiliano (2004). *Derivation and Representation of Syntactic Amalgams.* PhD dissertation, University of Maryland.
Haegeman, Liliane (1991). 'Parenthetical adverbials: The radical orphanage approach', in Shuji Chiba et al. (eds.), *Aspects of Modern English Linguistics: Papers Presented to Masatomo Ukaji on His 60th Birthday*. Tokyo: Kaitakusha, 232–254.
Heringa, Herman (2011). *Appositional constructions.* PhD dissertation, University of Groningen. LOT Dissertation Series 294.
Hinzen, Wolfram (2006). *Mind Design and Minimal Syntax.* Oxford: Oxford University Press.
Kavalova, Yordanka (2007). '*And*-parenthetical clauses', in Nicole Dehé & Yordanka Kavalova (eds.), *Parentheticals*. Amsterdam: John Benjamins, 145–172.
Kayne, Richard (1984). *Connectedness and Binary Branching*. Dordrecht: Foris.
Kluck, Marlies (2008a). 'Intertwined clauses, interacting propositions. A note on the interpretive aspects of sentence amalgamation', *Proceedings of ConSOLE XVI*, 77–101.
Kluck, Marlies (2008b). 'Syntactic amalgams in the paratactic domain'. Talk presented at the MIT Linglunch, 25 September 2008.
Kluck, Marlies (2011). *Sentence Amalgamation*. PhD dissertation, University of Groningen. LOT Dissertation Series 285.
Kluck, Marlies & Mark de Vries (to appear). 'Cumulative Rightward Processes', in Heike Walker, Manfred Sailer & Gert Webelhuth (eds.), *Rightward Movement from a Cross-linguistic Perspective*. Amsterdam: John Benjamins.
Koster, Jan (2000). 'Extraposition as parallel construal'. Manuscript, University of Groningen. URL: <http://www.let.rug.nl/~koster/papers/parallel.pdf>
Koster, Jan (2007). 'Structure Preservingness, Internal Merge, and the Strict Locality of Triads', in Simin Karimi, Vida Samiian & Wendy K. Wilkins (eds.), *Phrasal and Clausal Architecture: Syntactic derivation and interpretation. In honor of Joseph E. Emonds*. Amsterdam: John Benjamins, 188–205.
Kraak, A. & Wim Klooster (1968). *Syntaxis.* Culemborg: Stam-Kemperman.
Lakoff, George (1974). 'Syntactic amalgams', in Michael La Galy, Robert Fox & Anthony Bruck (eds.), *Papers from the 10th regional meeting of the Chicago Linguistic Society*, 321–344.
Langendoen, Terence (2003). 'Merge', in Andrew Carnie et al. (eds.), *Formal Approaches to Function in Grammar. In Honor of Eloise Jelinek*. Amsterdam: John Benjamins, 307–318.
Maastricht, Lieke van, Marlies Kluck & Mark de Vries (2010). 'How invisible are Transparent Free Relatives?' Talk presented at the Tabu-day, 3 June 2010, University of Groningen.
McCawley, James (1982). 'Parentheticals and Discontinuous Constituent Structure', *Linguistic Inquiry* 13: 91–106.
Nunes, Jairo (2001). 'Sideward Movement', *Linguistic Inquiry* 32: 303–344.
Nunes, Jairo (2004). *Linearization of Chains and Sideward Movement.* Cambridge, MA: MIT Press.
O'Connor, Kathleen (2008). *Aspects de la syntaxe et de l'interprétation de l'apposition à antécédent nominal*. PhD dissertation, Université Charles de Gaulle, Lille 3.

Potts, Christopher (2005). *The Logic of Conventional Implicatures*. Oxford: Oxford University Press.
Quirk, Randolph et al. (1985). *A Comprehensive Grammar of the English Language*. London: Longman.
Riemsdijk, Henk van (2006a). 'Grafts Follow from Merge', in M. Frascarelli (ed.), *Phases of Interpretation*. Berlin: Mouton de Gruyter, 17–44.
Riemsdijk, Henk van (2006b). 'Free relatives', in Martin Everaert & Henk van Riemsdijk (eds.), *The Blackwell companion to syntax: Volume II*. Oxford: Blackwell, 338–382.
Rubin, Edward (2003). 'Determining pair-merge', *Linguistic Inquiry* 34: 660–668.
Starke, Michal (2001). *Move Dissolves into Merge: a Theory of Locality*. PhD dissertation, University of Geneva.
Tsubomoto, Atsuro & John Whitman (2000). 'A type of head-in-situ con-struction in English', *Linguistic Inquiry* 31: 176–183.
Vries, Mark de (2005a). 'Merge: Properties and Boundary Conditions', *Linguistics in the Netherlands* 22: 219–230.
Vries, Mark de (2005b). 'Coordination and Syntactic Hierarchy', *Studia Linguistica* 59: 83–105.
Vries, Mark de (2007). 'Invisible Constituents? Parentheses as B-Merged Adverbial Phrases', in Nicole Dehé & Yordanka Kavalova (eds.), *Parentheticals*. Amsterdam: John Benjamins, 203–234.
Vries, Mark de (2009a). 'Specifying Coordination: An Investigation into the Syntax of Dislocation, Extraposition and Parenthesis', in Cynthia R. Dreyer (ed.), *Language and Linguistics: Emerging Trends*. New York: Nova, 37–98.
Vries, Mark de (2009b). 'On Multidominance and Linearization', *Biolinguistics* 3: 344–403.
Vries, Mark de (2010). 'Apparent nonlocality', *Linguistics in the Netherlands* 27: 129-140.
Wilder, Chris (2008). 'Shared constituents and Linearization', in Kyle Johnson (ed.), *Topics in Ellipsis*. Cambridge: Cambridge University Press, 229–258.
Williams, Edwin (1978). 'Across-The-Board Rule Application', *Linguistic Inquiry* 9: 31–43.
Zwart, Jan-Wouter (2006). 'Local Agreement', in Cedric Boeckx (ed.), *Agreement Systems*. Amsterdam: John Benjamins, 317–339.
Zwart, Jan-Wouter (2011). 'Structure and order: asymmetric merge.' in Cedric Boeckx (ed.), *The Oxford Handbook of Linguistic Minimalism*. Oxford: Oxford University Press, 96-118.