Prospects for Top-Down Derivation*

Jan-Wouter Zwart
University of Groningen
PO Box 716, NL-9700AS
Groningen, The Netherlands
c.j.w.zwart@rug.nl

Abstract

This article explores a model of grammar involving top-down derivations, where each step ("split-merge") yields an asymmetric pair of elements relevant to the expression of order, information, and grammatical features. These derivations are inevitably layered, in the sense that the output of a previous derivation may appear as an atom in the numeration for the next derivation. It is suggested that opacity effects follow from the layering of derivations, not from conditions on movement. While the main questions surrounding the model contemplated here involve the 'when' and 'what' of merge, this article focuses on the more preliminary question of the 'how' of merge.

Key words: syntax, top-down derivation, merge, dependency, opacity, A'-movement, phases, layered derivations, model of grammar, simplicity.

Table of Contents

* This article was written in the context of the research program Dependency in Universal Grammar (DIUG), sponsored by the Netherlands Organization for Scientific Research NWO. It has benefited from discussion with the members of the DIUG research group, Jordi Fortuny, Aysa Arylova and Evgenia Markovskaya. I am thankful to them, especially to Jordi Fortuny, to whom the idea of realizing a top-down derivation via split-merge came while he was practicing the violin. Thanks are also due to other members of the Syntax and Semantics group of the Center for Language and Cognition Groningen, in particular Jan Koster, Radek Šimík, and Mark de Vries, but noone excluded. I have been developing the concept of "simplest merge" over a period of several years, and owe a debt of gratitude to audiences in many places, including Harvard University, Indiana University (the 2004 Syntax Fest), the University of Michigan, University College London, Princeton University, Carson-Newman College, the University of Tromsø, and the Universities of Tilburg, Leiden, Utrecht and Groningen. Thanks in particular to Noam Chomsky, David Pesetsky, Norvin Richards, Cedric Boeckx, Steven Franks, Sam Epstein, Daniel Seely, Robert Truswell, Adele Goldberg, Robert Freidin, Edwin Williams, and Michael Putnam. All errors (and problems) are mine.

1. Introduction

This article presents a general discussion of the nature of syntactic derivation, the (model of the) process by which language users relate a linearly ordered string of elements to a hierarchical constituent structure. In the tradition of generative grammar, it is common to view structure as the result of a simple recursive operation by which elements from a resource are selected and combined: Merge. Viewed in this way, the derivational procedure operates in a bottom-up fashion. My main point in this paper will be that for a proper understanding of the generative procedure, we need to distinguish between a) the iterative application of a single rule within a domain (a *stage* of the derivation) and b) a recursive procedure by which the output of one stage may function as an item in the resource for a next stage (yielding a *layered derivation*). If these two processes (Merge and derivation layering) are properly distinguished, there is little reason for maintaining that the iterative procedure (Merge) works in a bottom-up fashion.

What I intend to do here is a) sketch the outlines of a top-down structure building operation (split-Merge), b) relate the structure building procedure to basic properties of syntactic structure (constituency, order, dependency, interpretability), and c) consider the nature of layered derivations. The discussion emphasizes simplicity and ignores many (sometimes dire) consequences, which in fact jeopardize the entire approach. A fuller discussion of these consequences must await another occasion.

The article has the following structure. In section 2 I propose the new approach to structure building, involving top-down merger, which I argue is the simplest form merge could take. In section 3 I explore the properties of the output of the derivation in terms of linear order, morphology, and interpretation. Section 4 presents the layered nature of derivations, which suggests a theory of opacity which I apply in section 5 to account for certain constraints on long-distance dependencies. Section 6 concludes.

2. Simplest merge

Every derivation of syntactic structure needs (a) a set of elements N manipulated in the course of the derivation, called 'numeration', and (b) a procedure establishing relations among the members of N, called 'merge'. Simplicity considerations then demand:

(1) *Simplicity requirements on the derivational procedure*

a. merge manipulates a single element of N at each step of the derivation

b. merge manipulates each element from N only once

These requirements are not met in standard conceptions of the derivational procedure, which describe merge as an operation combining two elements (hence manipulating more than one element), and which allow a merged element to be merged again ('internal merge', i.e. movement). While these deviations from the simplicity requirements seem minimal or perhaps unavoidable, it should be pointed out that they introduce stipulations unwanted in a truly minimalist approach.

Thus, a derivational procedure that allows merge to manipulate two elements at a single step in the derivation can disallow merge to manipulate more than two elements only by stipulation (since 2, unlike 1, is not the absolute minimal number). This stipulation is nevertheless needed, if the system is supposed to derive only binary branching structures (which we assume without further demonstration here). Likewise, a procedure that allows a merged element to be moved (remerged) essentially states that one of the two elements manipulated maybe contained within the structure being derived. But if one of the elements manipulated may be contained within the structure being derived, the possibility that the other element being manipulated at that step of the derivation is also contained within the structure being derived can be excluded only by stipulation (the 'extension condition'). Yet this stipulation is needed, if we want the system to be unable to derive bizarre and in fact endlessly looping structures not attested in human language.

Adhering to the simplicity requirements in (1), then, eliminates seemingly inevitable stipulations and yields a closer match between the structures generated by the derivational procedure and the actual phenomena of human language.

Concretely, I would like to propose that each step in the derivational procedure turns the numeration N (a set, or, perhaps more appropriately, an array) into the ordered pair $P = \langle x, y \rangle$, where $x \in N$ and $y = (N - x)$. In other words, merge splits N into a pair consisting of a (designated) member of N and its residue in N. The numeration, then, is reduced at each step, and each next step in the derivational procedure targets the residue of the numeration created by the previous step, until N is empty.

Consider how this procedure turns a five-member numeration into a binary branching structure.

(2) *Split-merge*

a. N = { a, b, c, d, e }

b. merge: split $x \in N$ off from N, yielding $\langle x, N - x \rangle$

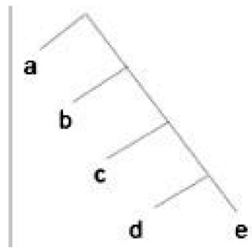
c. the derivation:

1. merge *a* yielding $\langle$ a, { b, c, d, e } $\rangle$
2. merge *b* yielding $\langle$ a, $\langle$ b, {c, d, e } $\rangle\rangle$
3. merge *c* yielding $\langle$ a, $\langle$ b, $\langle$ c, { d, e } $\rangle\rangle\rangle$
4. merge *d* yielding $\langle$ a, $\langle$ b, $\langle$ c, $\langle$ d, { e } $\rangle\rangle\rangle\rangle$
5. merge *e* yielding $\langle$ a, $\langle$ b, $\langle$ c, $\langle$ d, $\langle$ e, { } $\rangle\rangle\rangle\rangle\rangle$

The derivation in (2c) yields the ordered n-tuple in (3), which may be represented graphically as (4).

(3) ⟨ a, b, c, d, e ⟩

(4)



As can be seen, the procedure targets a single element of N at each step in the derivation, thus meeting simplicity requirement (1a). The simplicity requirement (1b) is also met, as the procedure does not allow an element *x* to be manipulated twice. This is because after *x* has been merged (i.e. split off from N), no next step in the procedure targets *x* or an entity containing *x*. We return to the question of how to describe movement phenomena within this system in sections 4 and 5.

The system described here takes the derivation to be a procedure that transforms an unordered or unstructered collection of elements (i.e. a set or an array) into a structure. This differs slightly but, I believe, significantly from the standard view, which takes merge to involve a transfering process taking elements out of a resource (the numeration) into a structure. This standard view raises questions as to the nature of the resource, the structure, and the element *x* merged after each step in the procedure, calling for a decision as to whether what is being merged is *x* or a token/copy of *x*. Similar questions arise when elements are 'moved' within the structure (via 'internal merge'). These questions are difficult to settle, and I am inclined to think that they are an artifact of the model of grammar separating the numeration and the structure as distinct objects (in space, as it were), which requires that the metaphor of movement is applied to each operation performed in the course of the derivation.

Putnam and Stroik (2008) raise the important point that every derivational system needs to provide an answer to the following questions: (a) What drives the derivational procedure? and (b) What ends it? Much work currently being done within the minimalist framework of generative grammar (including Putnam and Stroik 2008) assumes that grammatical features and the need to check and/or eliminate them provide the key to answering these questions. In the system contemplated here, features are not assumed to play this crucial role. The procedure is driven by the need to create order among the elements of the numeration, and it ends when a total ordering of the members of (the initial) numeration is established. As discussed in section 3, I take order, structure, and morphology to be intimately linked, suggesting that what the derivational procedure establishes is grammatical information.

The derivational procedure illustrated in (2) operates in a top-down fashion, in the sense that the first pair created corresponds to the highest pair in the tree structure representation in (4). This implies that properties of lexical heads (having to do with argument structure or subcategorization features) play no role in driving the derivation (see again section 3). Top-down derivations within generative grammar have been proposed and defended earlier, most notably by Phillips (2003) and Chesi (2007). In Phillips's system, adopted by Chesi, merge adjoins material to the most deeply embedded right branch of the structure, essentially splitting that branch by the addition of new material from the numeration. This system differs from the one contemplated here in that it takes merge to be an operation importing material into the structure from some resource, as in the more traditional bottom-up derivational system.[1]

3. What merge yields

3.1. Order

Standard conceptions of merge, in which merge combines two elements, take the output of merge to constitute a set rather than an ordered pair (Chomsky 2001, 2008). The distinction between a set and an ordered pair is quite tenuous in Chomsky's system, which involves the notion of a label designating the properties of the set deriving from one of its members (the head). The label is represented as a copy of the head merged to the set, as in (5):

(5) *Merge in the system of Chomsky (2001)*

a. set yielded by Merge: $S = \{ \alpha, \beta \}$

b. labeled set where α is the head of *S*: $S = \{ \alpha, \{ \alpha, \beta \}\}$

As noted by Langendoen (2003:310), $\{ \alpha, \{ \alpha, \beta \}\}$ (in fact, $\{\{\alpha\},\{\alpha, \beta\}\}$) is the set-theoretic notation of the ordered pair $\langle \alpha, \beta \rangle$, as defined in Kuratowski (1921). In Chomsky's system, then, ordering among the members of the output of Merge *S* comes about as a function of the properties of the members of *S*, in particular the head-nonhead distinction.

In the system proposed here, ordering is not a function of features of the elements merged, but a function of the derivation itself. More precisely, the circumstance that the steps in the derivation are temporally ordered yields an ordering of the elements affected by these steps.[2]

The relation between the temporal ordering of the steps in a derivation and the linear ordering of the elements involved in the derivation can be made precise in the following way. Adopting the derivational system in (2), we observe that the set of elements merged (i.e. split off from the numeration) grows with each step in the

1. This particular implementation of the idea of top-down derivation is essential to Phillips' explanation of the properties of Right Node Raising in English, which requires that the right branch be a lexical item; this explanation is lost in the system contemplated here.
2. See Jaspers (1998:109) for the same idea from the perspective of a bottom-up derivational procedure.

derivation. If we consider only the first two steps, the sets of elements merged at each step are:

(6) 1. after the first step: { a }
 2. after the second step: { a, b }

The set of sets of elements merged after step 2 is:

(7) { { a }, { a, b } }

which is the set-theoretical notation of the ordered pair ⟨ a, b ⟩. It follows that after step 5 in (2c) we obtain the set of sets of elements merged in (8), which is equivalent to the ordered n-tuple in (3):

(8) { { a } , { a, b }, { a, b, c}, { a, b, c, d }, { a, b, c, d, e } }

The idea of deriving order from nested sets originates with Fortuny (2008) (though details of the implementation differ).

As Fortuny (2008) shows, the actual linear order of the words and phrases involved in the derivation, which is established at the interface component dealing with sound, may be derived straightforwardly from the output of the derivational procedure if that output is an ordered n-tuple. The simplest implementation of that idea appears to be (9):[3]

(9) *Linear Correspondence Axiom (revised from Kayne 1994)*

$\langle \alpha, \beta \rangle = / \alpha\ \beta /$

It will be recalled that in the definition of Kayne (1994), linear order is a function of structural complexity, determined in a global manner (i.e. by inspecting the entire tree structure). To be precise, the order of the terminal elements of a linguistic tree structure is a function of the c-command relations among the nonterminal elements of the tree.[4] Crucially, mutual c-command of nonterminal nodes precludes a linear ordering of the terminals dominated by those nonterminal nodes, and therefore Kayne's system necessitates a number of stipulations relating to structure and to the definition of the c-command relation.[5]

In the system proposed here, no global computation of the relations among terminals or nonterminals is needed, and the ensuing stipulations can be avoided. It follows that the structure-order correspondence envisioned in Kayne (1994) may be

3. Material between slashes is ordered at the sound component, i.e. ordered in time.
4. *x* c-commands *y* iff *y* is (dominated by) a sister of *x*.
5. The most significant stipulation relating to structure is that a single-term complement is dominated by a vacuous branching node, and the most significant stipulation relating to c-command is that the sister of a specifier does not c-command that specifier or its terms.

made compatible with the 'bare phrase structure' desideratum expressed in Chomsky (1995).

3.2. Information

The derivational procedure as described in (2) creates relations among the elements in the numeration N. Each step in the procedure separates an element *x* from N, creating an opposition between *x* and the residue of N (i.e. N – *x*). Minimally, then, the procedure yields information in terms of constituency or part-whole relations (mereology).

A minimalist assumption would be to take the relation between *x* and the residue of N to be the only relation relevant to syntax. This also seems the strongest implementation of the Derivational Approach to Syntactic Relations (DASR) of Epstein et al. (1998), which holds that syntactic relations exist only among elements merged with each other. It also derives Zwart's (1993:373) conjecture that all syntactic licensing relations are sisterhood relations.

The derivation yields a sequence of ordered pairs. A constituent is defined as a member of an ordered pair. Members of an ordered pair are each other's sisters. The dominance relation is not defined in this system, nor is the concept of projection. I proceed on the assumption that these are not primitive concepts and relations, but space does not permit a fuller treatment at this point.

I propose to define 'syntactic position' in the following way. After merge has split off *x* from N, creating the ordered pair $\langle x, (N - x) \rangle$, the configuration in which we find *x* is defined as the syntactic position of *x*. Merge, then, creates a syntactic position for every element it splits off from the numeration. It follows that the members of N have no syntactic position before the application of merge, and neither have the members of the residue of N after each application of merge. The narrowest reading of the definition of 'syntactic position' as proposed here also denies that the sister of *x* has a syntactic position (as it is not split off from N). It remains to be seen whether this is a useful consequence.

It is important to realize that *x* split off from N is a single item (an atom), even if it may be structurally complex. We return to this in section 4. The crucial point is that the numeration may contain elements created in the context of a previous derivation. These elements are structurally complex, and this will be relevant to the interface components, where they receive a prosodic interpretation typical of phrases rather than of words (cf. Zwart 2003). However, in narrow syntax these elements are treated as atoms. This has the important consequence that the terms of a complex element *x* which is part of a numeration N in the context of a derivation D may not be split off from N (or *x*) in D.

There is a clear asymmetry, then, between *x* and its sister, (N–*x*), in that *x* is an atom throughout the derivation, whereas its sister is subject to further operations merge, splitting off further members of N and assigning them syntactic positions. Yet at each step in the derivation, the sister of *x* is a single whole, and relations between *x* and members of its sister (N–*x*) must by definition be relations between *x* and (N–*x*). In other words, the system proposed here entails that the relation of c-command reduces to the sisterhood relation.

More generally, we may state that (N–*x*) has no distinct grammatical properties, in the sense that its properties are a union of the properties of its members. Only members split off from the (residue of) N acquire distinct syntactic properties (such as a syntactic position and a grammatical function). It follows that every feature acquired by (N–*x*) as a function of its relation with *x* is shared by all members of *x*.

The asymmetry between *x* and (N–*x*) may be responsible for the fact that it is generally much easier to determine the grammatical category of a left branch (fronted) element than that of its sister. For instance, while a subject must be nominal, the predicate may be of any category, and even if it contains a verbal core, there are other candidates, such as the element [tense], for determining the categorial status of the predicate. Also, the status of a verb phrase is brought out much more clearly by fronting in languages like Dutch and German where word order freedom in the middle field makes a clear demarcation of the verb phrase difficult. It may be, then, that only elements split off from the numeration have a clear grammatical category, and that the residue of the numeration at each step of the derivation is essentially a diffuse category.

If the asymmetry between *x*, split off from N by merge, and its sister (N–*x*) is characterized correctly in the above, it follows that (N–*x*) is syntactically inert and is by definition on the receiving end of any grammatical relation established as a function of merge. More precise, (N–*x*) has no grammatical features to share with *x*, whereas nothing precludes that *x* shares grammatical features with (N–*x*). It follows that in every dependency created by merge, (N–*x*) is the dependent of *x* (and *x* is a nondependent).

We may generalize the conclusion and state:

(10) *Dependency as a function of merge*

Merge creates an ordered pair $\langle \alpha, \beta \rangle$ where β is the dependent of α

An alternative formulation would be to state that merge turns a numeration into a dependent of one of its members, i.e. it creates a dependency relation between one of its members and the residue of the numeration.

It remains to define dependency. The most general definition appears to be:

(11) *Dependency*

α is a dependent of β if α is interpreted by reference to β

According to this definition, a predicate is a dependent of its subject, an inflected verb is a dependent of the element controlling its agreement morphology, a phrase containing a variable is a dependent of the operator binding the variable, the complement of a preposition is a dependent of the preposition, etc.

The derivation as a whole yields a record of dependencies. Since these dependencies are interpreted at the interfaces, the system meets the minimalist requirement that every operation that is part of it serves interpretability at the interfaces.

3.3. Morphology

I assume a model of grammar in which inflectional morphology is realized at the interface component dealing with sound. More precisely, narrow syntax produces an ordered n-tuple of syntactic objects in need of morphophonological interpretation. The component of morphology (hence: Morphology) then takes a syntactic object and returns a form ('morphological conversion').

Inflectional features are relevant to the process of morphological conversion, in that Morphology returns the form that best matches the inflectional features of the syntactic object to be converted.[6] A minimalist assumption is that inflectional features have no further relevance, in particular no relevance to syntax. Inflectional features flag dependencies, and are a by-product of the syntactic dependency which is a function of merge.

Concretely, in (12), *John* has the features [person: 3] and [number: SG] which it shares with its sister, indicated with square brackets:

(12) John [goes to school every day]

The derivation of (12) involves the numeration in (13a)(*to school* and *every day* are outputs of previous derivations) and a first step in which *John* is split off from the numeration, creating the pair in (13b).

(13) a. { John, go, [to school], [every day] }

b. ⟨ John, { go, [to school], [every day] } ⟩

After merge, *John* has the syntactic position of subject, and its sister (the set ultimately spelled out as *goes to school every day*) is its dependent. The particular dependency relation established between *John* and *goes to school every day* is that of predication. As a function of the relation, *John* may share features with its sister. The residue of the numeration (i.e. the set in (13b)) then acquires these features as a function of merge. Since every member of an ordered pair is a single whole at that stage of the derivation, each term of the residue of the numeration has the potential to realize the features acquired as a function of merge. In English, only the verb has a paradigm of forms specialized for the relevant features, which is why the relation between the subject and the predicate is only flagged on the verb.[7]

Inflectional features, then, are typically emerging properties, acquired in the course of a derivation. This is particularly true of the features branded 'uninterpretable' in current minimalist work (starting from Chomsky 1995); the interpretable features, such as number on noun phrases, are taken to be inherently pre-

6. In the model of grammar assumed here, Morphology has access to the inflectional paradigms associated with lexemes which are stored in a Lexicon.
7. See Zwart 2006a,b for the typological variation in spelling out inflectional features on terms of a predicate.

sent on the terms of a numeration. In the system proposed here, neither type of inflectional features has the function of triggering syntactic operations, they merely narrow down spell-out options at the interface component dealing with sound.

Space does not allow me to treat instances of inflectional morphology in greater number or in more detail, but various aspects of inflectional morphology from the perspective adopted here are discussed in Zwart (2006a,b,c,e).

3.4. Semantics

As with inflectional morphology, a full discussion of the semantic relations among clausal constituents cannot be presented here, but I am operating on the assumption that the range of clause level semantic relations is limited, and that they may be analyzed as a function of merge as described here. That is, the semantic relation between an element split off from the numeration and the residue of the numeration is ideally described as an aspect of the dependency among these elements that the very operation of split-merge creates.

I assume that the default interpretation of the first element split off from the numeration is that of subject, which would cause the residue of the numeration to acquire the function of predicate. Other interpretations of the dependency relation between sisters (topic-comment, scope-domain, etc.) are also possible, and the question how each interpretation comes about must be left open at this point.

While this discussion is quite preliminary, it is perhaps important to note that no recourse to a feature triggering subject placement (i.e. the EPP-feature of Chomsky 2000:102) is envisioned here. This is because the subject is not first generated in a position lower in the syntactic tree structure, motivated by considerations of argument structure realization (i.e. the VP-internal subject hypothesis is not entertained). Argument structure, on the view advocated here, is not a driving force in syntax, but part of the interpretation of syntactic structure.[8]

It also follows that the system contemplated here does not allow for the possibility of A-movement, the operation that moves an element from an argument position to a position expressing its grammatical function. Apart from movement of the subject from its presumed VP-internal base position and object shift in languages like Dutch and German, A-movement is taken to be involved in the placement of the internal argument in subject position in passives (14a) and in raising the argument of an embedded clause (in brackets) to the subject position in its matrix clause (14b).

(14) a. John was arrested

b. John seems [to be intelligent]

8. Section 4 argues that this is inevitable in a framework that assumes layered derivations as defined there.

The system now requires these construction types to involve no movement, so that the proper interpretation of *John* as an argument of *arrested* and *intelligent*, respectively, has to take place at the interfaces.

We return to the topic of A'-movement in section 5.

3.5. A note on iteration vs. recursion

The split-merge process described in (2) is iterative rather than recursive: the process successively strips off members from the numeration. However, recursion ensues as soon as the numeration includes elements that are the output of a previous derivation. This turns the tables a bit on the demonstration of recursion in natural language. Whereas the patterns in (15) are typically taken to be the output of a recursive process, they need not be, in the system contemplated here:

(15) a. John thinks that Mary said that Bill hopes ... etc.

b. the hole in the sock on the foot of the man on the bench in the park ... etc.

The reason that these may not be the output of a recursive process is that nothing prevents (15a), for instance, to be built on the (partial) numeration in (16), in which case (15a) could be derived by an iteration of split-merge.

(16) { John, thinks, that, Mary, said, that, Bill, hopes, ... }

In contrast, a simple clause like (17a) must be recursive in the sense that *the man* must be included in the numeration as a single item (as in (17b)), hence must be the output of a previous derivation.

(17) a. The man saw the woman

b. { [the man], saw, [the woman] }

The reason that *the man* in (17a) must be the output of a separate derivation is that if it were not, i.e. if the numeration were as in (18a), merge would first split off *the*, yielding the pair in (18b), in which *man saw the woman* would be a constituent, contrary to fact.[9]

(18) a. { the, man, saw, the, woman }

b. ⟨ the, { man, saw, the woman } ⟩

Recursion, then, ensues as soon as the system requires two (sub)derivations to interact. It is to the properties of such complex, layered derivations that we now turn.

9. This reasoning does not require that *the woman* in (17b) is the output of a separate derivation, because when the numeration is down to { the, woman} and *the* is split off, the residue of the numeration { woman } does make a constituent (even if it were complex).

4. Layered derivations

4.1. The composition of the numeration

The syntactic derivation described so far turns an unordered set of elements, the numeration, into an ordered structure. The question of the nature of the numeration, then, is crucial to the evaluation of the system.

I see no reason to require that the numeration be a homogeneous set. The default position appears to be that the numeration may contain all kinds of linguistic objects, including features, morphemes, words, phrases, clauses. It follows that the members of the ordered pairs created by split-merge need not have the same phrase structure status, as supported by the empirical material discussed in Ackema and Neeleman (2004, chapter 4). As Ackema and Neeleman (2004:130) state

> Not only can lexical items be inserted in syntactic terminals, but it is also possible to match a complex word with a syntactic terminal, a complex syntactic category with a morphological terminal and a complex syntactic category with a nonterminal in a different syntactic category.

Phenomena supporting Ackema and Neeleman's statement include formations like (19), taken from Bauer (1983:70), where a complex syntactic category is combined with a bound morpheme:

(19) [sit around and do nothing]-ish

In the system contemplated here, 'insertion' takes the form of split-merge as described in (2), where a single element *x* from the numeration N is split off from N, creating a pair ⟨ *x*, N–*x* ⟩, where *x* has obtained a syntactic position, and N–*x* forms the input for further steps in the derivation. From this perspective, (19) is derived from a numeration (20a) via the steps in (20b):

(20) a. { [sit around and do nothing], -ish }

b. step 1: merge *sit around and do nothing*, yielding
⟨ [sit around and do nothing], { -ish } ⟩

step 2: merge *-ish*, yielding
⟨ [sit around and do nothing], ⟨ -ish, { } ⟩ ⟩

The derivation in (20b) follows trivially out of the numeration in (20a), so that we may take Ackema and Neeleman's statement to be about the composition of the numeration in his system.[10]

10. It is required that I elucidate the relation beween syntax and derivational morphology here. Like Ackema and Neeleman (2004:182), I take 'syntax' to include the derivation of both syntactic and morphological structures. My starting point is that the mechanism of split-merge as described in this article is responsible for both syntactic and morphological structure. Since I assume in narrow

4.2. A network of derivations

The idea that the syntactic component may treat syntactically complex elements as single items for particular operations (such as merge and move) is a traditional one, implicitly adopted in most treatments, and underlying much work being done in the framework of construction grammar these days (see in particular Goldberg 2006). In the proposal advanced here, it means that certain elements in the numeration must be the output of a previous derivation. These elements, then, have a dual nature: they are complex in the sense that they have been derived in a previous derivation, but they are single items in that they are listed as atoms in the numeration for a subsequent derivation.

It follows, then, that syntactic derivations are (perhaps inevitably) *networks* of derivations, and I would like to suggest that certain properties of natural language are best understood as a function of this punctuated nature of a derivation.

In particular, I would like to pursue the idea that the output of each (sub)derivation is interpreted by the interface components dealing with sound and meaning before being enlisted in the numeration for the next derivation. If so, we have two criteria for deciding whether a string of elements in the output of derivation D is listed as a single item in the numeration N for D:

(21) *Criteria for determining that an element is the output of a previous derivation*

1. configurational criteria (i.e. the need to yield constituents as in (17)-(18))
2. interpretive criteria (showing interface effects, i.e. idiosyncratic sound/meaning pairings)

Interface effects may include the following:

(22) *Interface effects*

a. conventionalization (the acquisition of conventional meaning: words, compounds, idioms)
b. categorization (the determination of a syntactic category, with the possibility of reanalysis)
c. morphological realization (spelling out of features acquired in the course of a derivation)
d. interpretation (in terms of focus and discourse status)
e. atomization (creating opacity)
f. linearization (conversion of structure to linear order)

---

syntax no derivational mechanism other than split-merge, i.e. no movement of any kind, the hypothesis being pursued here is that syntax and derivational morphology are the same, and that the insertion relation linking the two in Ackema and Neeleman 2004 reduces to the process of numeration composition. In what follows, I use the simple terms ‘syntax’, ‘syntactic’, etc. in the general sense of having to do with the derivation of structure, regardless of whether the structure is intended as essentially syntactic or morphological in nature.

To illustrate, compounds are syntactically regular combinations (representable in binary branching tree structures) showing a number of the interface effects listed in (22). A compound like *baseball* has acquired a conventional meaning, so that the word as a whole refers neither to a kind of base nor to a kind of ball, but to a game of sports. Likewise, a compound like *cutthroat*, composed of a verb *cut* and a noun *throat*, acquires the category N (nominal) in an idiosyncratic way, i.e. not from one of its component parts (a cutthroat is not a kind of throat, but a person who cuts throats). Compounds are also atoms in that the parts of a compound may not be merged separately in the derivation of a larger structure in which the compound appears. In addition, compounds in many (head-initial) languages show an idiosyncratic head-final ordering (as in *lawn-mow(-er)* vs. *(to) mow (the) lawn*).

The prototypical example of a phrase showing similar interface effects is the idiom, e.g. *jack of all trades* 'person doing different kinds of work'. Idioms are conventional sound-meaning correspondences, with a fixed syntactic category, interpretation, and linear order. Moreover, extraction of subparts of an idiom generally destroys the idiomatic interpretation:

(23) #All trades, he is a jack of.

This is unexpected if idioms are not listed in the numeration as atoms.

This leaves a number of issues undiscussed. First, many idioms contain open positions, as in *pull X's leg*, suggesting that idioms may also be created 'on the fly' i.e. in the course of a larger derivation; see Svenonius (2005) for relevant discussion. Second, idioms may be enriched, as in *to kick the proverbial bucket = to kick the bucket*, 'to die'. And finally, many idioms show lack of opacity, such as *make headway* 'make progress', allowing the passive *Headway was made*. The first two properties of idioms suggest that idiosyncratic sound-meaning correspondences may persist under (slight) variation. This does not preclude creation of the idioms in a separate derivation. The problem of lack of transparency requires a more detailed investigation of the relevant cases: it may be that *make* in *make headway* is in fact not part of the idiom, but a transparent verb of effectuation, and it needs to be established whether this is generally the case with idioms that allow (some) movement of its parts. Ignoring these complications for the moment, I would like to maintain that idioms are prototypically the output of a separate derivation layer, yielding an element to be listed as an atom in the numeration for the next derivation layer.

I have argued elsewhere (Zwart 2006c) that phenomena indicative of syntactic reanalysis may also be understood as a function of the transition from one derivation layer to the next. Thus, Kajita's (1977) proposal that (24a) involves a process of reanalysis as in (24b) may be essentially maintained, on the understanding that reanalysis is an interpretive process, taking place at the interface concluding a derivation layer.

(24) a. a far from simple matter

b. [far [from [simple]]] > [[far from] simple]

In my view, *far from simple* is construed in one derivation layer as a regular binary branching structure (parallel to *far from home*), but interpreted at the interface as a single item referring to a property (a degree of simplicity). As properties are structurally realized as adjectives, *far from simple* may be listed in the numeration for the next derivation layer as an adjective.

That this may be the correct approach is supported by the observation that reanalyzed structures are treated as heads for morphology. Thus, Dutch (25a) looks like a prepositional phrase headed by *bij* 'by', but is apparently reanalyzed as an adjective meaning 'smart', and receives a suffixed adjectival inflection at its edge (25b) (CG = common gender).[11]

(25) a. bij de hand
by the hand
(literally) 'by the hand'
(idiomatically) 'smart'

b. een bij de hand-e student
a by the hand-CG student$_{CG}$
'a smart student'

Importantly, the adjectival inflection in (25b) suggests that while idioms are opaque in the context of the next derivation layer, they are not immune to processes taking place at the interfaces concluding that next derivation layer (assuming, as before, that inflectional morphology is postsyntactic). If so, evidence brought to bear on the question whether some phrase in derivation D_2 is the output of a previous derivation layer D_1 must relate to (a) effects of the interface between D_1 and D_2, and (b) syntactic effects (e.g. opacity) internal to D_2, but not interface effects at the conclusion of D_2.[12]

4.3. Lexical decomposition

Minimalist analyses take verbs (at least transitive and unergative intransitive verbs) to be composed of (at least) a lexical root and an agentive/causative element 'little *v*' (Chomsky 1995:315). Under this analysis, a verb like *dance* is the result of conflation of a root DANCE and 'little *v*' (which may be paraphrased as DO or CAUSE), so

11. As noted by Booij 2002, the final consonant of *hand* 'hand' is devoiced before it takes on the adjectival inflection, yielding [hɑntə] rather than [hɑndə], cf. PL *handen* [hɑndə] 'hands', suggesting once more that *bij de hand* has been reanalyzed as an interface effect between derivation layers.
12. This is important if, as Chomsky (2001) suggests, verb movement is the effect of reordering processes at the interface, which would then eliminate verb movement as a property disproving idiom status. In other words, phrasal verbs like Dutch *piano-spelen* [piano-play] 'play the piano' may still be considered to be the output of a previous derivation D_1, even if the verb *spelen* 'play' is separated from *piano* 'piano' as a result of the verb second rule, at the interface of the next derivation layer D_2.

that *dance* is analyzed as 'do (a) dance'.[13] The external argument of *dance* is taken to be merged to the projection of 'little *v*', i.e. is generated internal to the verb phrase complex (the VP-internal subject hypothesis), but outside the VP proper (i.e. in the specifier position of the little *v*P; cf. Kratzer 1996).

Taking the lexical decomposition to be essentially correct, and adopting the idea that derivations are layered, we may ask whether the conflation process takes place in a separate derivation layer or not. To evaluate this question, note that the lexical decomposition analysis involves at least four hypotheses:

(26) *Elements of the lexical decomposition analysis*
   a. words are decomposable into meaningful units
   b. the units are heads in syntactic configurations
   c. argument roles are defined as positions in these syntactic configurations
   d. noun phrases expressing argument roles are merged in these positions

I accept (26a-c) without further discussion. However, if it turns out that the conflation process yielding verbs takes place in a separate derivation layer, it follows that (26d) cannot be maintained. The crucial observation here is that noun phrases expressing grammatical functions like subject and object are clearly associated with argument roles of the verb, but may (perhaps: must) be syntactically realized in positions outside the verb complex.

To evaluate the status of conflation in a layered derivation approach, we must ask whether (conflated) verbs show the interface effects in (22), which, if attested, indicate that an element is the output of a separate derivation layer. The ones that apply are repeated here:

(27) *Interface effects of conflated verbs*
   a. conventionalization (the acquisition of conventional meaning: words, compounds, idioms)
   b. categorization (the determination of a syntactic category, with the possibility of reanalysis)
   c. morphological realization (spelling out of features acquired in the course of a derivation)
   d. atomization (creating opacity)

Relating to conventionalization (27a), it is clear that conflated verbs acquire a conventional sound-meaning pairing. In the words of Hale and Keyser (1993:96), "all verbs are to some extent phrasal idioms, syntactic structures that must be learned as the conventional 'names' for various dynamic events". A conflated verb is often not fully compositional in meaning. For example, if *give* is a conflation of CAUSE and HAVE, it is still the case that *give* is different from *cause to have*, as one may cause someone to have *x* without actually giving them *x*.

13. On conflation, see Hale and Keyser (2002:chapter 3).

Second, relating to categorization (27b), the roots involved in the conflation process are often taken to be acategorical (Marantz 1997, Hale and Keyser 2002:98). For example, DANCE would be a root without category, and would be conflated with an empty V-node to yield a verbal root. In Ramchand (2008), a verb is a layered complex of three component projections, expressing causation, process, and result. Although the heads of these projections may be labeled as verbs, it is not clear that they have to be. What the analysis implies, rather, is that a complex involving these elements comes to function as a verb. If this is the correct interpretation of the decomposition/conflation process, categorization applies to the output of the process.

Third, relating to morphological realization (27c), it is clear that the output of conflation is often a monomorphemic element (e.g. *dance*), suggesting a conversion process typical of Morphology at the interface.

Finally, it is striking that in the overwhelming majority of the analyses adopting the general decomposition/conflation outlook, the component parts of the analysis (such as the verbal root and 'little *v*') do not show a syntactic life of their own: the root invariably conflates with 'little *v*', even in a language like English where the lexical verb is notoriously immobile. Attempts to assign a separate auxiliary to 'little *v*' (such as *do* in *do*-support constructions) are not convincing, since such auxiliaries combine with transitive verbs, which then should involve a 'little *v*' of their own if the analysis is to remain consistent. In other words, there appears to have been a clear understanding that the conflation process yields a lexical item (e.g. Hale and Keyser 1993:94f).

I believe that these observations suffice to allow the conclusion that conflation (as commonly understood) takes place in a separate derivation layer. It is a fully regular syntactic operation, but it yields an output which passes through the interfaces and is presented as an atom for inclusion in the numeration of a next derivation layer.

In Hale and Keyser's influential work, the decomposition/conflation analysis serves the purpose of defining argument roles configurationally (cf. (26c)). This can still be maintained, but not in the sense that noun phrases are generated in the positions which define argument roles. This is because conflation yields an atom, the subparts of which play no role in the next derivation layer (i.e. cannot be merged separately). Rather, argument positions must be regarded as hierarchically organized slots (thematic roles, indices) with which the noun phrases merged in the next derivation may be associated.

If this is the correct approach, derivations can no longer be thought of as being driven by selectional requirements of verbs. This idea of selectional requirements driving a derivation has always been problematic in the face of the phenomenon of pseudotransivity: a transitive verb like *kill* cannot be said to require the presence of an internal argument, in view of utterances like *John kills for a living*. It rather seems to be the case that if a noun phrase is present, it is interpreted in relation to properties of the verb.

The idea that verbal properties drive a derivation appears to have been the main motivation for construing derivations as working from the bottom up. If the layered derivation approach is correct, selectional requirements are satisfied at the interface (noun phrases 'binding' argument roles associated with the verb), and noun phrases may be thought of as being merged in their grammatical function position. This in turn allows us to think of derivations as working from the top down.

5. Phases

The concept of a derivation layer shows significant overlap with the concept of a derivation phase as defined in Chomsky (2000, 2001, 2008). This suggests that one of the two concepts is redundant.

A phase is a subsection of a (bottom-up) derivation, built on a subset (subarray) of the relevant numeration. At the end of the phase, a division is made between the edge of the phase and the remainder of the phase. The remainder is turned over to the interfaces, is processed there, and plays no role in the further derivation, which proceeds from the edge and may remerge material contained in the edge. Chomsky (2001:12) takes CP and little *v*P to constitute phases in this sense.

Derivation layers are similar to phases in that they create opacity. They differ in (at least) two respects: the output of a derivation layer yields an element to be included in the numeration for the next derivation layer, and derivation layers lack an edge. A principled distinction between derivation layers and phases is that derivation layers are not defined in terms of syntactic category: a CP may be the output of a separate derivation layer, but it does not have to be. A little *v*P, on the other hand, is always the output of a separate derivation layer, because, as we have seen in section 4.3, a little *v*P is essentially a verb.

If we are correct in taking a little *v*P to be essentially a verb, the opacity of the phase/derivation layer *v*P reduces to the principle of lexical integrity (more correctly, the principle of lexical integrity reduces to simplest merge, barring split-merge of subparts of numeration items). Arguments showing the phase status of the *phrase* *v*P (e.g. Legate 2003) must therefore be misguided.

A CP which is not a complement must be a phase/output of a separate derivation layer for configurational reasons (cf. (21)). This yields the general opacity of subject and adjunct clauses (the Condition on Extraction Domains of Huang 1982). A complement CP does not have to be the output of a separate derivation for configurational reasons, suggesting that complement clauses are transparent by default. This makes moves creating an escape hedge for long distance movement (including the edge concept of phase theory), going back to Chomsky (1973), superfluous.[14]

Before we can examine the nature of opacity in the framework contemplated here, we must make clear what is understood by the concept of movement. In the split-merge system proposed here, movement cannot exist. Three types of movement are generally assumed: head movement, A-movement (linking argument positions

14. That is to say, there is no *principled* reason why movement should be stepwise. I am aware of empirical evidence supporting stepwise movements, but have to refrain from addressing it here.

to grammatical function positions) and A'-movement (linking grammatical function positions to operator positions). For head movement, we may follow Chomsky (2001) in taking it to be outside of narrow syntax. This means that it should not be described in terms of (split-) merge. For A-movement, I have argued in section 4.3 that it must be redefined as a top-down relation associating noun phrases in grammatical function positions with argument role properties of the verb. But A'-movement cannot be reasoned away, because elements displaced via A'-movement are syntactically active in their grammatical function position (the range of phenomena designated as involving 'reconstruction', cf. Barss 2001).

Reconstruction is illustrated in examples like (28)-(29), showing that the grammatical function position (marked by *[e]*) of the fronted constituent is relevant for determining its case (28) and binding properties (29).

(28) a. **Wen** hast du **[e]** gesehen ? (German)
who:ACC have:2SG you:NOM GAP see:PART
'Who did you see ?'

b. Hast du **ihn** gesehen ?
have:2SG you:NOM he:ACC see:PART
'Have you seen him ?'

(29) a. **Himself**, John doesn't like **[e]**

b. John doesn't like **himself**

In both (28a) and (29a) the object grammatical function position is vacated, indicated by a gap in the examples, and the fronted category behaves as if it occupies the vacated position. In a bottom-up derivation, this can be understood if the grammatical properties of the fronted element are established at the point in the derivation where it merges in its grammatical function position, and are not affected by further movement at a later point in the derivation. If the top-down derivation contemplated here is to remain viable, a new proposal for the description of A'-movement is needed, doing justice to the reconstruction property in a different way.

Opacity for A-movement is sensitive to phases/derivation layers in the following way:

(30) A noun phrase α can be associated with an argument role of a verb β only if α and β are members of the same numeration.

Ideally, the "association" referred to in (30) is a subtype of dependency as defined in section 3.2, i.e. β must be a member of the residue of the numeration after split-merge of α. In the familiar movement terminology, (30) excludes A-movement out of subject clauses and adjunct clauses, while allowing movement out of complement clauses. A-movement out of complement clauses is furthermore typically blocked by [tense] or an intervening subject. It is unclear at this point how that should follow. Perhaps Ura (1994) is right, and the opacity induced by [tense] and/or subject is not absolute. Alternatively, these additional locality conditions on A-movement are not configurational, and should be understood in the same way that conditions on anaphor binding are (Chomsky 1982, 1986).[15]

A'-movement out of complement clauses is blocked by various factors, which often affect extraction of adjuncts more severely than extraction of arguments. Needless to say that this is a vast domain of research, too complicated to do justice here. I would like to assume here that the facts are not murky, and that movement of a wh-element out of a complement clause is rendered impossible when the specifier position of the embedded CP is filled by another wh-element, as in (31).

(31) *Who did you wonder why they arrested ?

The question at hand is: can we describe the process of A'-movement in such a way that the ungrammaticality of (31) follows more or less automatically?

We thus face two questions:

(32) *Questions regarding A'-dependency*

a. How can we understand the fact that an A'-moved element is syntactically active in its grammatical function position?

b. How can we describe opacity effects in A'-movement?

I think the reconstruction facts may be taken to indicate that A'-movement phenomena invariably involve two syntactic elements: a fronted category and its trace. In a top-down approach to movement, these two elements must be merged separately. It follows that both elements must be in the numeration, and must be available for independent split-merge operations. Yet the presence of the two elements in the numeration must not be random, as the fronted element requires the presence of a trace, and vice versa.

The cooccurrence restriction on fronted elements and their trace can be illustrated in Dutch, where the neuter wh-pronoun *wat* is interpreted as meaning 'something' when occurring in a grammatical function position (33a), and as an operator binding a variable when fronted (33b):

(33) a. Ik zie wat (Dutch)
I:NOM see:1SG what
'I see something.' (I see some *x*)

b. Wat zie ik ?
what see:1SG I:NOM
'What do I see ?' (What *x* [is such that] I see *x*)

15. I am assuming here that binding conditions relate to the concept of a proposition, with the antecedent occupying a grammatical function position in the syntactic expression of the proposition.

One way to capture the different uses of *wat* 'what' in (33a,b) would be to say that the numeration for (33a) contains the simple lexical item *wat*, while the numeration for (33b) contains a tandem of elements: *wat* in conjunction with an empty variable. Thus, *wat* in (33b) is a different lexical item from *wat* in (33a), and the difference is that operator *wat* in (33b) is what we might call a 'double atom' consisting of a lexical item and a variable.

Movement can now be described as the independent merger of the parts of a double atom, one part realizing an operator position, and the other part (the variable) realizing a grammatical function position. To account for the reconstruction effects, one would have to assume that the two members of a double atom remain linked to each other, such that they keep having identical feature specifications also after merger (and keep updating them in identical ways).

Needless to say that this is all very tentative, and it is too early to claim that the proposal has sufficient substance to salvage top-down derivations. Other approaches, mimicking movement, might also be pursued, but at this point I am only considering approaches that adhere to the simplicity requirements in (1).

Opacity effects now follow if we state:

(34) Members of an operator-variable pair are merged in a single derivation layer.

(34) need not be stipulated, since members of an operator-variable pair are members of a numeration, and all members of a numeration are ordered by iterative pair-merge in a single derivation layer.

The question of opacity, then, again reduces to the question of how separate derivation layers are established. Consider the contrast in (35).

(35) a. Who do you think they arrested [e] ?

b. *Who do you wonder why they arrested [e] ?

The facts follow if *they arrested* in (35a) is not composed in a separate derivation layer, and *why they arrested* in (35b) is. Recall that there is no configurational reason why a complement clause must be the output of a separate derivation (unlike with subject clauses and adjunct clauses). If the distinction between declarative and interrogative complement clauses is correctly characterized as one involving derivation layering, the distinction must reside in properties internal to the two types of clauses.

In this connection, it is perhaps relevant to note that interrogative phrases at the edge of an embedded clause (such as *why* in (35b)) do not interact with interrogative phrases in the matrix clause (Aoun, Hornstein, and Sportiche 1980; cf. Lasnik and Saito 1984:237). Compare:

(36) a. Who bought what ?

b. Who knows what John bought ?

c. Who knows that John bought what ?

The observation is that the scope of *what* interacts with the scope of *who* in (36a) and (36c), but not in (36b) (scope interaction is evidenced by the possibility of a pair-list answer: Bill [knows that John] bought a book, Mary [knows that John] bought a car, etc.). Naive application of the phase-based approach to opacity fails to predict the lack of interaction between *who* and *what* in (36b), as *what* is in the edge of the embedded clause and should be 'visible' to *who* in the same way that it is in (36a).

The facts follow, however, if embedded interrogatives are the output of a separate derivation layer. We have assumed that the output of a derivation layer is processed by the interface components, fixing (aspects of) its meaning. The fact that the scope of *what* in (36b) is 'frozen' is then just what one would expect as one of the interpretive interface effects of derivation layering (cf. (22d)).

This explanation should find some way of accounting for the possibility of scope interaction between *who* and *which book* in examples like (38) (cf. Chomsky 1973:283).

(38) Who remembers where we bought which book?

In (38), a pair-list answer is available for the pair *who-which book* but not for the pair *who-where*. This is impossible if embedded interrogatives are fully opaque as a result of their being generated in a separate derivation layer. Perhaps a modification is needed, then, stipulating that at the interface a selection is made among the terms of an output, singling out elements that are visible in the context of the next derivation. It needs to be determined how such a selection takes place, but that features of terms of outputs remain visible is presumably inevitable. For example, in (39), the noun phrase *which boy's mother* as a whole must be marked as an operator, even if the operator element *which* is embedded deep inside it.

(39) Which boy's mother did you insult [e] ?

Some 'percolation' of features/properties must therefore be assumed in a layered derivation approach as well.

It may be thought that, if a percolation mechanism (distinguishing between *where* and *which book* in (38)) is responsible for the absence of wide scope of elements appearing in the edge of an embedded interrogative, our assumption that embedded interrogatives are outputs of separate derivations is redundant. However, it can be shown that similar scope interactions are possible with elements inside constituents which must be the output of a separate derivation layer, such as (40):

(40) Who died after eating what ?

Here, *after eating what* is an adjunct and therefore must be the output of a separate derivation for configurational reasons. Still, a pair-list answer triggered by *who-what* is available. Likewise, a pair-list answer is available with wh-phrases contained within coordinate structure, which I suggest below must also be created in separate derivation layers:

(41) Who picked which book and which movie (as the year's best) ?

Assuming, then, that an embedded interrogative is the output of a separate derivation layer, wh-island effects follow from the general principle (34). In the split-merge approach contemplated here, there is no way that the operator and its variable can be distributed over two derivation layers. This is because the operator and the variable form a pair in the numeration, with members to be merged separately, but inevitably within a single derivation layer (as each derivation layer is fed by its own numeration).

The idea that an opaque domain is the output of a separate derivation layer may help in unifying various now unrelated opacity effects. For instance, it has been noted that backgrounded complement clauses show opacity effects (cf. the discussion in Goldberg 2006:147):

(42) a. What do you think that the mayor smoked ?

b. ?? What did it bother you that the mayor smoked ?

Since backgrounding involves prosodic adjustment, we may think that a backgrounded clause constitutes the output of a separate derivation layer, achieving its prosodic properties when passing through the interfaces (cf. (22c/d)). It would then follow that backgrounded clauses may not contain a variable associated with an operator in the matrix clause (again, because the members of an operator-variable pair cannot be distributed over separate derivation layers).[16]

Likewise, the following contrast noted by Truswell (2007) falls into place:

(43) a. What did John come in whistling ?

b. * What did John work whistling ?

In both (43a) and (43b), *whistling* is an adjunct and should be thought of as the output of a separate derivation layer (for configurational reasons, cf. (21a)), predicting equal ungrammaticality for both. The fact that (43a) is acceptable Truswell relates to the observation that *come in whistling* may be interpreted as referring to a single event, which is much harder with *work whistling*. If so, an alternative analysis is available in which *come in whistling* is in fact the output of a separate derivation layer, which is then included in the numeration for the next derivation layer, which also includes the operator-variable pair.

Observation of the properties of coordinate structures suggests that they, too, must be the output of a separate derivation layer (first proposed, to my knowledge, in Zwart 2005). A full discussion would take us too far afield here, but a simple relevant observation is that coordinate noun phrases have different properties from their individual conjuncts:

16. The fact that (42b) is not judged crashingly bad must then be attributed to independently available repair mechanisms, i.e. as a matter of performance.

(44) a. John is crazy

b. Mary is crazy

c. John and Mary are/*is crazy

This suggests that the coordinate structure passes through the interfaces where its feature make-up may be adjusted (cf. (22c)). If so, the general opacity of coordinate structures (giving rise to the Coordinate Structure Constraint of Ross 1967) follows from the fact that coordinate structures, as derivation layer outputs, are atomic.

I am well aware of the noted exceptions to the general opacity of coordinate structures, but it seems to me that in the relevant cases the derivation layers are not necessarily constituted in the same way as in standard coordination.[17] For example, in (44a) *try and finish* may be taken to refer to a single conative event, with *try* functioning as a kind of auxiliary conveying the conative modality. This suggests that *try and finish* is the output of a separate derivation here, and that the variable associated with *that* is not part of that complex element. Conversely, the string *and stay calm* in (44b) may be taken to itself constitute the output of a separate derivation, considering the fact that it functions as an adjunct paraphrasable as 'such that you stay calm'. If so, we do not expect it to interfere with the operator-variable relation.

(44) a. the thesis that he wanted to [try and finish] [e]

b. the kind of guy you can listen to [e] [and stay calm]

These examples illustrate that the layered derivation approach to opacity may be successful in accounting for patterns that require a certain derivational flexibility, having to do with the decision of organizing a derivation in stages. It seems to me that this is an advantage over the phase-based approach, where phase status is tied rigidly to categorial status.

6. Conclusion

In this article I hope to have shown the following.

First, the standard conception of the structure building operation Merge as involving a) selection of elements from a resource and b) merger of these elements into a constituent, fails to meet the most stringent simplicity requirements (cf. (1)). A simpler version of Merge (split-Merge) involves a process whereby one element x from the resource N is split off from N, yielding a pair of constituents $P = \langle x, (N-x) \rangle$. If split-Merge continues to operate on the residue of the resource $(N-x)$, we can say that the derivation involves the iterative application of a single operation until the resource is empty. At that point, all members of the resource have been assigned a syntactic position, i.e. they are members of an asymmetric sister pair.

17. See Kehler (2002) for relevant discussion, and references cited there.

Second, the emergence of crucial properties of syntactic structure (order, dependency, and interpretation) can be seen as a function of the asymmetry between the members of the pair resulting from split-Merge.

Third, the derivation is inevitably layered, in that the output of a derivation stage may be included in the resource for a next derivation stage. This yields the crucial property of natural language recursion. If this is correct, syntactic analysis should be concerned with the question of how derivations are layered, and what evidence is needed to decide that a derivational stage is concluded.

A preliminary discussion of such evidence was attempted in section 4. It should be noted at this point that if a derivation involves many layers, the bottom-up nature of the derivation is to some extent restored. For instance, if the "first-phase syntax" deriving verb phrase structure involves a separate derivation layer, as I have argued, such that its output comes to function as a verb in the next derivation layer, it necessarily precedes the creation of the functional structure associated with that verb.

The notion of a layered derivation is similar to the notion of a phase, but it fits in more naturally with a dynamic model of grammar where structure is not a priori given but derived by iterative application of a single operation. In a tentative final section, I therefore considered a layered derivation approach to opacity, suggesting that structures with transparent and opaque (complement and adjunct) clauses involve different derivation layerings. To what extent this approach may ultimately be called successful remains to be further investigated, but I believe it holds some promise. In this context I also suggested that A'-movement may be captured in a top-down approach such as the one contemplated here, if an operator-variable pair is a "double atom", the members of which (the operator and the variable) are merged separately, while keeping identical feature specifications throughout the derivation. This idea has not been sufficiently tested at this point, and the only reason for including it here is that it seems to be needed in a unified top-down derivation adhering to the simplicity requirements in (1).

References

Ackema, Peter; Neeleman, Ad (2004). *Beyond morphology: interface conditions on word formation*. Oxford: Oxford University Press.

Aoun, Josef; Hornstein, Norbert; Sportiche, Dominique (1980). "Some aspects of wide-scope quantification." *Journal of Linguistic Research* 1.3: 69-95.

Barss, Andrew (2001). "Syntactic reconstruction effects." In: Baltin, Mark; Collins, Chris (eds.), *The handbook of contemporary syntactic theory*. Malden, Blackwell, pp. 670-696.

Bauer, Laurie (1983). *English word formation*. Cambridge: Cambridge University Press.

Booij, Geert (2002). "The balance between storage and computation in phonology." In: Nooteboom, Sieb G.; Weerman, Fred; Wijnen, Frank (eds.), *Storage and computation in the language faculty*. Dordrecht: Kluwer, pp. 115-138.

Chesi, Christiano (2007). "An introduction to phase-based minimalist grammars: why *move* is top-down and from left-to-right." *CISCL Working Papers on Language and Cognition 1 (Studies in Linguistics)*: 38-75. http://www.ciscl.unisi.it/doc/doc_pub/STiL-2007-vol1.pdf

Chomsky, Noam (1973). “Conditions on transformations.” In: Anderson, Stephen R.; Kiparsky, Paul (eds.), *A Festschrift for Morris Halle*. New York: Holt, Rinehart & Winston, pp. 232-286.

Chomsky, Noam (1982). *Some concepts and consequences of the theory of Government and Binding*. Cambridge, MA: MIT Press.

Chomsky, Noam (1986). *Knowledge of language: its nature, origin, and use*. New York: Praeger.

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

Chomsky, Noam (2000). “Minimalist inquiries: the framework.” In: Martin, Roger; Michaels, David; Uriagereka, Juan (eds.), *Step by step: essays on minimalist syntax in honor of Howard Lasnik*. Cambridge, MA: MIT Press, pp. 89-155.

Chomsky, Noam (2001). “Derivation by phase.” In: Kenstowicz, Michael (ed.), *Ken Hale: a life in language*. Cambridge, MA: MIT Press, pp. 1-52.

Chomsky, Noam (2008). “On phases.” In: Friedin, Robert; Otero, Carlos P.; Zubizarreta, Maria Luisa (eds.), *Foundational issues in linguistic theory: essays in honor of Jean-Roger Vergnaud*. Cambridge, MA; MIT Press, pp. 133-166.

Epstein, Samuel D.; Groat, Erich M.; Kawashima, Ruriko; Kitahara, Hisatsugu (1998). *A derivational approach to syntactic relations*. New York: Oxford University Press.

Fortuny, Jordi (2008). *The emergence of order in syntax*. Amsterdam: John Benjamins.

Goldberg, Adele (2006). *Constructions at work: the nature of generalization in language*. Oxford: Oxford University Press.

Hale, Ken; Keyser, Samuel J. (1993). “On argument structure and the lexical expression of syntactic relations.” In: Hale, Ken; Keyser, Samuel J. (eds.), *The view from Building 20: essays in linguistics in honor of Sylvain Bromberger*. Cambridge, MA: MIT Press, pp. 53-109.

Hale, Ken; Keyser, Samuel J. (2002). *Prolegomenon to a theory of argument structure*. Cambridge, MA: MIT Press.

Huang, C.-T. James (1982). *Logical relations in Chinese and the theory of grammar*. Massachusetts Institute of Technology, doctoral dissertation.

Jaspers, Dany (1998). “Categories and recursion.” *Interface* 12.2: 81-112.

Kajita, Masaru (1977). “Towards a dynamic model of syntax.” *Studies in English linguistics* 5: 44-66.

Kayne, Richard S. (1994). *The antisymmetry of syntax*. Cambridge, MA: MIT Press.

Kehler, Andrew (2002). *Coherence, reference, and the theory of grammar*. Stanford, CA: CSLI Publications.

Kratzer, Angelika (1996). “Severing the external argument from its verb.” In: Rooryck, Johan; Zaring, Laurie (eds.), *Phrase structure and the Lexicon*. Dordrecht: Kluwer, pp. 109-137.

Kuratowski, Casimir (1921). “Sur la notion de l’ordre dans la théorie des ensembles.” *Fundamenta Mathematicae* 2: 161-171. http://matwbn.icm.edu.pl/ksiazki/fm/fm2/fm2122.pdf

Langendoen, D. Terence (2003), “Merge.” In: Carnie, Andrew; Harley, Heidi; Willie, MaryAnn (eds.), *Formal approaches to function in grammar: in honor of Eloise Jelinek*. Amsterdam: John Benjamins, pp. 307-318.

Lasnik, Howard; Saito, Mamoru (1984). “On the nature of proper government.” *Linguistic Inquiry* 15.2: 235-289.

Legate, Julie Anne (2003). "Some interface properties of the phase." *Linguistic Inquiry* 34.3: 506-516.

Marantz, Alec (1997). "No escape from syntax: don't try morphological analysis in the privacy of your own Lexicon." *Penn Working Papers in Linguistics* 4.2: 201-225.

Phillips, Colin (2003). "Linear order and constituency." *Linguistic Inquiry* 34.1: 37-90.

Putnam, Michael; Stroik, Thomas (2008). "Syntactic relations in Survive-Minimalism." Carson-Newman College and University of Missouri-Kansas, unpublished manuscript.

Ramchand, Gillian (2008). *Verb meaning and the Lexicon: a first phase syntax*. Cambridge: Cambridge University Press.

Ross, John Robert (1967). *Constraints on variables in syntax*. Massachusetts Institute of Technology, doctoral dissertation.

Svenonius, Peter (2005). "Extending the Extension Condition to discontinuous idioms." *Linguistic Variation Yearbook* 5: 227-263.

Truswell, Robert (2007). *Locality of wh-movement and the individuation of events*. University College London, doctoral dissertation.

Ura, Hiroyuki (1994). "Varieties of raising and the feature-based bare phrase structure theory." *MIT Occasional Papers in Linguistics* 7.

Zwart, Jan-Wouter (1993). *Dutch syntax: a minimalist approach*. University of Groningen, doctoral dissertation.

Zwart, Jan-Wouter (2003). "What's in a name? Syntactic and asyntactic accentuation in Dutch." In: Delsing, Lars-Olof; Falk, Cecilia; Josefsson, Gunlög; Sigurdsson, Hálldor Á. (eds.), *Grammatik i fokus/Grammar in focus: Festschrift for Christer Platzack 18 november 2003*. Lund: Department of Scandinavian Languages, vol. 2, pp. 395-401.

Zwart, Jan-Wouter (2005). "The coordinate structure constraint: a minimalist perspective." Paper presented at the Workshop on verb clusters and coordination, Leiden, November 4.

Zwart, Jan-Wouter (2006a). "Complementizer agreement and dependency marking typology." *Leiden Working Papers in Linguistics* 3.2: 53-72.
http://www.hum.leiden.edu/lucl/research/lwpl/lwpl_32.jsp

Zwart, Jan-Wouter (2006b). "Local agreement." In: Boeckx, Cedric (ed.), *Agreement systems*. Amsterdam: John Benjamins, pp. 317-339.

Zwart, Jan-Wouter (2006c). "Over het enten van interpolaties." *Tabu* 35.3/4: 163-180.

Zwart, Jan-Wouter (2006d). "Baker's Generalization in a derivational theory of binding." University of Groningen, unpublished manuscript. http://www.let.rug.nl/zwart/docs/bindingproc.pdf

Zwart, Jan-Wouter (2006e). "Structural case and dependency marking: a neo-Jakobsonian view." *Proceedings from the Parasessions of the forty-second meeting of the Chicago Linguistic Society, volume 42.2*: 77-92.