A parsimonious method for generating syntactic structure[1]

D. P. Medeiros, University of Arizona. Dec. 2020

Abstract

This paper reformulates (External) Merge as freely generating bare *n*-ary trees, labeled with a universal hierarchy by postorder traversal, and linearized by preorder traversal. Important word order universals follow: in several domains, all attested neutral orders are base-generated; unattested orders match a systematic gap in generative capacity. The framework unifies Universal 20 (Greenberg 1963, Cinque 2005) and the Final Over Final Condition (Holmberg 2000, Sheehan *et al* 2017) as consequences. We also find simple analyses of cross-serial dependency constructions, including English Affix-Hopping (Chomsky 1957), and Dutch (Bresnan *et al* 1982) cross-serial subject-verb dependencies.

Keywords: Merge, Final-Over-Final Condition, Universal 20, Cross-serial dependencies

1. Introduction

Chomsky describes the discrete infinite character of human syntax in terms of an abstract operation Merge. Merge takes as input lexical elements or syntactic objects built by other Merge operations, and produces as output a structured expression containing its inputs. There are various ways of working out the details, but in broad strokes something like Merge seems indispensable in a generative model of syntax.

Attention has focused on implementing Merge as bare set formation, which readily provides for a rich theory of syntactic structure. That implementation, whatever its successes and *a priori* appeal,[2] is not the only conceivable possibility. If another reasonable implementation of Merge makes different predictions about syntactic phenomenology, the alternatives should be evaluated by their empirical successes in addition to their conceptual properties.[3]

[1] I would like to thank the following individuals for providing feedback on an earlier draft: David Adger, Tom Bever, Noam Chomsky, and Guglielmo Cinque.

[2] There are several reasons to prefer a set-based implementation for Merge. One is the same reason that set theory is chosen as an axiomatic basis for mathematics: it is maximally conceptually sparse. Another reason is that sets are unordered, and semantic composition can be described in terms that eschew linear ordering. But see fn.6.

[3] Consider the choice between the real numbers and complex numbers for modeling physical phenomena. The reals seem conceptually inevitable, and a strict subset of the complex numbers. However, complex numbers provide a better basis for understanding phenomena like electromagnetism, and with their greater complexity comes mathematical beauty (*e.g.*, in the context of the Fundamental Theorem of Algebra).

In recent years, Chomsky has highlighted the need for syntactic theories to provide a basis for the duality of semantics: the existence, in natural language expressions, of two layers of meaning. One layer of meaning is the information-neutral thematic structure, including predicate-argument structure and selectional relations. Another layer of meaning concerns operator-variable structure, topic and focus, and the like. This cut should be tied to some syntactic distinction, such as a distinction in how Merge applies. If Merge joins disjoint syntactic objects, it is External Merge (EM). Where Merge applies to an object and one of its subparts, we have Internal Merge (IM).

> "The two types of Merge correlate well with the duality of semantics that has been studied from various points of view over the years. EM yields generalized argument structure, and IM all other semantic properties: discourse-related and scopal properties. The correlation is close, and might turn out to be perfect if enough were understood." (Chomsky 2007: 10)

The assumption of a universal ordering of EM is an essential component of the so-called cartographic program (Rizzi 1997, Cinque 1999), there realized in terms of hierarchies dictating order of insertion of lexical items into a bottom-up derivation. IM operations interleave with EM, (ultimately) yielding displacement. If EM applies in a common order, and syntactic structures are linearized the same way across languages (Kayne 1994), it follows that IM must be involved in deriving word order variation.

But languages plainly vary in word order, even in information-neutral contexts. Information-neutral contexts, by definition, do not involve discourse or scopal properties. So what drives Internal Merge in the derivation of neutral orders? Moreover, how can we explain the constraints on possible and impossible neutral word orders?

2. Generating Universal 20

As an example, consider possible and impossible neutral orders in the noun phrase, as described in Greenberg's Universal 20.

> “When any or all of the items (demonstrative, numeral, and descriptive adjective) precede the noun, they are always found in that order. If they follow, the order is either the same or its exact opposite.” (Greenberg 1963: 87)

According to Cinque's (2005) analysis, 14 of the 24 logically possible orders of these four elements are attested. Cinque shows that this pattern can be succinctly described within the EM and IM framework by assuming a universal underlying base, built by a uniform sequence of External Merge operations, together with phrasal movement but excluding head movement and remnant movement (*i.e.* any movement in the noun phrase must affect the noun, possibly pied-piping dominating structure).[4]

Cinque's analysis captures important facts: not just the possible and impossible nominal orders,[5] but their derivation as well, hence their bracketed structure. Any purported improvement on this account should preserve these descriptive successes, while either capturing additional empirical facts, or simplifying the theoretical apparatus.

It turns out that the same array of orders, and their associated bracketed structure, admit a method of generation that appears simpler than Cinque's account (or that of Abels & Neeleman 2012, Steddy & Samek-Lodovici 2011, or related analyses). This method involves free generation of *n*-ary branching structure[6] imposed on an arbitrary string of formatives, closely following Chomsky's assertion that Merge applies freely. Yet the account generates all and only the attested orders and appropriate bracketed structures, and once the bracketing is fixed in any of the legal ways, the assignment of hierarchy to the elements follows uniquely. This is an unexpected result, but notable in its simplicity. Here is the procedure:

(1) *Generative procedure over strings*

a. Start with a string of unidentified formatives.

x x x x

b. Place a left bracket just before each formative.

[x **[**x **[**x **[**x

c. Place a matching number of right brackets to form a *legal bracketing*.[7]

[4] Cinque adopts Kayne's (1994) Linear Correspondence Axiom (LCA), which requires extra structure to provide landing sites for movement. Abels & Neeleman (2012) argue that the LCA is unneeded; the relevant constraint is simply that movement is leftward.

[5] See Dryer (2018) for a different assessment of the typological facts, allowing some orders Cinque (2005) excludes, and explaining the pattern in quite a different way. The present account assumes Cinque’s typology is accurate.

[6] The *n*-ary branching structure in question is a tree with linear order; put another way, this version of External Merge produces an ordered tuple of its operands. This loses the competition with set-based Merge for mathematical simplicity. But allowing serial order within syntax-internal representations plausibly draws on capacity other animals possess.

[7] The running left-to-right total of right brackets cannot exceed the running total of left brackets, and the totals equal at the end of the string. Strings of this form are *Dyck words*; counting Dyck words of each length gives the Catalan numbers (1, 2, 5, 14, 42, ...).

[x] [x [x] [x]]

d. Scan the string *left-to-right*, indexing right brackets in increasing order.[8]

$[x]_1\ [x\ [x]_2\ [x]_3\ ]_4$

e. Copy indices from right brackets onto formatives following the corresponding left brackets.

$[x_1\ ]_1\ [x_4\ [x_2\ ]_2\ [x_3\ ]_3\ ]_4$

The indexing encodes the relative hierarchy of the formatives (see below), and the bracketed structure is the correct surface structure bracketing. In this case, we derive (2):

(2) [1] [4 [2] [3]]

The simple procedure in (1) generates all and only the attested noun phrase word orders, and their bracketed structure. Importantly, this does not simply repackage the Cinque-style EM and IM account. In particular, identifying Merge with brackets (one pair of brackets represents the Merge of what the brackets enclose), there is a fixed number of such operations for all orders: exactly *n* for *n* formatives. In a standard framework employing External Merge and Internal Merge, for the same lexical input there are *n*-1 External Merges, and variable *k* Internal Merges. Thus, the present account finds a level of uniformity in the generation of the attested orders. This perspective also dissolves the question of what drives movement: the various attested orders are simply the base-generable structures. There is no notion of steps of movement, and no need to explain them.[9] Conversely, unattested orders are not ruled out by constraints on movement, but simply correspond to impossible bracketings; see below.

No binarity constraint applies here: brackets may enclose singletons, triples, and beyond, effectively permitting *n*-ary branching. The placement of left brackets before

[8] The long-standing convention in linguistics is to number hierarchies top down, from least to most embedded. If we followed that convention, we should apply increasing indices to right brackets in the reverse of postorder traversal order -- outside-in and right-to-left, instead of inside-out and left-to right. This leads linguists to characterize the forbidden permutation as *213, for example in the literature on verb clusters. But this conflicts with the convention in computer science and mathematics, where the PostPre permutations described here are the stack-sortable words, avoiding *231 permutations. I adopt the more general convention here, at risk of some confusion.

[9] This also means that we lose any obvious syntax-internal explanation for the relative typological frequency of different orders (for example, the harmonic orders N-Adj-Num-Dem and Dem-Num-Adj-N are the most common), on which see Cinque (2005).

each surface element, and nowhere else, differs from standard practice; linguists would expect something like [[*ab*]*c*] to be a possible structure, but that is ruled out here. This does not mean that "left-branching" structure is impossible. Rather, structure traditionally analyzed as left-branching maps to a horizontal relation between nodes, while right-branching structure comes out as a vertical relation among nodes.[10] While this formulation departs from the usual way of thinking about brackets and their relation to lexical elements, it yields the right orders and their structure at a stroke. Table 1 shows all possibilities generated with four string formatives.

Brackets	Formatives	Index rt brackets	Index formatives	Order	Nominal order
(((())))	(x(x(x(x))))	$(x(x(x(x)_1)_2)_3)_4$	$(x_4(x_3(x_2(x_1)_1)_2)_3)_4$	4321	Dem-Num-Adj-N
((()()))	(x(x(x)(x)))	$(x(x(x)_1(x)_2)_3)_4$	$(x_4(x_3(x_1)_1(x_2)_2)_3)_4$	4312	Dem-Num-N-Adj
((())())	(x(x(x))(x))	$(x(x(x)_1)_2(x)_3)_4$	$(x_4(x_2(x_1)_1)_2(x_3)_3)_4$	4213	Dem-Adj-N-Num
((()))()	(x(x(x)))(x)	$(x(x(x)_1)_2)_3(x)_4$	$(x_3(x_2(x_1)_1)_2)_3(x_4)_4$	3214	Num-Adj-N-Dem
(()(()))	(x(x)(x(x)))	$(x(x)_1(x(x)_2)_3)_4$	$(x_4(x_1)_1(x_3(x_2)_2)_3)_4$	4132	Dem-N-Num-Adj
(()()())	(x(x)(x)(x))	$(x(x)_1(x)_2(x)_3)_4$	$(x_4(x_1)_1(x_2)_2(x_3)_3)_4$	4123	Dem-N-Adj-Num
(()())()	(x(x)(x))(x)	$(x(x)_1(x)_2)_3(x)_4$	$(x_3(x_1)_1(x_2)_2)_3(x_4)_4$	3124	Num-N-Adj-Dem
(())(())	(x(x))(x(x))	$(x(x)_1)_2(x(x)_3)_4$	$(x_2(x_1)_1)_2(x_4(x_3)_3)_4$	2143	Adj-N-Dem-Num
(())()()	(x(x))(x)(x)	$(x(x)_1)_2(x)_3(x)_4$	$(x_2(x_1)_1)_2(x_3)_3(x_4)_4$	2134	Adj-N-Num-Dem
()((()))	(x)(x(x(x)))	$(x)_1(x(x(x)_2)_3)_4$	$(x_1)_1(x_4(x_3(x_2)_2)_3)_4$	1432	N-Dem-Num-Adj
()(()())	(x)(x(x)(x))	$(x)_1(x(x)_2(x)_3)_4$	$(x_1)_1(x_4(x_2)_2(x_3)_3)_4$	1423	N-Dem-Adj-Num
()(())()	(x)(x(x))(x)	$(x)_1(x(x)_2)_3(x)_4$	$(x_1)_1(x_3(x_2)_2)_3(x_4)_4$	1324	N-Num-Adj-Dem
()()(())	(x)(x)(x(x))	$(x)_1(x)_2(x(x)_3)_4$	$(x_1)_1(x_2)_2(x_4(x_3)_3)_4$	1243	N-Adj-Dem-Num
()()()()	(x)(x)(x)(x)	$(x)_1(x)_2(x)_3(x)_4$	$(x_1)_1(x_2)_2(x_3)_3(x_4)_4$	1234	N-Adj-Num-Dem

Table 1: From free bracketing to word orders. Left to right, columns show legal bracketings; with formatives included; with right brackets indexed; with formatives indexed; as a hierarchically numbered surface form; and exemplified as a noun phrase order. These are the attested orders, according to Cinque (2005).

4 A closer look at the details

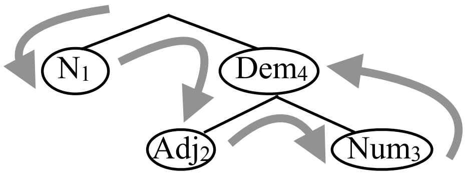
This section explores selected aspects of the account in greater depth. This includes describing the architecture in terms of trees and tree traversal algorithms, showing how the brackets for nominal orders correspond to Cinque's derivations, and examining how the account excludes unattested orders.

[10] A question for future research is whether the asymmetry between X-Y and Y-X orders predicted here can be aligned with Wagner's (2005) observations about prosodic asymmetries correlated with order of predicates and arguments, and modifiers and heads.

4.1 The procedure in terms of tree traversals

The procedure (1) equates to hierarchization (*i.e.*, labeling) of trees by postorder traversal, and linearization by preorder traversal. Postorder traversal visits nodes in the tree top-down, and right to left. To illustrate, (2) shows 1423 nominal order (N-Dem-Adj-Num) in tree form. The direction of postorder traversal is indicated by large grey arrows; subscript indices record the order in which the nodes are visited.

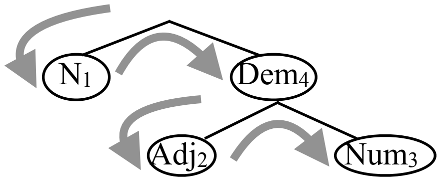
(2) *Postorder traversal*



As shown, postorder indexing allows the nodes to be mapped to a linear representation of the underlying syntax; in this example, we take the elements of the Universal 20 hierarchy bottom-up. (See section 5 for refinements in this linear hierarchy.)

Once the tree has been hierarchized this way, linear order is read off by preorder traversal, which goes top down, left-to-right. The path of preorder traversal is shown with grey arrows in (3); this path visits the nodes in surface order, N-Dem-Adj-Num.

(3) *Preorder traversal*



Note that the notion of tree here is the computer science data structure, which differs from traditional syntactic trees. Words are associated with all nodes; there are no non-terminal nodes. Figure 1 summarizes the generative action of this system on trees.

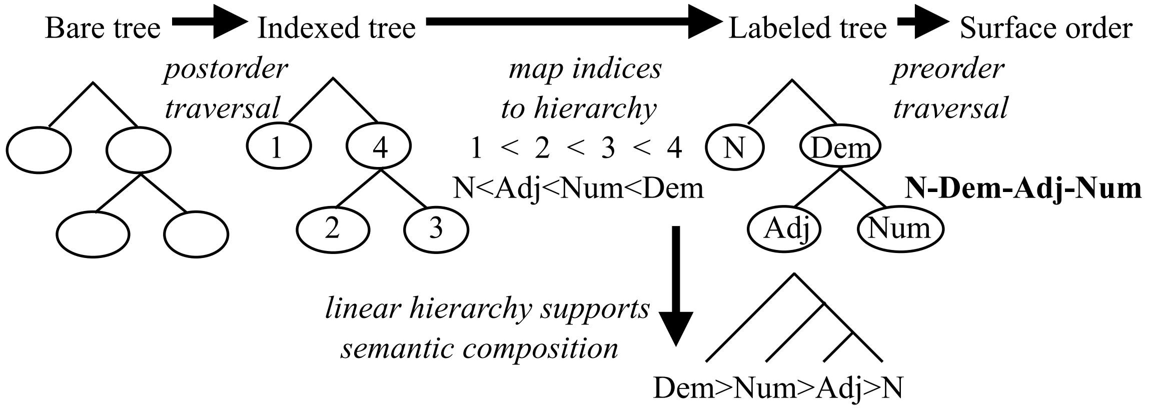


Figure 1: *Generating N-Dem-Adj-Num (1423) order*
Free Merge builds a bare *n*-ary branching tree. Postorder traversal indexes each node. Indices are mapped to a universal syntactic sequence (in this case, the hierarchy for Universal 20), yielding lexical labels on nodes. Preorder traversal of the labeled tree gives surface order; here, N-Dem-Adj-Num. Separately, the linear representation of syntactic hierarchy supports semantic composition in the familiar bottom-up order.

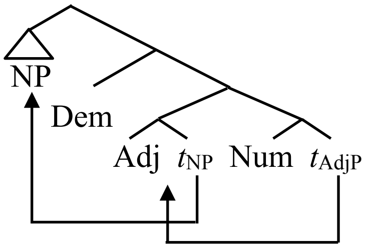
4.2 Correspondence with traditional bracketed representations

Returning to bracketed strings, the bracketing generated in this account closely matches that in Cinque's derivations. To illustrate the correspondence, we continue with the example of *1423* order. Translating to the Universal 20 hierarchy, the structure is (5).

(5) [N] [Dem [Adj] [Num]]

Illustrated below is a (simplified) Cinque-style derivation of this order.

(6)



In this derivation, two movements occur: the [Adj-NP] complex moves to precede the Num element, followed by subextraction of the NP to a specifier position before Dem. The bracketed expression (7) represents the resulting structure:

(7) [[NP] [Dem [[Adj *t*] [Num *t*]]]]

Keeping only bracket pairs where the left bracket immediately precedes a lexical element (within the NP as well, i.e. NP ~ [N]), and ignoring traces, we get (8):

(8) [N] [Dem [Adj] [Num]]

As claimed, (8) is identical to expression (5) derived by the generative procedure in (1).

4.3 Unattested orders require impossible bracketing

Consider in more detail how unattested orders are ruled out. With a hierarchy of just three elements (say, N=1, Adj=2, Dem=3), there are six logically possible word orders; five are attested as neutral noun phrase orders. One permutation, *231 (*Adj-Dem-N, usually described as *213 according to linguists' conventions; see fn. 8), does not occur as a basic noun phrase order. The present proposal explains this systematic gap.

Since left brackets occur immediately before each surface element, and nowhere else, we can begin to fill in what a *231 order would look like as a bracketed string.

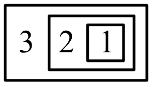
(9) $[_2\, 2 \dots [_3\, 3 \dots [_1\, 1 \dots$

Right brackets are indexed left-to-right, so they occur in the sequence $]_1 \dots]_2 \dots]_3$. Furthermore, right brackets follow the left bracket and element they match. Therefore, the entire sequence of right brackets must follow the element 1. This gives us:

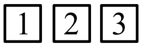
(10) $[_2\, 2\ [_3\, 3\ [_1\, 1\ ]_1\ ]_2\ ]_3$

This is not a legal bracketing; the boundaries of bracketings 1 and 2 cross. To clarify this point, we can think of brackets as denoting the edges of "boxes". In generated orders, any pair of boxes may be in a containment relation, or be disjoint; they cannot overlap partially. Illustrating with 321 and 123 order and appropriate bracketing:

(11) $[_3\, 3\ [_2\, 2\ [_1\, 1\ ]_1\ ]_2\ ]_3$

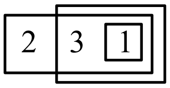


(12) $[\ 1\ ]_1\ [_2\, 2\ ]_2\ [_3\, 3\ ]_3$



But the unattested *231 order entails overlapping boxes:

(13) $[_2\, 2\ [_3\, 3\ [_1\, 1\ ]_1\ ]_2\ ]_3$



Given the way procedure (1) works (bracket an undifferentiated string, and the relative hierarchy of word order positions follows), unattested *231 order cannot be generated. Instead, the relevant bracketing must form a 321 order; bracketing determines hierarchy.

5.0 Generating the Final-Over-Final Condition

A crucial aspect of the explanation of Universal 20 here is the particular way in which the nominal hierarchy is mapped to freely-generated branching structure. This includes not just the choice of post-order traversal, one of several standard tree traversal algorithms,[11]

[11] The bracket-indexing scheme here, together with the usual way of reading linear order, generates what Feil *et al* (2005) call the *PostPre permutations*: trees labeled by postorder traversal, and read by preorder traversal.

but also how to compress linguistic hierarchical relations into a consistent linear order that can be mapped to the sequence of nodes visited. In this regard, it is notable that fixed relations among syntactic elements seem to come in (at least[12]) two flavors: roughly, selection and adjunction, or head-complement and head-adjunct relations.

Postorder traversal visits nodes/right brackets inside-out, left-to-right. It is natural to assign indices in the same order: the innermost leftmost right bracket/node is 1, the next is 2, etc. We define the hierarchical ordering relation '<' in the usual way with respect to this indexing of the traversal sequence; for example, $1 < 2$.

In these terms, I propose that a head H and its adjunct A are mapped to this sequence such that $H < A$. That corresponds to a traditional tree structure in which the head is more deeply embedded than its adjunct, a familiar analysis.

If H has several adjuncts A_1, A_2, with A_1 the closest in traditional representations, we will have $H < A_1 < A_2$. Restricting attention to a hierarchy comprised of a head and a series of adjuncts to that head, we will find *231-avoidance: *A_1-A_2-H . This pattern is seen in Cinque's version of Universal 20, and arguably in verb clusters.[13]

What about the other kind of relation, between heads and their complements? In standard analyses, heads and complements are in a symmetric hierarchical relationship. The present account provides no basis for such symmetry, and we must make a choice: heads must be hierarchically above, or below, their complements (because we are mapping syntactic hierarchy onto the necessarily-linear tree traversal sequence).

Suppose that head-complement relations obey the same $H < X$ convention: head H and complement C are mapped to the post-order traversal index sequence such that $H < C$.[14] This will produce the basic phenomenology of the Final-Over-Final Condition (FOFC; Sheehan *et al* 2017) in structures characterized by head-complement relations.

To see this, consider a configuration with nested complementation: head H_a takes a complement headed by H_b, which in turn has complement C. The hierarchical order is then $H_a < H_b < C$, and the forbidden permutation is *H_b-C-H_a. That banned order is traditionally described as a head-final phrase (H_aP) dominating a head-initial phrase (H_bP), exactly the configuration ruled out by FOFC.

For example, if head Aux takes a complement headed by V, and V has its own complement Obj, the hierarchy is Aux < V < Obj. We correctly exclude the unattested

[12] Additional stipulations appear required to model conjunction, set aside here.

[13] Though see Salzmann (2019) on attested 213 (for us, 231) verb clusters. And see Abels (2016) on Universal 20 effects in other domains, including among verbs and arguments.

[14] While the H < Adjunct hierarchical relation reflects traditional analyses, breaking head-complement symmetry this way is a stipulation. On the other hand, because the relations are ordered the same way, the distinction between head-adjunct and head-complement relations collapses (cf. Abels 2016's notion of "satellite"); both obey the condition $H < X$.

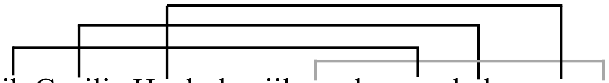
*231 order *V-Obj-Aux. Since the reasoning is about heads and complements (rather than just verbs and auxiliaries), we expect this to generalize to any partial hierarchy characterized by head-complement relations, reconstructing core predictions of FOFC.

What about structures with both adjuncts and complements? Sheehan (2017) argues that FOFC extends to certain adjunct relations. Concretely, parallel to the FOFC effect *V-Obj-Aux, *V-Adv-Aux is unattested. A full discussion is put aside, but note that this effect is correctly predicted here. This follows from the already-assumed hierarchical sequence, Aux < V < Adv; unattested *V-Adv-Aux is the forbidden *231 permutation.

In existing models of syntactic combination, complements are the closest element to the head. Adjuncts are further away (within or Chomsky-adjoined to the same phrase, or introduced by further functional superstructure). Essentially the same relation is encoded by the present ordering, H < Comp < Adjunct (*i.e.*, the complement is the unique closest element to the head). However, in the usual understanding, while the H-adjunct relation resolves structurally as one involving asymmetric hierarchy (the adjunct is "above" the head, on either theory of adjuncts), the head-complement relation is famously, and problematically, symmetric. The present approach avoids this unwanted symmetry (by stipulation), with promising consequences for word order constraints.

6.0 Generating some well-known crossing dependencies

Bresnan *et al* (1982) discuss unbounded crossing subject-verb dependencies in Dutch. Example (14), taken from Steedman (2000: 25), illustrates:



(14) ...omdat ik Cecilia Henk de nijlpaarden zag helpen voeren
...because I Cecilia Henk the hippos saw help feed
'...because I saw Cecilia help Henk feed the hippos'

Shieber (1985) discusses similar facts in Swiss German, which also exhibits long-distance cross-serial case dependencies. Interestingly, the system already established can base-generate these orders.[15] I assume the example above contains the categories in (15), abstracting away from internal structure of the object *de nijlpaarden* 'the hippos' and segmenting a Tense suffix from inflected and non-finite verbs, even if realized as zero.

(15) ...omdat ik Cecilia Henk de nijlpaarden zag-0 help-en voer-en
C S_1 S_2 S_3 O_3 V_1 T_1 V_2 T_2 V_3 T_3

[15] Stabler (2004) discusses four different classes of cross-serial dependency constructions, with distinct formal properties. I restrict attention to the two classes in this section.

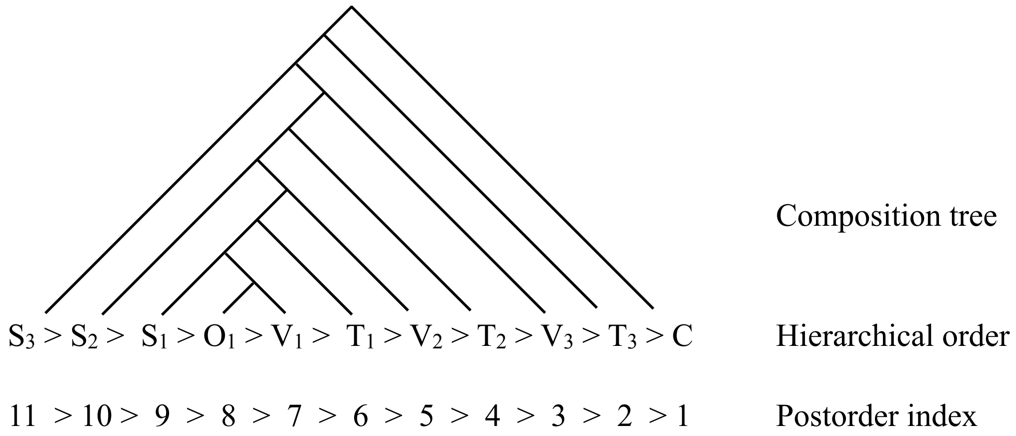
The categories in this complex sentence will be rendered as a single linear hierarchy, which we assemble incrementally. Recall that all arguments and adjuncts of a head get a higher index than the head: H < Arg; H < Adj. Where a head takes both arguments and adjuncts, I assume the relative hierarchy is H < Arg < Adv. If there are multiple arguments of a head, the complement is closest to the head: H < Comp < Arg'.

In particular, for verb head V and complement object O, V < O. The same hierarchy holds for a verb and complement clause: V < CP. A ditransitive verb would have V < DO < IO. If there is an adverbial and an object, the hierarchy is V < O < Adv.

Adding the layer for Tense and subject, the order is T < V < O < S. To align with modern proposals, we might include 'little *v*' between T and V: T < *v* < V < O < S. Since no overt morpheme obviously realizes little *v* in these examples, I omit it for simplicity. If a complementizer C is present, I assume it takes TP as complement: C < T < V < O < S.

We can now integrate the relative order of verb and complement clause with the basic clause order just elaborated.[16] For ease of reading the complex structures that follow, I display the hierarchical relations in descending order, as that yields a more familiar structure: the clause hierarchy is written S > O > V > T > C, a common surface order. For a single layer of clausal embedding, $[_{CP} \ldots [_{\underline{CP\ \ }}]\,]$, we have: $S_2 > \underline{S_1 > O_1 > V_1 > T_1} > V_2 > T_2 > C$. Replacing O_1 with one more embedded clause, we arrive at the unified three clause order (16), for the Dutch sentence (14). I show postorder indices aligned to the hierarchy string; the superposed tree indicates bottom-up semantic composition.

(16) *Integrated hierarchy for (14) with postorder index and composition tree*

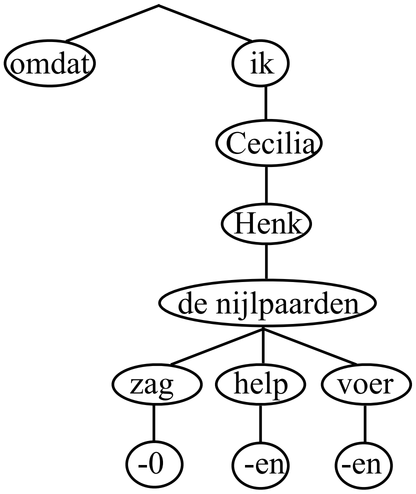


[16] At least for these structures, we are implicitly developing a simple account of recursion by substitution. I leave fuller consideration of recursion in other domains to future work.

Given this mapping from syntactic hierarchy to post-order index sequence, we can easily recover the tree structure corresponding to the Dutch surface order,[17] shown in (17).

(17) ...omdat ik Cecilia Henk de nijlpaarden zag-0 help-en voer-en

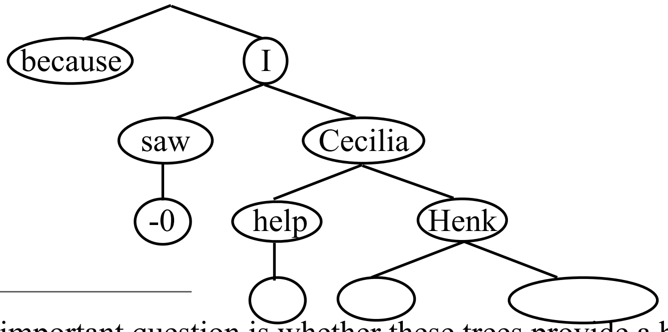
Category	C	S_3	S_2	S_1	O_1	V_3	T_3	V_2	T_2	V_1	T_1
Index	1	11	10	9	8	3	2	5	4	7	6
Brackets	( )	(	(	(	(	(	( ))	(	( ))	(	( ))))))



With the relevant syntactic hierarchy resolved as a universal linear sequence, we can readily represent other orders of the same elements, as in English in (18).

(18) ...because I saw -0 Cecilia help -0 Henk feed -0 the hippos

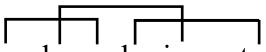
Category	C	S_3	V_3	T_3	S_2	V_2	T_2	S_1	V_1	T_1	O_1
Index	1	11	3	2	10	5	4	9	7	6	8
Brackets	( )	(	(	( ))	(	(	( ))	(	(	( ))	( ))))



[17] An important question is whether these trees provide a basis for a successful theory of prosody (see also fn. 9). While it is promising that the trees derived here correspond closely to Cinque's derivations of nominal orders, I leave this question for future work. Unlike the nominal trees, the clausal trees in this section differ from standard analyses.

-0
feed
the hippos
-0

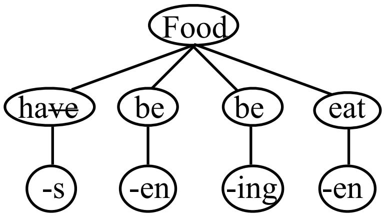
Finally, this architecture can generate the more limited pattern of crossing dependencies that arises in English Affix-Hopping (Chomsky 1957), as seen in (19).



(19) Food ha-s be-en be-ing eat-en

As Chomsky pointed out, the verbal affixes group with the preceding auxiliary in distribution and meaning, despite being separated in surface order by an intervening verb. To accommodate this pattern, suppose that an auxiliary Aux and associated displaced affix -Fx is resolved with this hierarchical order: Aux < -Fx < VP-Complement.[18] This order is indeed generated by the present account, with structure as shown in (21)[19].

(21)	Food	ha~~ve~~	-s	be	-en	be	-ing	eat	-en
Cat.	Obj	Aux$_1$	T	Aux$_2$	-Fx$_1$	Aux$_3$	-Fx$_2$	V	-Fx$_3$
Index	9	2	1	4	3	6	5	8	7
Brackets	(	(	( ))	(	( ))	(	( ))	(	( )))



7.0 Conclusion

Implementing Merge as an operation building bare ordered trees, lexicalized and linearized by traversal algorithms, we derive and unify Universal 20 and FOFC

[18] One can read the proposed hierarchy as saying that the affix (*e.g.*, *-ing*) is a head sandwiched between its selecting auxiliary (*be*) and the host verb. Or we might treat the auxiliary and associated affix as a unit for interpretation, mirroring Chomsky's (1957) analysis where the two were introduced as a single unit (*be*+*ing*) from the lexicon.

[19] It is unclear if passive movement of the object should be base-generated, or if it is obligatorily "real" movement. It is at least possible to generate with just this mechanism.

permutation avoidance patterns, and find simple analyses of constructions exhibiting cross-serial dependencies.

What is so striking is that all of these consequences follow simply from the free action of this new version of Merge, together with a single convention for hierarchical ordering: X < H. Admittedly, the new version of Merge has some moving parts: free n-ary branching to build trees, on which hierarchy is written by postorder traversal, and from which word order is read by preorder traversal. and linear order read by two standard tree traversal algorithms. While that is more machinery than set-based Merge, it also does the work of additional mechanisms and constraints on information-neutral movement, structure-building, and linearization in the standard model.

In this view, there is nothing "extra" about the displacement deriving neutral orders; the full set of typologically possible orders is base-generated. This unification of movement with structure-building goes further than Chomsky's framing of movement as Internal Merge, where Internal Merge involves one or more additional operations over and above the constant number of External Merges required to join the lexical items involved. Not so here, where exactly the same number of External Merge operations (*i.e.*, pairs of matched brackets) is involved for all neutral orders: exactly *n* such for *n* items.

That said, there is still a clear need for actual movement in the present framework, beyond the base-generation of neutral orders: effects like *wh*-movement and topic and focus displacement produce other orders.[20] Note, though, the residue of real movements under this account is the set of non-information-neutral transformations. This result aligns with Chomsky's suggestion that the duality of semantics is tied to the distinction between External Merge and Internal Merge: EM builds the base thematic structure, and IM induces discourse-information effects. That result fails to obtain on the standard view that IM derives different information-neutral orders, as in Cinque's account of Universal 20. But in this system, External Merge (the generative procedure) produces the array of neutral orders, and Internal Merge (real movement, however implemented) applies only semantically contentful transformations. That seems the proper cut.

Of course, the theory developed here is a fragment. I have not demonstrated how this system generalizes to word order in every domain, nor attempted to spell out how real movement works, nor accounted for any number of important grammatical phenomena such as coordination, ellipsis, binding, agreement, and so on. These are important topics, and much more work will be required to determine if they might find satisfying accounts within this framework.

References:

Abels, K., & Neeleman, A. (2012). Linear asymmetries and the LCA. *Syntax* 15(1), 25-74.

[20] Thanks to David Adger for discussion on this point.

Abels, K. (2016). The fundamental left–right asymmetry in the Germanic verb cluster. *The Journal of Comparative Germanic Linguistics* 19(3), 179-220.

Bresnan, J., Kaplan, R.M., Peters, S., & Zaenen, A. (1982). Cross-serial dependencies in Dutch. *Linguistic Inquiry* 13(4), 613-635.

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

Chomsky, N. (2007). Approaching UG from Below. In Hans-Martin Gärtner, H.-M., & Sauerland, U. (eds.). *Interfaces + Recursion = Language? Chomsky's Minimalism and the View from Syntax-Semantics.* Berlin: Mouton de Gruyter.

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

Cinque, G. (2005). Deriving Greenberg's universal 20 and its exceptions. *Linguistic Inquiry* 36(3), 315-332.

Dryer, M. (2018). On the order of demonstrative, numeral, adjective, and noun. *Language* 94(4), 798-833.

Feil, T., Hutson, K., & Kretchmar, R.M. (2005). Tree traversals and permutations. *Congressus Numerantium* 172, 201-221.

Greenberg, J. (1963). Some universals of grammar with particular reference to the order of meaningful elements. In Greenberg, J., (ed.), *Universals of language*, 73-113. Cambridge, MA: MIT Press.

Holmberg, A. (2000). Deriving OV order in Finnish. in Svenonius, P., (ed.), *The Derivation of VO and OV*. Philadelphia: John Benjamins.

Kayne, R. (1994). *The Antisymmetry of Syntax*. Cambridge, MA: MIT Press.

Rizzi, L. (1997). The fine structure of the left periphery. In Haegeman, L. (ed.), *Elements of Grammar: Handbook of Generative Syntax*. Dordrecht: Kluwer.

Salzmann, M. (2019). On the limits of variation in Continental West-Germanic verb clusters. Evidence from VP-stranding, extraposition and displaced morphology for the existence of clusters with 213 order. *Journal of Comparative Germanic Linguistics* 22, 55-108.

Sheehan, M., Biberauer, T., Roberts, I., & Holmberg, A. (eds.). (2017). *The Final-over-Final Condition: A syntactic universal* (Vol. 76). Cambridge, MA: MIT Press.

Sheehan, M. (2017). The final-over-final condition and adverbs. in Sheehan, M., Biberauer, T., Roberts, I., & Holmberg, A. (eds.) *The Final-over-Final Condition: A syntactic universal* (Vol 76), 97-120. Cambridge, MA: MIT Press.

Shieber, S.M. (1985). Evidence against the context-freeness of natural language. *Linguistics & Philosophy* 8(3), 333-344.

Stabler, E. (2004). Varieties of crossing dependencies: structure dependence and mild context sensitivity. *Cognitive Science* 28: 699-720

Steddy, S. & Samek-Lodovici, V. (2011). On the ungrammaticality of remnant movement in the derivation of Greenberg's universal 20. *Linguistic Inquiry* 42(3), 445-469.

Steedman, M. (2000). *The Syntactic Process*. (Vol. 24) Cambridge, MA: MIT press.

Wagner, M. (2005). Prosody and recursion. Ph.D. dissertation, MIT.