Formal Basis of a Language Universal

Miloš Stanojević
University of Edinburgh
School of Informatics
m.stanojevic@ed.ac.uk

Mark Steedman
University of Edinburgh
School of Informatics
steedman@inf.ed.ac.uk

Steedman (2020) proposes as a formal universal of natural language grammar that grammatical permutations of the kind that have given rise to transformational rules are limited to a class known to mathematicians and computer scientists as the "separable" permutations. This class of permutations is exactly the class that can be expressed in combinatory categorial grammars (CCGs). The excluded non-separable permutations do in fact seem to be absent in a number of studies of crosslinguistic variation in word order in nominal and verbal constructions.

The number of permutations that are separable grows in the number n of lexical elements in the construction as the Large Schröder Number S_{n-1}. Because that number grows much more slowly than the n! number of all permutations, this generalization is also of considerable practical interest for computational applications such as parsing and machine translation.

The present article examines the mathematical and computational origins of this restriction, and the reason it is exactly captured in CCG without the imposition of any further constraints.

1. Introduction

Permutation of elements in a construction, either within a single language or across languages, creating discontinuous semantic dependencies, is a widespread characteristic of natural languages that has motivated transformational rules (Chomsky 1957, passim), among other expressive grammar formalisms. Steedman (2020) proposes as a formal universal of natural language grammar that grammatical permutations are limited to a class known to mathematicians and computer scientists as the "separable" permutations, which are those orders over which a tree can be constructed such that all leaves descending from any node form a continuous subset $i \ldots j$ of the original ordered set $1, \ldots, n$.

The evidence for this claim is based on a number of studies of two data sets of alternations in two major constructions. The first concerns the permutations of the four

elements Det(erminer) Num(erator) Adj(ective) N(oun) that surface in that order in English noun phrases (NP) like "These five young lads," but in Yoruba and many other languages are found in the mirror-image permutation. The crosslinguistic variation in this construction has been studied by Cinque (2005) and Abels and Neeleman (2012) (among others) for languages with a single dominant order, and also within a single language with very free word order for the same construction by Nchare (2012). The second construction is the verb group, also consisting of four elements, which we can think mnemonically of as the verb-sequence "must have been dancing," whose variations with various verb-types both across and within the various Germanic languages and dialects, has been studied by Wurmbrand (2004; 2006) and Abels (2016).

In both constructions, the attested orders fall on highly skewed Zipfian power-law distributions, with the rarest orders found in only a very few cases. Such distributions make it impossible to decide whether further unattested orders are entirely excluded by hard constraints on the universal space of possible grammars, or merely disfavored by sampling error arising from soft constraints such as "Harmony" (proposed by Culbertson, Smolensky, and Legendre 2012, among others considered below) that may merely make them vanishingly rare in samples of the size that we can actually access (Abels and Neeleman 2012). Of course, a further possibility is that the unattested orders arise from a combination of hard and soft constraints.

The present study defers to such authors on the specific characteristics of the soft constraints and the distributions themselves, and considers only the nature and origin of a specific hard constraint on the formal operations that define the set of permutations that such distributions can apply to in the first place.

Both the nominal and verbal constructions concerned can be thought of in grammatical terms as made up of elements or "categories" with a "natural order of dominance" stemming from their semantics. For these constructions, that order of dominance happens in English to be the same as their left-to-right order on the page, inducing a right-branching tree-structure. However, in other languages, the same relations of dominance allow all of the word orders that would arise if the same tree were treated as a mobile, with daughters allowed to rotate around each other, as well as some word orders involving apparent displacements that go beyond rotation. While the constructions of interest allow only one natural order of dominance, other sets of categories including multiple words of the same category may allow more than one natural order to merge. (For example, the set consisting of the words "divorced, alleged, spy" has two natural orders of dominance, the other one corresponding to a distinct sense "alleged divorced spy.")

We can therefore abstract over the two constructions of interest as being made up of elements numbered $1 > 2 > 3 > 4$ according to that natural order of dominance, under the following correspondence:

(1) These > five > young > lads
     must > have > been > dancing
       1   >   2 >   3 >   4

There is widespread suspicion that not all of the four-factorial permutations of these constructions are allowed (Williams 2003; Svenonius 2007).

In considering whether a given permutation is allowed for the NP construction, Cinque (2005; 2010) emphasizes the need to exclude from consideration word orders for the NP in which the adjective is extraposed or has the markers of a relative clause. The exclusion is crucial, since such modifiers may be adjoined to the entire NP, rather than to

N, thereby defining a *different* natural order of dominance, with the extraposed element as 1. Cinque rejects a number of the orders claimed by Dryer (2018) on this ground. The present paper follows Cinque strictly concerning the attested fixed NP orders.

Cinque also proposes to exclude free word-order alternation of the kind investigated by Nchare (2012) for the Shupamem NP, on the grounds that alternating orders reflect distinctions of focus. Accounts of focus alternations have sometimes assumed the existence of an FP-like focus-phrasal external functional projection, offering an extra target for a different variety of movement resembling extraposition (Szabolcsi 1983, 1994; Giusti 2006). However, this is an additional and somewhat theory-internal assumption. English achieves similar alternations of focus or contrast in the NP by in situ accent placement without varying word order, for which Steedman (2014) offers a lexicalized alternative semantics without focus-movement or focus-projection, other than by surface-compositional syntactic derivation. We therefore depart from Cinque on this point, and provisionally accept Nchare's free orders as arising from the same mechanism as Cinque's.

The orders attested in these three studies for the NP and VP constructions can then be summarized as in Figure 1, in which the alphabetic ordering of the permutations (which will be used for reference throughout this article) follows Cinque. (Abels 2016 adopts a different convention.) Cinque (2007 n.13) notes the possibility that Figure 1(m)

		(Cinque 2005)	(Nchare 2012)	(Abels 2016)
a.	1 2 3 4	√	√	√
b.	1 2 4 3	√	√	√
c.	1 4 2 3	√	—	√
d.	4 1 2 3	√	—	√
e.	2 1 3 4	—	√	—
f.	2 1 4 3	—	√	(√)
g.	2 4 1 3	—	—	—
h.	4 2 1 3	—	—	(√)
i.	3 1 2 4	—	√	—
j.	3 1 4 2	—	—	—
k.	3 4 1 2	√	√	√
l.	4 3 1 2	√	√	√
m.	1 3 2 4	(√)	√	(√)
n.	1 3 4 2	√	√	√
o.	1 4 3 2	√	√	√
p.	4 1 3 2	√	√	√
q.	2 3 1 4	—	√	—
r.	2 3 4 1	√	√	√
s.	2 4 3 1	√	√	(√)
t.	4 2 3 1	√	√	√
u.	3 2 1 4	—	√	—
v.	3 2 4 1	—	√	—
w.	3 4 2 1	√	√	√
x.	4 3 2 1	√	√	√

Figure 1
Attested permutations.

constitutes a fifteenth order for the NP, attested for only one language so far, Dhivehi (Maldivian; Cain 2000), indicated by parentheses in the figure.[1]

We note at this point that Dryer (2018) claims several further word orders as basic for NP in fixed-order languages, in addition to those listed by Cinque. In particular, he claims (2018: pages 17, 29) that Kilivila (Senft 1986) and four other languages have order (g), which is excluded by the theory developed below, as their default order. However, Cinque argues that Senft's example of this order involves adjectival extraposition, noting that Senft (1986, pages 96) also cites the canonical order (a) for the full four-element construction itself. Of the other four languages cited by Dryer, he himself notes for Yapese (Jensen 1977) that the adjective in his example of order (g) is marked as a relative clause including the copula, hence arguably also extraposed from NP. Of the remaining three languages for which Dryer claims this order, Katu (Costello 1969, page 22) does not lexically distinguish demonstratives from locatives, but the one example Costello gives (1969, page 34 (87)) involving both an adjective and a demonstrative locative has the canonical reverse order N Adj Dem. The example given by Tryon (1967, page 60) for Dehu/Drehu includes the copula with the adjective, so is arguably also extraposed.

In the lexically multifunctional language Teop (Mosel 2017), to which Dryer also attributes (g) as base order, adjectives are expressed as adjoined adjectival phrases, with their own copies of the article or agreement and numerator, so that a strong possibility of extraposition or apposition clearly exists here also.

In terms of the CCG theory developed below, adjective extraposition requires the addition of a distinct NP-adjunct category, syntactically and semantically non-isomorphic to the standard adjective, inducing a *different* Natural Order of Dominance over the four elements, with the extraposed adjectival as 1, the demonstrative as 2, the numerator as 3, and the noun as 4. Thus, none of these languages constitutes a counterexample to the claim below that order (g) is universally excluded for the standard categories. We have strictly followed Cinque's account of the attested NP orders, and exclude several other orders listed by Dryer whose attestation appear to depend on adjective extraposition, including (h).

Of the orders for the Germanic verb group admitted by Abels (2016), he gives only qualified support for Figure 1f, h, m, and s, in some cases because they are only attested as alternates, similarly marked by parentheses.

It is striking that just two of the 24 permutations over these two constructions are entirely without even equivocal support for attestation in these three studies, namely, Figure 1g and j—that is, 2413 and its mirror-image 3142. These are the permutations that would give rise to the following word orders for the two constructions:

(2)　g.

Five	lads	these	young
$NumP|N'$	N	$NP|NumP$	$N'|N$
2	4	1	3

　　　j.

Young	these	lads	five
$N'|N$	$NP|NumP$	N	$NumP|N'$
3	1	4	2

---

1 Some of the possibilities admitted for the Shupamem NP are conditional on the presence of clitic agreement and definiteness markers, and certain orders are associated with focus effects—see Nchare (2012) chapter 3 and Steedman (2020) for details that we pass over here.

(3) g.

have	dancing	must	been
$VP_2\|VP_3$	VP_4	$VP_1\|VP_2$	$VP_3\|VP_4$
2	4	1	3

   j.

been	must	dancing	have
$VP_3\|VP_4$	$VP_1\|VP_2$	VP_4	$VP_2\|VP_3$
3	1	4	2

The above examples of permutations that in CCG are neither allowed as base orders nor as alternates for the two constructions include lexical types indicating the dominance relations among these elements, using an order-free categorial notation, in which a category of the form $X|Y$ indicates an element that combines with a sister constituent of type Y adjacent to the right or to the left to yield a constituent of type X.[2]

These categories immediately offer a hint as to a hard constraint forcing the two orders 2413 and its mirror-image 3142 to remain unattested. A formal proof is set out below, but it is obvious from inspection of (2) and (3) that these are the only two permutations in which *no category of the form $X|Y$ is adjacent to a category Y, or to a category $Y|Z$ that could yield Y as its result.*

It is also interesting that the two permutations 2413 and its mirror-image 3142 are distinguished by mathematicians and computer scientists as the only two permutations on the order 1234 that are *non-separable* (Avis and Newborn 1981; Bose, Buss, and Lubiw 1998). This observation motivated Steedman (2020) to hypothesize that for a fixed natural order of dominance of arbitrary length all permutations generated by CCG are within the set of *separable permutations*.

The main contribution of the present article is to show that this hypothesis indeed holds for sentences of arbitrary length. In addition, we also establish a close connection between CCG derivations and *separating trees*, a connection between CCG parsing and the number of paths in a lattice, and the relation of this result to the other grammar formalisms, namely, ITG (Wu 1996), TAG (Joshi 1985), MG (Stabler 1996), and CAT (Williams 2003).

The paper is structured in the following way. In Section 2 we give a short overview of CCG. In Section 3 we give a more specific definition of natural order of dominance and prove that Steedman's (2020) hypothesis is true. Section 4 shows an alternative, possibly more intuitive, way of enumerating the permutations that CCG allows, first by establishing the isomorphism between the normal forms of CCG derivations and *separating trees*, and then by showing the connection between CCG shift-reduce parsing and the Large Schröder Number of paths in a lattice on the Cartesian plane. Section 5 establishes the statistical significance of the fact that the two non-separable permutations that are *predicted* to be unattested are among the vanishingly small number of permutations that are in fact empirically unattested so far. Section 6 puts this result in the context of other grammar formalisms and their expressiveness. Appendix A gives CCG derivations for all 22 separable permutations of the attested NP and VP word orders.

---

2 This notation is isomorphic to the Minimalist Programmatic notation of Chomsky (1995b; 2001), in which the present $|Y$ is written $=Y$, and which is essentially categorial (Chomsky 1995a, 2000; Stabler 2011; Adger 2013).

2. Combinatory Categorial Grammar

Combinatory Categorial Grammar (CCG) is a radically lexicalized theory of grammar in which all language-specific syntactic and semantic information concerning word order and subcategorization is specified in lexical entries or *categories*, and is projected onto the sentences of the language by universal rules that are *combinatory*, in the sense that they apply to pairs of non-empty, strictly contiguous, categories. The latter property immediately guarantees that no CCG can recognize the unattested permutation in (2) and (3).

2.1 The Categorial Lexicon

The lexical fragment for the very common English NP order in CCG is a *directional* categorial grammar, in which all instances of $|$ are instantiated as $/$, meaning that they have to combine with an element to the right, thus:[3]

(4) These $= NP/NumP$
     five $= NumP/N'$
     young $= N'/N$
     lads $= N$

Slashes identify categories of the form X/Y and $X\backslash Y$ as *functions* taking an argument of syntactic type Y to the right or to the left, and as yielding a result of type X.

By contrast, the following lexical fragment defines the even more common mirror-image word order glossed as "Lads young five these," as required, for example, for Yoruba (Hawkins 1983, page 119).

(5) "These" $= NP\backslash NumP$
     "five" $= NumP\backslash N'$
     "young" $= N'\backslash N$
     "lads" $= N$

2.2 Rules of Function Application

The universally available rules (6) of syntactic combination called **forward** and **backward application** (respectively labeled $>$ and $<$) allow syntactic derivation from such lexicons.

(6) *The Application Rules*
     a. $X/_\star Y \quad Y \ \Rightarrow X$ $\qquad\qquad\qquad\qquad$ ($>$)
     b. $Y \quad X\backslash_\star Y \Rightarrow X$ $\qquad\qquad\qquad\qquad$ ($<$)

The type $\star$ of the slashes in $X/_\star Y$ and $X\backslash_\star Y$ limit the categories to which these rules can apply, and can be ignored for the moment.

---

3 Of course, we need further lexical categories to allow, e.g., *These young lads, Five lads*, etc., as NP. This might be done via underspecification using X-bar-theoretic features (Chomsky 1970), but we will ignore them for now.

The forward rule (6a) allows the following derivation for the English lexicon (4):

(7)

These	five	young	lads
$NP/NumP$	$NumP/N'$	N'/N	N

$$\frac{\qquad\qquad}{N'}>$$
$$\frac{\qquad\qquad}{NumP}>$$
$$\frac{\qquad\qquad}{NP}>$$

The rightward arrow $>$ on all combinations in (7) indicates that it is the rightward functional application rule (6a) that has applied in these cases.[4]

The derivation in (7) is clearly isomorphic to the standard linguistic context-free phrase structure *tree*, except that it is the right way up—that is, its leaves are at the top and its root at the bottom.

Together with the rightward and leftward application rules (6) for combination to the right and left, such categories immediately generate eight of the 24 permutations, as follows:

(8) a.

These	five	young	lads
$NP/NumP$	$NumP/N'$	N'/N	N

Very many languages

b.

These	five	lads	young
$NP/NumP$	$NumP/N'$	N	$N'\backslash N$

Many languages

n.

These	young	lads	five
$NP/NumP$	N'/N	N	$NumP\backslash N'$

Few languages

o.

These	lads	young	five
$NP/NumP$	N	$N'\backslash N$	$NumP\backslash N'$

Many languages

r.

Five	young	lads	these
$NumP/N'$	N'/N	N	$NP\backslash NumP$

Few languages

s.

Five	lads	young	these
$NumP/N'$	N	$N'\backslash N$	$NP\backslash NumP$

Many languages

w.

Young	lads	five	these
N'/N	N	$NumP\backslash N'$	$NP\backslash NumP$

Very few languages

x.

Lads	young	five	these
N	$N'\backslash N$	$NumP\backslash N'$	$NP\backslash NumP$

Very many languages

Cinque (2005) gives counts for each word order in his sample of 700 or so languages, quantized to four levels. The present counts are updated according to Cinque's 2013 sample of around 1,500 languages, with "very few" corresponding to 10–29, "few" to

---

4 Although compositional semantics and logical form are suppressed for the purposes of this article, the semantics of the rules in (6) is also the application of semantic functions such as *young*′ to arguments such as *lads*′ to yield logical forms such as *young*′ *lads*′. In general, if the functor $X \mid Y$ has logical form f and the argument Y has logical form a, then the result X always has logical form $f\ a$ (read "f of a"). Thus, semantics is "surface compositional" in CCG.

30–99, "many" to 100–299, and "very many" to 300+ (cf. Merlo 2015, Table 1). The discontinuous alpha-numeration in (8) again reflects the place of these orders in Cinque's ordering of the 24 permutations of these elements introduced earlier in Figure 1, which we take as standard. All eight orders are among those endorsed by Cinque and Nchare.[5]

The above eight orders are the ones obtained by treating the tree (7) as a mobile, allowing sisters to rotate around each other. The same eight orders are similarly available for the Germanic serial verb construction, although for these orders we have no meaningful frequency estimates:

(9) a.
$$\frac{\text{will}}{VP_1/VP_2} \ \frac{\text{help}}{VP_2/VP_3} \ \frac{\text{teach}}{VP_3/VP_4} \ \frac{\text{swim}}{VP_4}$$

b.
$$\frac{\text{will}}{VP_1/VP_2} \ \frac{\text{help}}{VP_2/VP_3} \ \frac{\text{swim}}{VP_4} \ \frac{\text{teach}}{VP_3\backslash VP_4}$$

n.
$$\frac{\text{will}}{VP_1/VP_2} \ \frac{\text{teach}}{VP_3/VP_4} \ \frac{\text{swim}}{VP_4} \ \frac{\text{help}}{VP_2\backslash VP_3}$$

o.
$$\frac{\text{will}}{VP_1/VP_2} \ \frac{\text{swim}}{VP_4} \ \frac{\text{teach}}{VP_3\backslash VP_4} \ \frac{\text{help}}{VP_2\backslash VP_3}$$

r.
$$\frac{\text{help}}{VP_2/VP_3} \ \frac{\text{teach}}{VP_3/VP_4} \ \frac{\text{swim}}{VP_4} \ \frac{\text{will}}{VP_1\backslash VP_2}$$

s.
$$\frac{\text{help}}{VP_2/VP_3} \ \frac{\text{swim}}{VP_4} \ \frac{\text{teach}}{VP_3\backslash VP_4} \ \frac{\text{will}}{VP_1\backslash VP_2}$$

w.
$$\frac{\text{teach}}{VP_3/VP_4} \ \frac{\text{swim}}{VP_4} \ \frac{\text{help}}{VP_2\backslash VP_3} \ \frac{\text{will}}{VP_1\backslash VP_2}$$

x.
$$\frac{\text{swim}}{VP_4} \ \frac{\text{teach}}{VP_3\backslash VP_4} \ \frac{\text{help}}{VP_2\backslash VP_3} \ \frac{\text{will}}{VP_1\backslash VP_2}$$

Again all of these orders are attested in Germanic, although Abel's endorsement of (s) is equivocal, on the grounds that it is only attested as an alternate in his sample.

It is striking that all possible orders based on purely applicative derivations appear to be attested for both constructions, and that all of the NP orders admitted by Cinque (2013) with "very many" or "many" attestations are among them, suggesting soft constraints favoring "harmony" or consistent directionality across categories and purely applicative derivations.[6]

However, the other 14 orders attested by the authorities in Figure 1 involve discontinuities. For example, in Figure 1c, in its instantiation as the Masai NP order "These lads five young," element 3 the adjective "young" is separated from its 4, the noun "lads" by 2, the numerator "five." How do we allow the attested orders without also allowing the unattested orders in (2) and (3)?

---

5 Cinque's (2013) study doubles the size of his sample to around 1,500 languages, and adds no new attested orders to his original 14—unsurprisingly, given the power-law distribution of the frequencies.

6 Abels and Neeleman (2012) and Abels (2016) assume that these orders are base-generated, without movement.

To do so, something more than the simple application rules is required. Cinque and others propose transformational movement as that "something more." The present approach offers an alternative to movement, or any other syntactic operation of "action at a distance" over non-contiguous elements.

2.3 Rules That Change Word Order

CCGs also include universally available rules of functional composition, strictly limited in the first-order case to the following four combinatory rules:[7]

(10) *The Harmonic Composition Rules*
    a. $X/_\diamond Y \quad Y/Z \quad \Rightarrow_{\textbf{B}} X/Z$ $(>\textbf{B})$
    b. $Y\backslash Z \quad X\backslash_\diamond Y \quad \Rightarrow_{\textbf{B}} X\backslash Z$ $(<\textbf{B})$

(11) *The Crossing Composition Rules*
    a. $X/_\times Y \quad Y\backslash Z \quad \Rightarrow_{\textbf{B}_\times} X\backslash Z$ $(>\textbf{B}_\times)$
    b. $Y/Z \quad X\backslash_\times Y \quad \Rightarrow_{\textbf{B}_\times} X/Z$ $(<\textbf{B}_\times)$

The types $\times$ and $\diamond$ of the slashes in $X/_\times Y$ and $X/_\diamond Y$ limit the categories to which these rules can apply.

In the full theory, these rules are generalized to "second level" cases, in which the secondary function is of the form $(Y|Z)|W$ such as the following "forward crossing" instance:

(12) *The Forward Crossing Level Two Composition Rule*
    $X/_\times Y \quad (Y\backslash Z)\backslash W \Rightarrow_{\textbf{B}_\times} (X\backslash Z)\backslash W$ $(>\textbf{B}_\times^2)$

The combination of crossing rules and generalized composition is the source of (slightly) greater than context-free expressive power in CCG, allowing analyses of trans-context-free constructions like Germanic crossed dependencies (Bresnan et al. 1982; Steedman 1985, 2000).

All syntactic rules in CCG are subject to a generalization called the Combinatory Projection Principle (CPP), which says that rules must apply consistent with the directionality specified on the primary function $X|Y$, and must project unchanged onto their result the directionality of any argument(s) specified on the secondary function $Y|Z$:[8]

(13) *The Combinatory Projection Principle (CPP)*
    Syntactic combinatory rules are binary rules that apply to contiguous non-empty categories of the specified syntactic types in the specified linear order, such that the syntactic type and directionality of any argument in the inputs that also appears in the result are the same.

---

 7 It will be convenient in the following discussions to refer to the function $X|Y$ in such rules as the "primary" function, and to the other function $Y|Z$ (etc.) as "secondary." While we continue to suppress explicit semantics for the purposes of the present article, like the application rules (6) the composition rules (10) and (11) have an invariant surface-compositional semantics, such that if the meaning of the primary function $X|Y$ is a functor f and that of the secondary function $Y|Z$ is g, then the meaning of the result $X|Z$ is $\lambda z.f(g\,z)$, the *composition* of the two functors, which if applied to an argument of type Z and meaning a, yields an X meaning $f(g\,a)$.
 8 This principle is defined more formally in Steedman (2000; 2012) in terms of three more elementary Principles of Adjacency, Consistency, and Inheritance.

Because directionality is originally specified in the lexicon, this principle amounts to the statement that syntax cannot override the word order specified there.

In particular, this principle universally excludes rules like the following from CCG:

(14)
$$
\begin{array}{lll}
Y & X/Y & \not\Rightarrow X \\
Y/Z & X/Y & \not\Rightarrow X/Z \\
X/Y & Y/Z & \not\Rightarrow X\backslash Z \\
X/Y & Z \quad Y & \not\Rightarrow X\ Z
\end{array}
$$

The same principle excludes all movement, copying, deletion-under-identity, or other action-at-a-distance, all structure-changing operations such as "restructuring," "reanalysis," or "reconstruction," and all "traces" and other syntactic empty categories, and makes syntactic derivation strictly type-dependent, rather than structure-dependent.

The types $\diamond$ and $\times$ on the slashes on the primary function $X|Y$ in the composition rules (10) and (11), like the type $\star$ on the application rules (6), allows categories to be lexically restricted as to which of the rules can apply to them or to their projections under the CPP. The absence of specific slash-typing on the secondary function $Y|Z$ is an abbreviation meaning that it schematizes over all slash types. However, the CPP (13) requires that the corresponding slash type on the argument $|Z$ in the result $X|Z$ is the *same* slash type.

This slash-typing system is different from the one used by Baldridge and Kruijff (2003) in that types are not organized into a type hierarchy. Avoiding slash type hierarchy allows for a more strict specification of which rules can be applied. The type we now write as $X/_{\diamond\star}Y$ was written following Steedman (2005) and Baldridge and Kruijff (2003) as $X/_{\diamond}Y$, and $X/_{\times\star}Y$ was written $X/_{\times}Y$.[9]

The inclusion of the harmonic composition rules (10) allows some additional derivations, and supports a variety of "non-constituent" coordinations (Steedman 1985; Dowty 1988), of which the following is a simple example:[10]

(15)

These	five	fat	and	seven	lean	cattle
$NP/_{\diamond\star}NumP$	$NumP/_{\diamond\star}N'$	$N'/_{\diamond\star}N$	$(X\backslash_\star X)/_\star X$	$NumP/_{\diamond\star}N'$	$N'/_{\diamond\star}N$	N

$$
\begin{array}{c}
\cfrac{NumP/_{\diamond\star}N' \quad N'/_{\diamond\star}N}{NumP/_{\diamond\star}N}{}_{>\mathbf{B}} \qquad\qquad \cfrac{NumP/_{\diamond\star}N' \quad N'/_{\diamond\star}N}{NumP/_{\diamond\star}N}{}_{>\mathbf{B}} \\[2ex]
\cfrac{(NumP/_{\diamond\star}N)\backslash_\star(NumP/_{\diamond\star}N)}{}{}_{>} \\[2ex]
\cfrac{NumP/_{\diamond\star}N}{}{}_{<} \\[2ex]
\cfrac{NumP}{}{}_{>} \\[2ex]
\cfrac{NP}{}{}_{>}
\end{array}
$$

The crossing composition rules (11), unlike the harmonic rules (10), have a reordering effect that is relevant to the present discussion. For example, in English they allow a non-movement-based account of the "Heavy NP Shift" construction, thus:

---

9 The present slash-type system is believed to be equivalent to the rule-based type-system of Steedman (2000), distinct from the earlier one investigated by Kuhlmann, Koller, and Satta (2015), and restoring weak equivalence to TAG.

10 The variable X in the conjunction category schematizes over a bounded number of types. The category's $\star$ slash types impose the across-the-board constraint on coordination (Steedman 2012), and are a consequence of its semantics, which follows Partee and Rooth (1983).

(16)

I	will	buy	tomorrow	a very heavy book
NP	$(S\backslash NP)/VP$	VP/NP	$VP\backslash VP$	NP

$$\frac{VP/NP \quad VP\backslash VP}{VP/NP}{}_{<\mathbf{B}_\times}$$

$$\frac{VP/NP \qquad\qquad\qquad NP}{VP}{}^{>}$$

$$\frac{(S\backslash NP)/VP \qquad VP}{S\backslash NP}{}^{>}$$

$$\frac{NP \qquad\qquad S\backslash NP}{S}{}_{<}$$

It will be obvious from the above derivation that allowing the crossing composition rules (11) to apply to unrestricted categories induces *alternation of word order*, as here between the Heavy-Shifted order in (16) and the canonical order *I will buy a book tomorrow*. Steedman (2020) shows that such word-order alternations can be excluded for fixed NP word-order languages of the kind considered by Cinque by restricting the slash type of the functor categories in the lexicon as either $\times$ ("only crossing-compose") or $\star$ ("only apply").

If, as in the alternation in English between (15) and the canonical order, we want a category to combine by *both* forward harmonic composition *and* forward application, then we assign a lexical category of the form $X/_{\diamond\star}Y$, bearing the union of $\diamond$ and $\star$ slash types, as in derivation (15). If we want all three rule-types to apply to a forward category, to allow free word order, then we assign it the union of all three slash types $X/_{\diamond\times\star}Y$, which to save space and maintain compatibility with earlier notations we write as the universal forward slash X/Y as in (16). However, such details of language-specific fine-tuning of grammars can be ignored for the present purposes.

3. The Generalization Concerning the Permutations Allowed by CCG

In this section we prove that the earlier claim that CCG generates only separable permutations holds for constructions of arbitrary length. Section 3.1 extends the definition of natural order of dominance to an arbitrary number of elements and shows how the natural order of dominance follows from the CCG lexicon that defines constructions. Section 3.2 then defines the notion of separable permutations. Sections 3.3 and 3.4 define the proposition more formally and prove its correctness. Section 3.5 shows how this result applies to the standard version of CCG including lexical type-raising and the full range of combinatory rules.

3.1 Natural Order of Dominance

The notion of natural order of dominance has been used up until this point informally, in the context of the categories defining the four core elements of the NP and VP constructions that point the way to a generalization. To see the generalization to other constructions of arbitrary length, a more formal definition is needed.

We define the Natural Order of Dominance (NOD) over categories of type $X|Y$ where both X and Y are categories of first order, as follows. A word a_i with a category $X|Y$ *immediately dominates* a word a_j with category $Y|Z$.[11] A NOD over n words is then

---

11 This definition of immediate dominance is a special case of dependency structure for Pure First-Order CCG as defined by Koller and Kuhlmann (2009). We shall see below that this definition in fact generalizes to the second-order case where X and/or Y are function categories.

defined as an ordering $a_1, \ldots a_n$ such that a_i immediately dominates a_{i+1} for all $i < n$. (There may be more than one NOD if there is more than one word of the same type.)

Because every word's category can select for only one argument category, and because each word a_i in a NOD immediatly dominates word a_{i+1}, that NOD defines a total ordering of words paired with their categories. We can write any NOD as a sequence of categories $X_1|X_2$, $X_2|X_3$, ... $X_{n-1}|X_n$, $X_n|X_{n+1}$ where each category $X_i|X_{i+1}$ is a category of a word taking position i in the NOD.

In the previous sections we have invoked NOD over the elements of noun and verb phrases with categories: $NP|NumP$, $NumP|N'$, $N'|N$, N and $VP_1|VP_2$, $VP_2|VP_3$, $VP_3|VP_4$, VP_4. These sequences match the generalized definition of NOD except for the last element being atomic X_n instead of a function category $X_n|X_{n+1}$. To allow such sequences, we extend the definition of NOD to allow the last element of the node $X_n|X_{n+1}$ to be an atomic X_n. Because $|X_{n+1}$ will never be canceled, it makes no difference to NOD whether it is present or not. In the proofs we will use the $X_n|X_{n+1}$ form for convenience.

The linearization of these n words that is consistent with their NOD will be called their **Canonical Linearization** (CL).

3.2 Separable Permutations and Separating Trees

Separable permutations were first formulated in algorithmic form by Avis and Newborn (1981) and have since then been rediscovered in different (but formally equivalent) forms such as bootstrap percolation in permutation matrices (Shapiro and Stephens 1991), permutations avoiding the "forbidden" patterns 2413 and 3142 (West 1996), and inversion transduction grammar (ITG) permutations (Wu 1996, among others).

The pattern-avoiding definition of West (1996) can be stated in the following way: The permutation π is separable if and only if there are no digits $a < b < c < d$ that appear in π as $b \ldots d \ldots a \ldots c$ or $c \ldots a \ldots d \ldots b$. For example, 4 1 6 3 5 2 is not separable because the digits $3 < 4 < 5 < 6$ form the forbidden pattern $4 \ldots 6 \ldots 3 \ldots 5$. Notice that the elements of the forbidden pattern do not need to be immediately next to each other: They are forbidden even when discontiguous.

Bose, Buss, and Lubiw (1998) define the separable permutations in terms of *separating trees*, which we define for present purposes as follows:

(17) A separating tree over a permutation of a totally ordered set 1 ... n is a binary tree in which each node dominates a continuous range i ... k of elements of the ordering, which its chidren split exhaustively into two contiguous continuous subranges i ... j and $j + 1$ ... k. Nodes are labeled $+$ if the range of their left child precedes that of their right child, and $-$ if the range on the left succeeds that on the right.

For example, a permutation such as 4 3 1 2 5 8 6 7 over the range $[1 \ldots 8]$ can be split into two contiguous sub-sequences over continuous ranges, namely: 4 3 1 2 over range $[1 \ldots 4]$ and 5 8 6 7 over range $[5 \ldots 8]$, with the resulting binary node labeled $+$. If we continue to recursively factorize contiguous sub-sequences over continuous ranges, we form the separating tree shown in Figure 2a.

It is important for what follows to note that the identities of the leaves of a separable tree are redundant, in the sense that they are fully determined by the tree structure and $+/-$ labels on the non-terminal nodes.

Bose, Buss, and Lubiw (1998) show that labels on separating trees define a sorting operation that reorders the children based on their ranges: If the label on a node is $+$
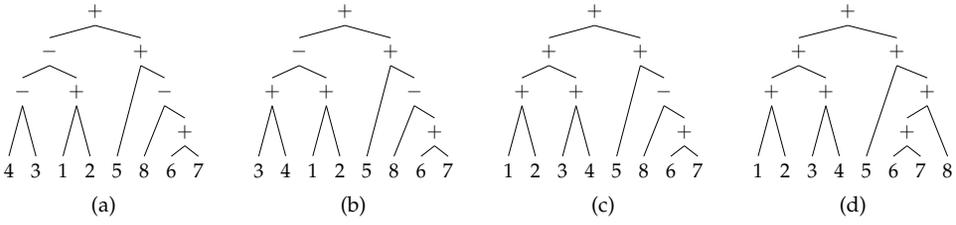
Figure 2
Separating trees.

then the range in the canonical linearization covered by the left child is ordered before that of the right child, and the existing order and label $+$ is retained. If the label is $-$, then the CL range covered by the left child is ordered after that of the right, and the $-$ operator is applied to restore the canonical order of its children, changing the label to $+$. This sorting procedure is exemplified in the series of transformations of the trees in Figure 2 that swap children of all nodes labeled with $-$ until the canonical order results.

The restriction of separating trees to binary structures limits the permutations that they can cover. For instance, we cannot create a separating tree for the permutation 2 4 1 3 because it cannot be split into contiguous subtrees covering continuous ranges.

Bose, Buss, and Lubiw (1998) show that the set of permutations that is described by separating trees is precisely the set of separable permutations. In other words, every separable permutation can be mapped to a separating tree and every separating tree can be mapped to a separable permutation.

Although it is clear that every separating tree maps to a unique separable permutation, some separable permutations can be mapped to multiple different separating trees. We will return to this point in Section 4.

3.3 The Claim

We observe that there is a direct relation between CCG derivations and separating trees. The intuition is clear from Figure 3: All CCG derivations for the canonical linearization of the five-element NOD, exemplified in Figure 3a, consist entirely of forward combinations, while the isomorphic separating trees for the canonical linearization, exemplified in Figure 3b, consist entirely of positive $+$ nodes (cf. Figure 2d). Similarly, CCG derivations for non-canonical linearizations, exemplified in Figure 3c, include backward combinations, which correspond to $-$ nodes in the isomorphic separating tree in Figure 3d.[12]

We seek to prove that the following claims hold for any NOD of n elements:

1. CCG can derive *only* separable permutations of the canonical linearization.

2. CCG can derive *all* separable permutations of the canonical linearization.

3. The cardinality of the set of permutations derivable by CCG is bounded by S_{n-1}, the $n{-}1$th element of the Large Schröder Number series (OEIS series A006318).

---

12 The actual combinatory rules involved will either be crossed or harmonic, but we can ignore the distinction here because of our use of the universal | slash.
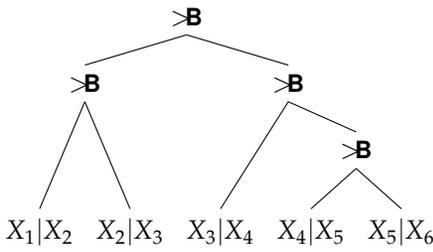
(a) CCG derivation for canonical linearization.
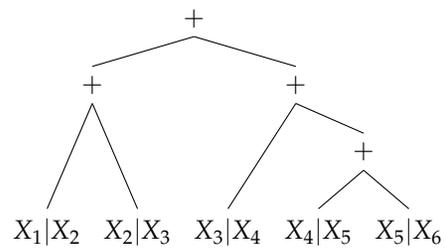
(b) Separating tree for canonical linearization.
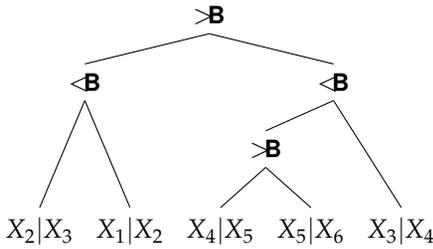
(c) CCG derivation for non-canonical linearization.
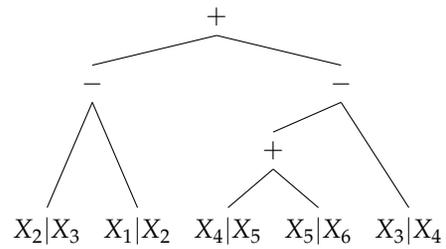
(d) Separating tree that derives the same non-canonical linearization.

Figure 3
Relation between CCG and separating trees.

We begin by proving three helpful lemmas. To prove the first of these lemmas we use **strong mathematical induction**, sometimes also called *complete induction*. To prove that some proposition $P(n)$ holds for all natural numbers using strong induction it is sufficient to prove the following statements:

(i)   base case $P(1)$ and

(ii)   inductive step $[P(1) \land P(2) \land \cdots \land P(k)] \implies P(k + 1)$ holds for all $k \in \mathbb{N}$

Strong induction is different than the more typical weak induction in the strength of the assumption that is taken in the inductive step: Weak induction assumes only $P(k)$, whereas strong induction assumes $P(1) \land \cdots \land P(k)$. Both are equally sound proving strategies, but strong induction is more convenient for proving propositions about trees. A more detailed distinction between weak and strong induction is available in many textbooks that cover proof methods (Eccles 1997; Liebeck 2010; Gunderson 2014).

3.4 The Proof

Lemma 1
For a fixed NOD of n elements with canonical linearization $X_1|X_2 \ \ X_2|X_3 \ \ \ldots \ \ X_n|X_{n+1}$:

(i)   CCG derivations can form all possible contiguous binary branching trees;

(ii)   Every node that spans from i to j in every tree derives a category $X_i|X_{j+1}$.

Proof. Base case:
For $n = 1$ we have a single category in the canonical linearization $X_1|X_2$. This is the only possible tree and it is binary in a trivial sense. Its only node spans from 1 to 1 deriving category $X_1|X_{1+1}$.

Inductive step: Assuming the strong inductive hypothesis holds for all $1 \leq k \leq n$, we prove that it also holds for $n + 1$. We analyze the canonical linearization $X_1|X_2 \quad X_2|X_3 \quad \ldots \quad X_{n+1}|X_{n+2}$ by splitting it into two contiguous parts where the first one is $X_1|X_2 \quad \ldots \quad X_k|X_{k+1}$ and the second one is $X_{k+1}|X_{k+2} \quad \ldots \quad X_{n+1}|X_{n+2}$ for any $k \in [1 \ldots n]$. Each of these two parts forms a smaller NOD with length $\leq n$. From the inductive assumption it follows that for both of these parts CCG can form all possible binary branching trees, and that all nodes spanning i to j within these parts will be labeled with $X_i|X_{j+1}$. That means that the root node of the left and right part will be $X_1|X_{k+1}$ and $X_{k+1}|X_{n+2}$, respectively. Now we can combine the left and right part into the whole structure using forward function composition $>\mathbf{B}$ to give us a new node $X_1|X_{n+2}$. This node satisfies the labeling condition of the lemma. The binary branching condition is also satisfied because we put no constraints on the size of the spans of the left and the right part (as long as they are non-empty), and by the inductive assumption, each part can derive all possible binary trees. □

Lemma 2
All word orders that can be derived by CCG for a given NOD can also be derived by an isomorphic separating tree that reorders the same permutation into the canonical linearization for that NOD.

Proof. It is sufficient to show that any CCG derivation tree corresponds to an isomorphic separating tree over the same permutation.

Consider any arbitrary CCG derivation tree for a canonical linearization of a NOD of length n. Example: One such derivation for the NOD for $n = 5$ is in Figure 3a.

From Lemma 1, we know that all nodes with span i to j will have label $X_i|X_{j+1}$ and all binary nodes will use forward function composition $>\mathbf{B}$. We can derive the same order with an isomorphic separating tree where instead of using CCG forward combination we have the $+$ operator at every node, because the children are in the canonical order. Example: Figure 3b.

Non-canonical linearization in CCG requires backward combinations. For any node in a CCG derivation for the canonical linearization whose left child has category $X_i|X_{j+1}$ and right child category $X_{j+1}|X_{k+1}$ we can obtain the inverse order by reversing the directionality of the combination and reversing the order of the child nodes, making $X_{j+1}|X_{k+1}$ precede $X_i|X_{j+1}$, so that all words in the range $j + 1$ to k in the canonical order precede words i to j. Example: Figure 3c shows this reordering effect for the two nodes where forward is replaced by backward combination.

Any CCG derivation reordered in this way corresponds to an isomorphic separating tree in which the unrotated nodes corresponding to forward combination are relabeled as $+$ and the rotated nodes corresponding to backward combination are relabeled as $-$. Example: Figure 3d.

By the result of Bose, Buss, and Lubiw discussed in Section 3.2, applying the $-$ operator to convert all such nodes to $+$ and invert the linear order of their children reverses the process to recover the canonical linearization of which this is a separable permutation. □

Lemma 3
CCG can derive all possible separating trees for an NOD of arbitrary length.

Proof. By Lemma 1, CCG can derive all possible binary branching trees on the canonical linearization. By Lemma 2, each node in each of these binary trees can be labeled either $+$ or $-$, defining a separating tree isomorphic to the CCG derivation for the corresponding linearization. Because we saw in Section 3.2 that a separating tree determines the identities of its leaves, the set of all possible binary trees with all possible node labelings is the set of all possible separating trees under the definition (17) there.   □

Finally, we can prove the main claims that relate CCG derivations of NOD and separable permutations of canonical linearization of NOD.

Theorem 1
CCG derives only separable permutations.

Proof. By Lemma 2, all CCG derivations can be mapped to separating trees. All separating trees correspond to a separable permutation (Bose, Buss, and Lubiw 1998). Therefore CCG derives only separable permutations.   □

Theorem 2
CCG derives all separable permutations over a NOD.

Proof. By Lemma 3, CCG can derive all possible separating trees. Every separable permutation corresponds to a separating tree (Bose, Buss, and Lubiw 1998). Therefore, CCG derives all separable permutations.   □

Theorem 3
The cardinality of the set of permutations derivable by CCG is bounded by S_{n-1}, the $n-1$th Large Schröder Number.

Proof. By Theorems 1 and 2, CCG derives all and only separable permutations over an NOD of length n. The number of separable permutations of length n is the n–1th Large Schröder Number (Shapiro and Stephens 1991). Therefore, the number of permutations derivable by CCG for that NOD is also n–1th Large Schröder Number.   □

The recurrent formula for this series is given below, the base case being $S_0 = 1$:

$$(18) \qquad S_n = S_{n-1} + \sum_{k=0}^{n-1} S_k S_{n-k-1}$$

3.5 Generalization to Full CCG

The above results have been proven using a restricted version of CCG called "pure first-order" CCG (Koller and Kuhlmann 2009), in which all variables in categories $X_i|X_{i+1}$ correspond to atomic types such as N, rather than function categories, such as $(S \backslash NP)$. The standard version of CCG is a second-order calculus, in which arguments can have function types, a possibility that is central to the CCG analysis of relativization, coordination, and adjunction, including the phenomenon of parasitic gaps. Indeed, all

cases of so-called movement reduce in CCG to contiguous merger with a semantically second-order functor.

Categories that select for multiple arguments are "Curried" unary functors in CCG, with a function as their result, such as $(X_i|X_j)|X_{i+1}$. They do not define a NOD over the arguments X_{i+1}, X_j, because those arguments do not stand in a dominance relation, and they potentially allow the second-level composition rules such as (12) to apply. Nevertheless, they have a canonical linearization in which all combinations are forward, and (by Lemma 2) dominate separating trees over separable permutations of that canonical ordering, although they no longer allow *all* separable permutations. (The large Schröder numbers thus constitute an *upper bound* on the allowable permutations of canonical linearizations in CCG.)

The present results generalize immediately to full CCG. The semantics of the combinatory rules is invariant, so it makes no difference to the combinatorial possibilities if the "canceling" variable Y in a rule such as application matches an atomic category or a function category. (In fact, we have already exploited this possibility in writing the atomic symbols *VP* and *N* for what are semantically predicate or functional types.)

Type-raising is a lexical operation that exchanges the role of argument and function by making the former into a function over the latter and that does allow permutations that would not be allowed by the original categories. It does so by changing the NOD among the categories. However, because this is a lexical operation whose effects are determined before syntactic derivation can apply, the main result of the present paper still applies: Only separable permutations on the canonical linearization order defined by the lexical categories can be captured.

4. Normal-Form Separating Trees and CCG Parsing

The results shown above depended on the fact that the canonical linearization on the category sets of interest is binary tree-complete, and therefore separating tree-complete under rotation, and therefore separable-permutation-complete. However, we noted earlier that there are more separating trees than separable permutations, because many separating trees yield the same permutation. Indeed, for the canonical linearization of NOD of length n and its inverse, the number of separating trees is C_{n-1}, the n–1th Catalan Number (Church and Patil 1982), which is much larger than the corresponding Large Schröder number.

For example, Figure 4 shows all eight separating trees derived by rotation from the two binary trees on the canonical linearization of the canonical linearization CL_3 of three elements, in which $+$ on each of the two nodes means that it is unrotated and $-$ means that it is rotated. These are indeed all separable permutations of three elements, but the two orders 1,2,3 and 3,2,1 are represented by two separating trees, where $C_3 = 2$. For the purpose of efficient parsing with CCG, we would like there to be only one CCG derivation, and one separating tree, for each separable permutation.

4.1 Normal-Form Separating Trees

Shapiro and Stephens (1991) observed that the spurious ambiguity of separating trees arises when two nodes that are in parent–child relation have the same label, as for the trees in Figure 4a and 4b. They proposed a normal form that makes the mapping between the set of normal form separating trees and the set of separable permutations bijective. This normal form states that no node can be labeled with the same $+/-$ label as its right child. This excludes trees like those in Figure 4a and 4e and means
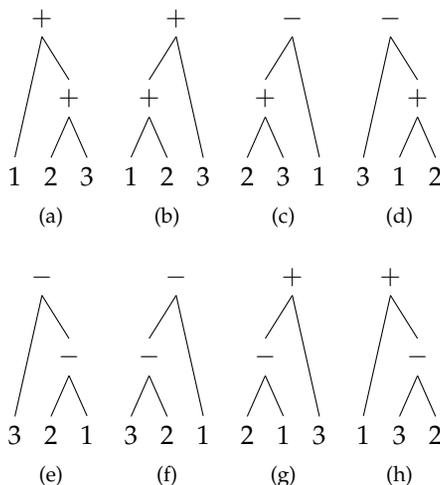
Figure 4
All possible separating trees for a three-word sentence.

that the number of normal-form separating trees is the same as that of the separable permutations.

This normal form for separating trees is reminiscent of the CCG derivational normal-form proposed by Eisner (1996) to enforce a one-to-one mapping between CCG syntactic derivations and semantic interpretation that they generate (cf. Hockenmaier and Bisk 2010). Eisner normal-form forbids particular sequences of combinatory rules that are of the same type from occurring in a parent-child relationship in the derivation.

Using normal-form separating trees, we can distinguish the possible cases, shown in Figure 5. Figures 5a and 5b show the configurations where the right child is a single word. Figures 5c and 5d show those where the right child is a non-single word constituent that adheres to the normal form. The length of the examples is identified as $n + 1$ in order to be consistent with the conventional numeration for the Large Schröder numbers. The length of the constituent A is k. We are interested in the number of normal-form separating-trees S_n, since that will also be the number of separable permutations of length $n + 1$.

The number of trees that conform to the pattern in Figure 5a clearly depends on the number of possible concrete sub-trees that can act as constituent A. Here A is of length $k = n$, which means that the number of separating trees of the pattern in Figure 5a is S_{n-1}. If we ignore the label of the root node, the other three patterns show all possible ways of constructing the right child: having a single terminal node, or binary node labeled $+$ or $-$. This means that we can express the sum of trees confirming to these three patterns as the number of trees that have their root unlabeled (since we are ignoring the root label) and have the left and right child confirming to the normal form for separating trees. For a given k, which plays a role of a split point in parsing, the number of those trees would be $S_{k-1}S_{n+1-k-1}$, but because k can be any number between 1 and n, and is not fixed, we need to sum over all of those values $\sum_{k=1}^{n} S_{k-1}S_{n-k}$. We can shift this sum to start from zero to obtain the equivalent formula $\sum_{k=0}^{n-1} S_k S_{n-k-1}$. When we put the formulas for all four patterns together we get the Large Schröder recurrence in Equation (18). We can conclude that the number of normal-form separating trees for a NOD of length n is also S_{n-1}.
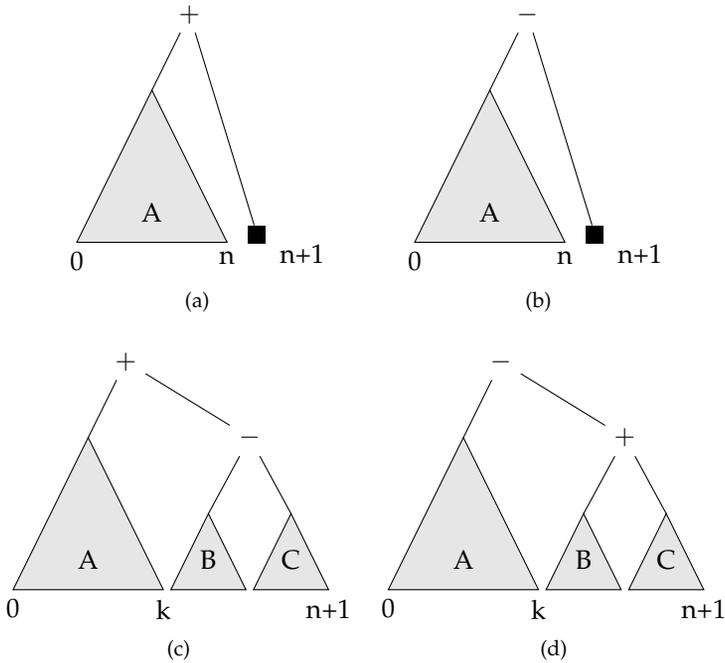
Figure 5
Normal form branching configurations.

4.2 The Relation of Large Schröder Numbers to Shift-Reduce Parsing

The Large Schröder numbers are often also presented in relation to the number of lattice paths in the Cartesian plane such that each path starts at lower-left corner $(0,0)$, ends at upper right corner (n,n), contains no points above the diagonal $y = x$ (that is, can only be in the lower right triangle), and is composed only of unit steps $\rightarrow$, $\uparrow$, and $\nearrow$ or more concretely $(1,0)$, $(0,1)$, and $(1,1)$ (Weisstein 2018). In this section we show how this view of Large Schröder numbers can be interpreted as Shift-Reduce parsing with CCG derivations corresponding to the normal-form separating trees defined above.

Shift-Reduce is one of the most widely used parsing algorithms in computational linguistics and it is probably the simplest way to construct a parser. It is based on having a parsing state with a stack (filled with constituents built so far) and a buffer (filled with words in the sentence suffix that are yet to be processed). A *shift* transition takes a word from the buffer and puts it on top of the stack. For binarized grammars like CCG, a *reduceX* transition replaces the two topmost constituents on the stack with a single new constituent labeled X. In our CCG derivations we will have two reduce operations: reduce+ for forward combinators and reduce− for backward combinators. We will again take an example sentence of length $n + 1$. The initial parsing state will be a stack containing the first word and a buffer containing the other n words.

We can visually represent any valid successful shift-reduce parsing trace by starting at the point $(0,0)$ and taking $\rightarrow$ to represent a shift transition and $\uparrow$ labeled $+$ or $−$ to represent a reduceX transition. The Shift-Reduce transition sequence for the derivation tree in Figure 6a is shown in Figure 6b. Clearly, such paths will never cross the diagonal,

27
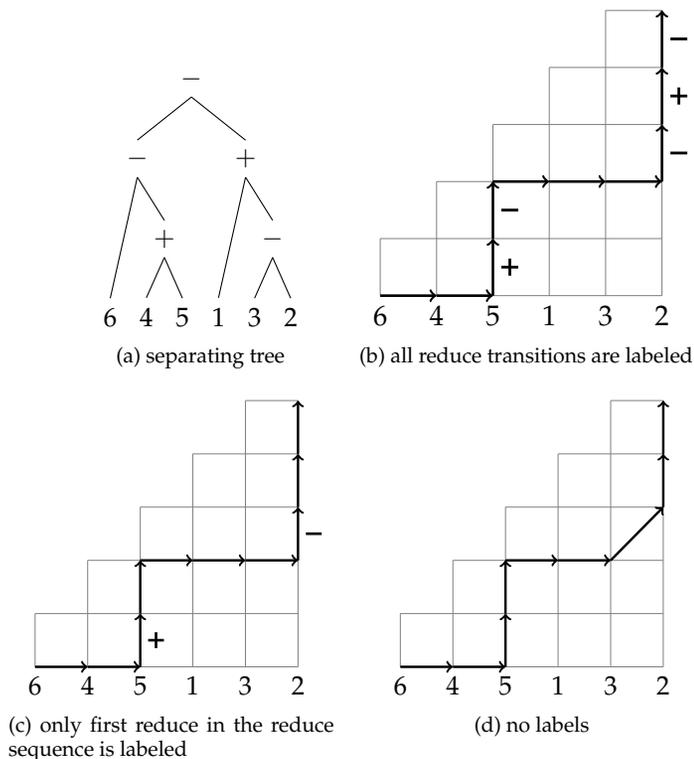
Figure 6
A separating tree and different representations of the shift-reduce transition sequence that recognizes it.

because we can never have more reduce operations than shifts and every successful parse has n shifts and n reduce steps, so that every path ends at (n, n).

The set of SR parse paths can be further restricted to those corresponding to the the normal-form parses and separating trees defined in the preceding Section 4.1, as follows. First, the normal form says that a parent node cannot have the same label as its right child node. What that means in SR parsing terms is that *we cannot have two consecutive transitions both of which are reduce+ or reduce−*. Therefore, when we have a sequence of reduce operations, expressed as an uninterrupted vertical sequence of ↑, we need only know the label of the first reduce transition in the sequence because the other labels can be inferred from the normal form: They will alternate between + and −. This modified representation is shown in Figure 6c.

The final modification is to use a slightly different encoding of the initial reduce in each of the reduce sequences. The first reduce step in the sequence always comes after a shift, which means that it forms a shape $\rightarrow \uparrow$. If that sequence is [shift, reduce−], we will encode it with a diagonal ↗ and if it is [shift, reduce+] we will encode it with existing→↑ but without any label, as shown in Figure 6d. This encoding defines a one-to-one mapping between CCG normal-form shift-reduce parses and paths in the Schröder lattice.

These observations support another way to think about the number of permutations that CCG allows for a NOD of length n in terms of SR normal-form parsing. When
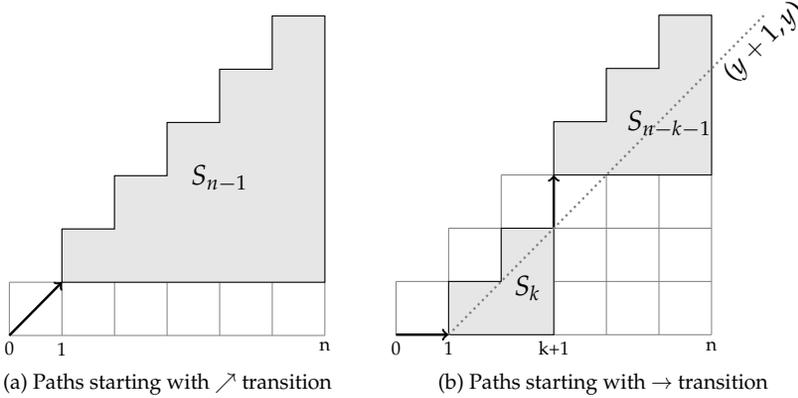
Figure 7
Counting paths in the lattice for different initial transitions.

we start in the initial position $(0,0)$ we can take paths that start either with $\nearrow$ or $\rightarrow$ ($\uparrow$ is illegal because it would cross the upper diagonal). If we take $\nearrow$ (Figure 7a) we find ourselves in a lattice of the same shape, smaller by one unit, which means that the number of normal-form paths through it is S_{n-1}. If, on the other hand, we take $\rightarrow$ we will end up in position $(1,0)$. Any path that goes from $(1,0)$ to (n,n) must at some point have a vertical arrow that crosses diagonal $(y+1,y)$, as shown in Figure 7b. The first point at which the path crosses that diagonal will be some concrete point $(k+1,k)$. The number of paths that go from $(1,0)$ to (n,n) and contain arc $\uparrow$ at $(k+1,k)$ is $S_k S_{n-k-1}$. Again, we need to sum over all possible paths that cross the diagonal $(y+1,y)$ over each possible k, which gives $\sum_{k=0}^{n-1} S_k S_{n-k-1}$. Summing results for initial $\nearrow$ and initial $\rightarrow$ again yields the Large Schröder number recurrence function from Equation (18).

5. Statistical Significance of the Global Result

The above results confirm that all 22 separable permutations are allowed, at least as alternates, for any construction defined by four elements of the form $A|B, B|C, C|D, D$, and the 2 non-separable permutations are forbidden. The fact that in both the constructions surveyed, the distribution of the frequencies with which the separable permutation orders are actually observed are so skewed means that some orders are so vanishingly rare as to only be attested in one construction, and/or even by only one or two languages. The possibility that this skewness arises from soft constraints related to ease of processing or learnability is interesting in their own right, and is the subject of ongoing linguistic research (Culbertson, Smolensky, and Legendre 2012; Merlo 2015; Dryer 2018; Merlo and Ouwayda 2018).[13]

Nevertheless, according to the present theory, the two permutations (g) and (j) are excluded for a different reason, namely, by a hard constraint that is a consequence of the theory of grammar itself. The fact that the two disallowed orders are the only two that are totally unattested in the union of Cinque's, Nchare's, and Abels's surveys is a

---

13 This possibility is also acknowledged by Abels and Neeleman (2012).

strong result that is unlikely to have arisen by chance. The strength of this result can be assessed quantitatively, as follows.

There are 21 permutations that are attested for NP. We will assume that all 21 are sampled without replacement from a uniform distribution of 24 permutations. Without knowing which 21 are attested, we could form 24 theories that exclude one permutation, 24×23 that exclude two, and $24 \times 23 \times 22$ that exclude 3 permutations. For each one of them, there is a different number of possible observations (i.e., different sets of 21 permutations) that would *not* falsify them. For instance, a theory that forbids 3 permutations will be consistent only with one possible set of 21 observed permutations, while a theory that forbids 2 (like the one presented here) will be consistent with 22 different sets of 21 permutations. The probability of our predictions being not falsified by chance is the number of observations (permutation sets) that would be consistent with our theory divided by the total number of possible observations—that is, approximately 1 in 100:

$$(19) \qquad p = \frac{\binom{22}{21}}{\binom{24}{21}} = \frac{22}{2024} \approx 0.0109$$

If on the other hand we accept (h), as tentatively attested on the basis of Abels' and Wurmbrand's accounts, the probability of getting this stronger result by chance drops to around 1 in 250:

$$(20) \qquad p = \frac{\binom{22}{22}}{\binom{24}{22}} = \frac{1}{276} \approx 0.0036$$

Both results are evidently unlikely to come up by chance.

6. Discussion

The prediction of CCG concerning possible word orders is significant from both scientific and engineering perspectives. The Large Schröder series S_n grows much more slowly with n than the total factorial number of permutations $n!$, so that the proportion $\frac{n!-S_{n-1}}{n!}$ of non-separable permutations that are disallowed by CCG grows rapidly with n, as 0%, 0%, 0%, 8%, 25%, 45%, 64%, 79%, 89%, ... The proportion $\frac{S_{n-1}}{n!}$ of separable permutations that are allowed shrinks correspondingly rapidly with n.

From a practical NLP perspective there are many areas where this saving can be achieved, with machine translation and parsing, including "sequence-to-sequence" semantic parsing, being the obvious use-cases. If we know the limited number of word-order variations among languages, alignment algorithms (or attention as its neural equivalent [Bahdanau, Cho, and Bengio 2015]) can be constrained (or biased) in the right direction (Zhang and Gildea 2005; Dong and Lapata 2016).[14]

---

14 This observation is also implicit in Whitelock's (1992) early proposal for "shake and bake" translation for Japanese.

Unsupervised parser induction is another application where introducing an inductive bias (or hard constraint) for contiguous derivation trees with a normal form significantly reduces the number of alternatives that learner needs to consider (Clark and Lappin 2010). These savings are especially significant in comparison to other theories of grammar that use discontinuous constituents, empty strings, or very powerful, often Turing-complete, unification mechanisms.

We saw earlier that the number and type of permutations allowed by CCG appears to correspond to the crosslinguistic variation in word order that is actually attested. CCG also provides a strong and easily testable prediction of what word orders will be found in the future in human languages that have not so far been examined (Steedman 2020). These properties stem from the combinatory projection principle (13), and in particular from the restriction to *binary* combinatory rules.

So far, the empirical studies of crosslinguistic variation cited above have been confined to no more than four core elements of NP and VP. However, we have shown that the results proved here are completely general to sets of categories of any size that define a natural order of dominance. The verb group in particular can be naturally extended to serial verbs of various kinds. The cartographic spines of functional projections identified by Cinque and Rizzi (2008) offer several tens of functional heads in a supposedly universal order of dominance, although there is more to say here about the relation of syntax and morphology and the degrees of crosslinguistic variation they allow. Any firm evidence for any non-separable permutation of these elements in any language for which the natural order of dominance can be unequivocally determined would immediately falsify the present theory. As we noted earlier, the number of forbidden non-separable falsifying cases rises very rapidly as the size of such sets increases beyond 4.

However, the power-law distributions that we observe on all such dimensions of variation, and the fact that the performance factors that lie behind them favor continuity and semantic homomorphism, mean that those falsifying cases will always lie out in the long tail. We have seen that the total number of languages that have ever been studied in the requisite detail is in the case of the nominal construction only just enough to take us far enough out into that long tail to achieve a confidence level against ever seeing a non-separable permutation of even 1 in 100. That means our chances of reaching a definitive answer to the question for even a five-element construction are quite daunting.

Naturally, our work is not the first work to try to find a formal constraint of this kind. Here we review some results from other formalisms.

ITG Inversion Transduction Grammars (Wu 1996) are a type of syntax transduction grammar developed for the task of machine translation. ITG can generate only separable permutations of the words in the source language, so it is natural to ask how this property of ITG differs from the one proposed here. The main difference lies in what is considered the reference word order that is being permuted. Here, CCG gives a clear definition: The reference word order is the CL that reflects the NOD, and ultimately the semantics. To say that CCG generates only separable permutations of the NOD is to make a clear prediction about what permutations can be found in natural languages. However, the same is not the case with ITG, which does not identify any NOD. For ITG the reference word order is that of the source language in the MT task. Taking a different language as a source language gives a different prediction of what will be found in the world's languages. Take, for instance, the previous example "these five young lads." If ITG takes English as a source language then it will make the same prediction as CCG. But if we had, for example, taken Spanish as a source language, with the word order "these five lads young," ITG would be able to derive "five lads these young," which has

16 out of the complete 24 possible permutations. These 16 include the 14 permutations claimed by Cinque (2005), but exclude some of those claimed by Nchare (2012) for the free word-order language Shupamem. The assumption that Stabler makes is that the grammar uses neither empty strings nor head-movement, both of which are prevalent in most linguistic work on minimalist syntax. Interestingly, Stabler's 16 include permutation (j), which is one of the two non-separable permutations that are excluded under the present theory.

The limitation of combinatory reduction to the separable permutations is also implicit in Svenonius's (2007) ***1-3-X-2** restriction on adjunct placement (among other constructions), where X is an adjunct to 1, and in Williams's (2003, pages 203-211) proposal for his categorial calculus **CAT**. CAT has a standard directional categorial lexicon and rule of application, with a combinatory operation REASSOCIATE equivalent to composition, and an operation FLIP, unavailable in CCG, which reverses the directionality on the argument of a functor category.

Williams's CAT variety of Minimalism is closely related to CCG. However, without the addition of morpho-lexical type-raising, permitting arguments to compose, CAT is unable to express the variety of constructions that CCG makes available, which include relativization, coordination-reduction, Germanic scrambling, and Hungarian VM fronting (Steedman 2020). In the absence of lexical type-raising these constructions require the extension of CAT by the addition of powerful rules of "action at a distance" unavailable in CCG, such as movement, copying, and/or deletion, together with attendant constraints to limit their considerable expressive power.

A related proposal to ours by Medeiros (2018) hypothesizes that the set of allowed permutations is the one that could be sorted with a single stack, so-called **stack-sortable permutations** or 231-avoiding permutations (Knuth 1968). This set of permutations exactly covers Cinque's 14, but does not fit the other empirical data considered here: 7 permutations out of 21 that are observed are predicted not to be possible by Medeiros. As with Williams (2003), it is also unclear that Medeiros's mechanism will also handle phenomena of unbounded or *wh* movement, as CCG does. However, there is an interesting relation between the sorting automata for 231-avoiding permutations and the automata that sort separable permutations: 231-avoiding permutations are sortable with a single pop-stack while separable permutations are sortable with a sequence of $n-1$ pop-stacks (Avis and Newborn 1981).

7. Conclusion

This article has argued that the facts of attested word order in the constructions considered exploits all and only the degrees of freedom that CCG allows, despite the fact that some of the orders that CCG allows for some constructions are vanishingly rare due to unrelated performance soft constraints that induce Zipfian (power-law) distributions discussed by Culbertson, Smolensky, and Legendre (2012) and Culbertson and Adger (2014).

Explanatory adequacy in a theory depends on being able to explain why the data has just the degrees of freedom it does, and why other things *don't* happen. If the latter have been prevented by hard constraints, as in various ways they have been for these constructions by Cinque, Svenonius, Abels, Neeleman, Nchare, and Stabler, then those constraints themselves have in turn to be explained.

No hard constraints are needed at the level of competence grammar in CCG. The limitation on the grammatical permutation of n elements in natural grammars to the Large Schröder number S_{n-1} of separable permutations is a formal universal stemming

from the limited expressivity of CCG itself—specifically, its restriction of its operators to strictly adjacent pairs of non-empty constituent types via the Combinatory Projection Principle (13). This result is a corollary of the Combinatory Categorial theory of grammar, and the formally explicit reduction that it affords of what in Minimalist terms would be called MOVE or "internal merge" to contiguous (that is, "external") MERGE.

Appendix A: The Ensemble of Attested Word Orders Over 4 Elements

In this appendix we review the attested orders of four core elements of NP and VP as reported by Cinque, Nchare, and Abels and Neeleman. We also give CCG derivation trees for these orders from Steedman (2020).

If our lexicon consists of four entirely unrestricted categories of the form $A|B$, $B|C$, $C|D$, D, then for both the nominal and verbal constructions all 22 separable permutations are allowed, and only the two non-separable permutations are unanalyzable, as a consequence of the combinatory projection principle (13), as follows.

The NP

For the NP, the 22 orders are allowed under the following derivations ("$\times$" marks the two sequences (g) and (j) that are unanalyzable). Only essential compositions are indicated: all other combinations are application. For non-basic orders, the annotation "from z" indicates the basic pure-applicative order among those in (8) on whose lexicon a particular derived order is based:[16]

(21) a.   These   five   young lads   Both (basic: v.many)
        $\overline{NP/NumP}$ $\overline{NumP/N'}$ $\overline{N'/N}$ $\overline{N}$

      b.   These   five   lads young   Both (basic: many)
        $\overline{NP/NumP}$ $\overline{NumP/N'}$ $\overline{N}$ $\overline{N'\backslash N}$

      c.   These   lads   five   young   Cinque (from b: v.few)
        $\overline{NP/NumP}$ $\overline{N}$ $\overline{NumP/N'}$ $\overline{N'\backslash N}$
                                    $\underline{\qquad\qquad}^{>\mathbf{B}_\times}$
                                    $NumP\backslash N$

      d. Lads   these   five   young   Cinque (from b: few)
        $\overline{N}$ $\overline{NP/NumP}$ $\overline{NumP/N'}$ $\overline{N'\backslash N}$
                        $\underline{\qquad\qquad}^{>\mathbf{B}}$
                        NP/N'
              $\underline{\qquad\qquad\qquad\qquad}^{>\mathbf{B}_\times}$
              $NP\backslash N$

      e.   Five   these   young lads   Nchare (from r)
        $\overline{NumP/N'}$ $\overline{NP\backslash NumP}$ $\overline{N'/N}$ $\overline{N}$
        $\underline{\qquad\qquad\qquad}_{<\mathbf{B}_\times}$
              NP/N'

      f.   Five   these   lads young   Nchare (from s)
        $\overline{NumP/N'}$ $\overline{NP\backslash NumP}$ $\overline{N}$ $\overline{N'\backslash N}$
        $\underline{\qquad\qquad\qquad}_{<\mathbf{B}_\times}$
              NP/N'

---

16 See Appendix A for the definition of the quantized counts.

g. × $\overline{\text{Five}}$ $\overline{\text{lads}}$ $\overline{\text{these}}$ $\overline{\text{young}}$ not attested (disallowed)
      $\overline{NumP/N'}$ $\overline{N}$ $\overline{NP\backslash NumP}$ $\overline{N'\backslash N}$
      $\underline{\qquad\qquad}_*$
                   $\underline{\qquad\qquad\qquad}_*$
                                      $\underline{\qquad\qquad}_*$

h. ? $\overline{\text{Lads}}$ $\overline{\text{five}}$ $\overline{\text{these}}$ $\overline{\text{young}}$ not attested (from s)
      $\overline{N}$ $\overline{NumP/N'}$ $\overline{NP\backslash NumP}$ $\overline{N'\backslash N}$
                   $\underline{\qquad\qquad\qquad}_{<\mathbf{B}_\times}$
                          $\overline{NP/N'}$
      $\underline{\qquad\qquad\qquad\qquad\qquad}_{>\mathbf{B}_\times}$
                          $NP\backslash N$

i. $\overline{\text{Young}}$ $\overline{\text{these}}$ $\overline{\text{five}}$ $\overline{\text{lads}}$ Nchare (from n)
    $\overline{N'/N}$ $\overline{NP/NumP}$ $\overline{NumP\backslash N'}$ $\overline{N}$
                   $\underline{\qquad\qquad\qquad}_{>\mathbf{B}_\times}$
                          $NP\backslash N'$
    $\underline{\qquad\qquad\qquad\qquad\qquad}_{<\mathbf{B}_\times}$
                          NP/N

j. × $\overline{\text{Young}}$ $\overline{\text{these}}$ $\overline{\text{lads}}$ $\overline{\text{five}}$ not attested (disallowed)
      $\overline{N'/N}$ $\overline{NP/NumP}$ $\overline{N}$ $\overline{NumP\backslash N'}$
      $\underline{\qquad\qquad\qquad}_*$
                   $\underline{\qquad\qquad}_*$
                          $\underline{\qquad\qquad\qquad}_*$

k. $\overline{\text{Young}}$ $\overline{\text{lads}}$ $\overline{\text{these}}$ $\overline{\text{five}}$ Both (from n:v.few)
    $\overline{N'/N}$ $\overline{N}$ $\overline{NP/NumP}$ $\overline{NumP\backslash N'}$
                          $\underline{\qquad\qquad\qquad}_{>\mathbf{B}_\times}$
                                 $NP\backslash N'$

l. $\overline{\text{Lads}}$ $\overline{\text{young}}$ $\overline{\text{these}}$ $\overline{\text{five}}$ Both (from o: few)
    $\overline{N}$ $\overline{N'\backslash N}$ $\overline{NP/NumP}$ $\overline{NumP\backslash N'}$
                          $\underline{\qquad\qquad\qquad}_{>\mathbf{B}_\times}$
                                 $NP\backslash N'$

m. $\overline{\text{These}}$ $\overline{\text{young}}$ $\overline{\text{five}}$ $\overline{\text{lads}}$ Nchare (from n)
    $\overline{NP/NumP}$ $\overline{N'/N}$ $\overline{NumP\backslash N'}$ $\overline{N}$
                   $\underline{\qquad\qquad\qquad}_{<\mathbf{B}_\times}$
                          $NumP/N$

n. $\overline{\text{These}}$ $\overline{\text{young}}$ $\overline{\text{lads}}$ $\overline{\text{five}}$ Both (basic: few)
    $\overline{NP/NumP}$ $\overline{N'/N}$ $\overline{N}$ $\overline{NumP\backslash N'}$

o. $\overline{\text{These}}$ $\overline{\text{lads}}$ $\overline{\text{young}}$ $\overline{\text{five}}$ Both (basic: many)
    $\overline{NP/NumP}$ $\overline{N}$ $\overline{N'\backslash N}$ $\overline{NumP\backslash N'}$

p. $\overline{\text{Lads}}$ $\overline{\text{these}}$ $\overline{\text{young}}$ $\overline{\text{five}}$ Both (from o: v.few)
    $\overline{N}$ $\overline{NP/NumP}$ $\overline{N'\backslash N}$ $\overline{NumP\backslash N'}$
                          $\underline{\qquad\qquad\qquad}_{<\mathbf{B}}$
                                 $NumP\backslash N$
    $\underline{\qquad\qquad\qquad\qquad\qquad}_{>\mathbf{B}_\times}$
                          $NP\backslash N$

q. $\overline{\text{Five}}$ $\overline{\text{young}}$ $\overline{\text{these}}$ $\overline{\text{lads}}$ Nchare (from r)
    $\overline{NumP/N'}$ $\overline{N'/N}$ $\overline{NP\backslash NumP}$ $\overline{N}$
    $\underline{\qquad\qquad\qquad}_{>\mathbf{B}}$
          $NumP/N$
    $\underline{\qquad\qquad\qquad\qquad\qquad}_{<\mathbf{B}_\times}$
                   NP/N

r. $\dfrac{\text{Five}}{NumP/N'}$ $\dfrac{\text{young}}{N'/N}$ $\dfrac{\text{lads}}{N}$ $\dfrac{\text{these}}{NP\backslash NumP}$ 　　　　Both (basic: few)

s. $\dfrac{\text{Five}}{NumP/N'}$ $\dfrac{\text{lads}}{N}$ $\dfrac{\text{young}}{N'\backslash N}$ $\dfrac{\text{these}}{NP\backslash NumP}$ 　　　　Both (basic: many)

t. $\dfrac{\text{Lads}}{N}$ $\dfrac{\text{five}}{NumP/N'}$ $\dfrac{\text{young}}{N'\backslash N}$ $\dfrac{\text{these}}{NP\backslash NumP}$ 　　　　Both (from s: few)

$$\cfrac{\cfrac{}{N}\quad \cfrac{NumP/N' \qquad N'\backslash N}{NumP\backslash N}{>}\mathbf{B}_{\times}\quad NP\backslash NumP}{}$$

u. $\dfrac{\text{Young}}{N'/N}$ $\dfrac{\text{five}}{NumP\backslash N'}$ $\dfrac{\text{these}}{NP\backslash NumP}$ $\dfrac{\text{lads}}{N}$ 　　　　Nchare (from w)

$$\cfrac{\cfrac{NumP\backslash N' \qquad NP\backslash NumP}{NP\backslash N'}{<}\mathbf{B}}{NP/N}{<}\mathbf{B}_{\times}$$

v. $\dfrac{\text{Young}}{N'/N}$ $\dfrac{\text{five}}{NumP\backslash N'}$ $\dfrac{\text{lads}}{N}$ $\dfrac{\text{these}}{NP\backslash NumP}$ 　　　　Nchare (from w)

$$\cfrac{N'/N \qquad NumP\backslash N'}{NumP/N}{<}\mathbf{B}_{\times}$$

w. $\dfrac{\text{Young}}{N'/N}$ $\dfrac{\text{lads}}{N}$ $\dfrac{\text{five}}{NumP\backslash N'}$ $\dfrac{\text{these}}{NP\backslash NumP}$ 　　　　Both (basic: v.few)

x. $\dfrac{\text{Lads}}{N}$ $\dfrac{\text{young}}{N'\backslash N}$ $\dfrac{\text{five}}{NumP\backslash N'}$ $\dfrac{\text{these}}{NP\backslash NumP}$ 　　　　Both (basic: v.many)

As noted earlier, to limit a given language to just one of these word orders, we need to restrict its lexicon via slash-typing to allow *only* the rules we see in the relevant derivation in (21) to apply to the categories in question. For example, to restrict the lexicon of Maasai to allow only order (c) and no other, we need the following more restricted version of the lexicon for basic order (21b):

(22)
"These" $= NP/_{\star} NumP$
"five" $= NumP/_{\times} N'$
"young" $= N'\backslash_{\star} N$
"lads" $= N$

This lexicon allows the following derivation and no other for NP via rule (11a):

(23) c. $\dfrac{\text{These}}{NP/_{\star} NumP}$ $\dfrac{\text{lads}}{N}$ $\dfrac{\text{five}}{NumP/_{\times} N'}$ $\dfrac{\text{young}}{N'\backslash_{\star} N}$

$$\cfrac{\cfrac{NP/_{\star} NumP \quad \cfrac{N \quad \cfrac{NumP/_{\times} N' \qquad N'\backslash_{\star} N}{NumP\backslash_{\star} N}{>}\mathbf{B}_{\times}}{NumP}{<}}{NP}{>}}{}$$

This is Cinque's attested order (21c).[17]

---

[17] Depending on the facts concerning coordinations like (15) in the language in question, we may want to use $/_{\diamond\star}$ or $\backslash_{\diamond\star}$ slash types in place of $/_{\star}$ or $\backslash_{\star}$ to allow harmonic composition as well as application in a language like English.

For a language like Shupamem with many orders, in the worst case we can simply allow multiple entries for lexical items supporting all those orders. However, in practice, smaller lexicons with entries supporting more than one order seem to apply. (For example, in Shupamem, Dem seems to bear the completely unrestricted bidirectional slash type $NP|NumP$—Steedman [2020, page 19].)

The VP
Turning to the Germanic VP construction surveyed in Abels (2016), we see the identical picture to that for the elements of the NP (21) emerge for the verb group. That is, only 22 out of the total 24 permutations are allowed:

(24) a.
$$\frac{\text{w}ill}{VP_1/VP_2} \quad \frac{\text{h}elp}{VP_2/VP_3} \quad \frac{\text{t}each}{VP_3/VP_4} \quad \frac{\text{s}wim}{VP_4}$$
Abels: (Basic)

   b.
$$\frac{\text{w}ill}{VP_1/VP_2} \quad \frac{\text{h}elp}{VP_2/VP_3} \quad \frac{\text{s}wim}{VP_4} \quad \frac{\text{t}each}{VP_3\backslash VP_4}$$
Abels: (Basic)

   c.
$$\frac{\text{w}ill}{VP_1/VP_2} \quad \frac{\text{s}wim}{VP_4} \quad \underbrace{\frac{\text{h}elp}{VP_2/VP_3} \quad \frac{\text{t}each}{VP_3\backslash VP_4}}_{VP_2\backslash VP_4}{\scriptstyle >\mathbf{B}_\times}$$
Abels: (from b)

   d.
$$\frac{\text{s}wim}{VP_4} \quad \underbrace{\frac{\text{w}ill}{VP_1/VP_2} \quad \underbrace{\frac{\text{h}elp}{VP_2/VP_3} \quad \frac{\text{t}each}{VP_3\backslash VP_4}}_{VP_1\backslash VP_3}{\scriptstyle >\mathbf{B}}}_{VP_1\backslash VP_4}{\scriptstyle >\mathbf{B}_\times}$$
Abels: (from b)

   e.
$$\underbrace{\frac{\text{h}elp}{VP_2/VP_3} \quad \frac{\text{w}ill}{VP_1\backslash VP_2}}_{VP_1/VP_3}{\scriptstyle <\mathbf{B}_\times} \quad \frac{\text{t}each}{VP_3/VP_4} \quad \frac{\text{s}wim}{VP_4}$$
(from r)

   f.
$$\underbrace{\frac{\text{h}elp}{VP_2/VP_3} \quad \frac{\text{w}ill}{VP_1\backslash VP_2}}_{VP_1/VP_3}{\scriptstyle <\mathbf{B}_\times} \quad \frac{\text{s}wim}{VP_4} \quad \frac{\text{t}each}{VP_3\backslash VP_4}$$
Abels: (from s)

   g. $\times$
$$\underbrace{\frac{\text{h}elp}{VP_2|VP_3} \quad \frac{\text{s}wim}{VP_4}}_{*} \quad \underbrace{\frac{\text{w}ill}{VP_1|VP_2VP_4} \quad \frac{\text{t}each}{VP_3|VP_4}}_{*}{\scriptstyle}$$
Not attested (disallowed)

   h.
$$\frac{\text{s}wim}{VP_4} \quad \underbrace{\underbrace{\frac{\text{h}elp}{VP_2/VP_3} \quad \frac{\text{w}ill}{VP_1\backslash VP_2}}_{VP_1/VP_3}{\scriptstyle <\mathbf{B}_\times} \quad \frac{\text{t}each}{VP_3\backslash VP_4}}_{VP_1\backslash VP_4}{\scriptstyle >\mathbf{B}_\times}$$
Abels: (from s)

   i.
$$\frac{\text{t}each}{VP_3/VP_4} \quad \underbrace{\underbrace{\frac{\text{w}ill}{VP_1/VP_2} \quad \frac{\text{h}elp}{VP_2\backslash VP_3}}_{VP_1\backslash VP_3}{\scriptstyle >\mathbf{B}_\times} \quad \frac{\text{s}wim}{VP_4}}_{VP_1/VP_4}{\scriptstyle <\mathbf{B}_\times}$$
(from n)

j. × $\dfrac{\text{t}each}{VP_3/VP_4}$ $\dfrac{\text{w}ill}{VP_1|VP_2}$ $\dfrac{\text{s}wim}{VP_4}$ $\dfrac{\text{h}elp}{VP_2|VP_3}$

$\underline{\qquad\qquad}*$

$\underline{\qquad\qquad}*$

$\underline{\qquad\qquad\qquad}*$ Not attested (disallowed)

k. $\dfrac{\text{t}each}{VP_3/VP_4}$ $\dfrac{\text{s}wim}{VP_4}$ $\dfrac{\dfrac{\text{w}ill}{VP_1/VP_2}\quad\dfrac{\text{h}elp}{VP_2\backslash VP_3}}{VP_1\backslash VP_3}{>}\mathbf{B}_\times$ Abels: (from n)

l. $\dfrac{\text{s}wim}{VP_4}$ $\dfrac{\text{t}each}{VP_3\backslash VP_4}$ $\dfrac{\dfrac{\text{w}ill}{VP_1/VP_2}\quad\dfrac{\text{h}elp}{VP_2\backslash VP_3}}{VP_2\backslash VP_3}{>}\mathbf{B}_\times$ Abels: (from o)

m. $\dfrac{\text{w}ill}{VP_1/VP_2}$ $\dfrac{\dfrac{\text{t}each}{VP_3/VP_4}\quad\dfrac{\text{h}elp}{VP_2\backslash VP_3}}{VP_2/VP_4}{<}\mathbf{B}_\times$ $\dfrac{\text{s}wim}{VP_4}$ Abels: (from n)

n. $\dfrac{\text{w}ill}{VP_1/VP_2}$ $\dfrac{\text{t}each}{VP_3/VP_4}$ $\dfrac{\text{s}wim}{VP_4}$ $\dfrac{\text{h}elp}{VP_2\backslash VP_3}$ Abels: (Basic)

o. $\dfrac{\text{w}ill}{VP_1/VP_2}$ $\dfrac{\text{s}wim}{VP_4}$ $\dfrac{\text{t}each}{VP_3\backslash VP_4}$ $\dfrac{\text{h}elp}{VP_2\backslash VP_3}$ Abels: (Basic)

p. $\dfrac{\text{s}wim}{VP_4}$ $\dfrac{\text{w}ill}{VP_1/VP_2}$ $\dfrac{\dfrac{\text{t}each}{VP_3\backslash VP_4}\quad\dfrac{\text{h}elp}{VP_2\backslash VP_3}}{\dfrac{VP_2\backslash VP_4}{VP_1\backslash VP_4}{>}\mathbf{B}_\times}{<}\mathbf{B}$ Abels: (from o)

q. $\dfrac{\dfrac{\text{H}elp}{VP_2/VP_3}\quad\dfrac{\text{t}each}{VP_3/VP_4}}{\dfrac{VP_2/VP_4}{VP_1/VP_4}{<}\mathbf{B}_\times}{>}\mathbf{B}$ $\dfrac{\text{w}ill}{VP_1\backslash VP_2}$ $\dfrac{\text{s}wim}{VP_4}$ (from r)

r. $\dfrac{\text{h}elp}{VP_2/VP_3}$ $\dfrac{\text{t}each}{VP_3/VP_4}$ $\dfrac{\text{s}wim}{VP_4}$ $\dfrac{\text{w}ill}{VP_1\backslash VP_2}$ Abels: (Basic)

s. $\dfrac{\text{h}elp}{VP_2/VP_3}$ $\dfrac{\text{s}wim}{VP_4}$ $\dfrac{\text{t}each}{VP_3\backslash VP_4}$ $\dfrac{\text{w}ill}{VP_1\backslash VP_2}$ Abels: (Basic)

t. $\dfrac{\text{s}wim}{VP_4}$ $\dfrac{\dfrac{\text{h}elp}{VP_2/VP_3}\quad\dfrac{\text{t}each}{VP_3\backslash VP_4}}{VP_2\backslash VP_4}{>}\mathbf{B}_\times$ $\dfrac{\text{w}ill}{VP_1\backslash VP_2}$ Abels: (from s)

u. $\dfrac{\text{t}each}{VP_3/VP_4}$ $\dfrac{\dfrac{\text{h}elp}{VP_2\backslash VP_3}\quad\dfrac{\text{w}ill}{VP_1\backslash VP_2}}{\dfrac{VP_1\backslash VP_3}{VP_1/VP_4}{<}\mathbf{B}_\times}{<}\mathbf{B}$ $\dfrac{\text{s}wim}{VP_4}$ (from w)

v. $\dfrac{\overline{\text{t}eetch}}{VP_3/VP_4}$ $\dfrac{\text{h}elp}{VP_2\backslash VP_3}$ $\dfrac{\text{s}wim}{VP_4}$ $\dfrac{\text{w}ill}{VP_1\backslash VP_2}$ (from w)

$$\frac{\text{t}each \quad \text{h}elp \quad \text{s}wim \quad \text{w}ill}{\dfrac{VP_3/VP_4 \quad VP_2\backslash VP_3 \quad VP_4 \quad VP_1\backslash VP_2}{VP_2/VP_4}}{}_{<\mathbf{B}_\times}$$

w. $\dfrac{\text{t}each}{VP_3/VP_4}$ $\dfrac{\text{s}wim}{VP_4}$ $\dfrac{\text{h}elp}{VP_2\backslash VP_3}$ $\dfrac{\text{w}ill}{VP_1\backslash VP_2}$ Abels: (Basic)

x. $\dfrac{\text{s}wim}{VP_4}$ $\dfrac{\text{t}each}{VP_3\backslash VP_4}$ $\dfrac{\text{h}elp}{VP_2\backslash VP_3}$ $\dfrac{\text{w}ill}{VP_1\backslash VP_2}$ Abels: (Basic)

Once again, the two non-separable permutations (24) are among those unattested by Abels for the orders for $VP_1\ VP_2\ VP_3\ VP_4$, including those that he is equivocal toward but which others have claimed. Once again, we can restrict word order for a given language using slash types, and once again the kinds of alternation found for serial verbs in languages like Hungarian can in practice be captured in quite small lexicons using more flexible types (Steedman 2020). Crucially, none of these attested or predicted word orders requires any kind of movement or action-at-a-distance.

Interestingly, the one order predicted under the present hypothesis that was not attested for the NP (21), is among the orders attested by Abels for the Germanic verb group (albeit somewhat grudgingly, as "spontaneously, possibly as alternate," citing Wurmbrand (2004, page 59), who found it accepted by some Austrian German speakers).

Acknowledgments

References

Abels, Klaus. 2016. The fundamental left–right asymmetry in the Germanic verb cluster. *Journal of Comparative Germanic Linguistics*, 19:179–220. DOI: https://doi.org/10.1007/s10828-016-9082-9

Abels, Klaus and Ad Neeleman. 2012. Linear asymmetries and the LCA. *Syntax*, 15:25–74. DOI: https://doi.org/10.1111/j.1467-9612.2011.00163.x

Adger, David. 2013. *A Syntax of Substance*. MIT Press.

Avis, David and Monroe Newborn. 1981. On pop-stacks in series. *Utilitas Mathematica*, 19:129–140.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Research*, San Diego, CA.

Baldridge, Jason and Geert-Jan Kruijff. 2003. Multi-Modal Combinatory Categorial Grammar. In *Proceedings of 11th Annual Meeting of the European Association for Computational Linguistics*, pages 211–218, Budapest. DOI: https://doi.org/10.3115/1067807.1067836

Becker, Tilman, Owen Rambow, and Michael Niv. 1992. The derivational generative power, or, scrambling is beyond LCFRS. Institute for Research in Cognitive Science, University of Pennsylvania.

Bose, Prosenjit, Jonathan F. Buss, and Anna Lubiw. 1998. Pattern matching for permutations. *Information Processing Letters*, 65(5):277–283. DOI: https://doi.org/10.1016/S0020-0190(97)00209-3

Bresnan, Joan W., Ronald M. Kaplan, Stanley Peters, and Annie Zaenen. 1982. Cross-serial dependencies in Dutch. *Linguistic Inquiry*, 13:613–636. DOI: https://doi.org/10.1007/978-94-009-3401-6_11

Cain, Bruce Dwayne. 2000. *Dhivehi (Maldivian): A synchronic and diachronic study*. Ph.D. thesis, Cornell, Ithaca, NY.

Chen-Main, Joan and Aravind K. Joshi. 2008. Flexible composition, multiple adjoining and word order variation. In *Proceedings of the Ninth International Workshop on Tree Adjoining Grammar and Related Frameworks*

(TAG+9), pages 9–16, Tübingen.

Chomsky, Noam. 1957. *Syntactic Structures*. Mouton, The Hague. DOI: `https://doi.org/10.1515/9783112316009`

Chomsky, Noam. 1970. Remarks on nominalization. In R. Jacobs, and P. Rosenbaum, editors, *Readings in English Transformational Grammar*. Ginn, Waltham, MA, pages 184–221. Reprinted in Chomsky (1972).

Chomsky, Noam. 1995a, Bare phrase structure. In Gert Webelhuth, editor, *Government and Binding Theory and the Minimalist Program*. Blackwell, Oxford, pages 383–439.

Chomsky, Noam. 1995b. *The Minimalist Program*. MIT Press, Cambridge, MA.

Chomsky, Noam. 2000. Minimalist inquiries: The framework. In Roger Martin, David Michaels, and Juan Uriagereka, editors, *Step by Step: Essays in Minimalist Syntax in Honor of Howard Lasnik*. MIT Press, Cambridge, MA, pages 89–155.

Chomsky, Noam. 2001. Derivation by phase. In Michael Kenstowicz, editor, *Ken Hale: A Life in Language*. MIT Press, Cambridge, MA, pages 1–52.

Church, Kenneth and Ramesh Patil. 1982. Coping with syntactic ambiguity. *Computational Linguistics*, 8:139–149.

Cinque, Guglielmo. 2005. Deriving Greenberg's universal 20 and its exceptions. *Linguistic Inquiry*, 36:315–332. DOI: `https://doi.org/10.1162/0024389054396917`

Cinque, Guglielmo. 2007. The fundamental left-right asymmetry of natural languages. *Working Papers in Linguistics*, 17:77–107.

Cinque, Guglielmo. 2010. *The Syntax of Adjectives*. Linguistic Inquiry Monograph 57. MIT Press, Cambridge MA. DOI: `https://doi.org/10.7551/mitpress/9780262014168.003.0005`

Cinque, Guglielmo. 2013. On the movement account of Greenberg's universal 20: Refinements and replies: Materials. Manuscript, University of Venice.

Cinque, Guglielmo and Luigi Rizzi. 2008. The cartography of syntactic structures. *Studies in Linguistics: CISCL Working Papers on Language and Cognition*, 2:43–59. DOI: `https://doi.org/10.1093/oxfordhb/9780199544004.013.0003`

Clark, Alexander and Shalom Lappin. 2010. Unsupervised learning and grammar induction. *The Handbook of Computational Linguistics and Natural Language Processing*,

57. DOI: `https://doi.org/10.1002/9781444324044.ch8`

Costello, Nancy. 1969. The Katu noun phrase. *Mon-Khmer Studies*, 3:21–35.

Culbertson, Jennifer and David Adger. 2014. Language learners privilege structured meaning over surface frequency. *Proceedings of the National Academy of Sciences*, 111:5842–5847. DOI: `https://doi.org/10.1073/pnas.1320525111`, PMID: 24706789, PMCID: PMC4000832

Culbertson, Jennifer, Paul Smolensky, and Géraldine Legendre. 2012. Learning biases predict a word order universal. *Cognition*, 122:306–329. DOI: `https://doi.org/10.1016/j.cognition.2011.10.017`, PMID: 22208785

Dong, Li and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin. DOI: `https://doi.org/10.18653/v1/P16-1004`

Dowty, David. 1988. Type-raising, functional composition, and nonconstituent coordination. In Richard Oehrle, Emmon Bach, and Deirdre Wheeler, editors, *Categorial Grammars and Natural Language Structures*. Reidel, Dordrecht, pages 153–198.

Dryer, Matthew. 2018. On the order of demonstrative, numeral, adjective, and noun. *Language*, 94:81–138. DOI: `https://doi.org/10.1353/lan.2018.0072`

Eccles, Peter J. 1997. *An Introduction to Mathematical Reasoning: Numbers, Sets and Functions*. Cambridge University Press. DOI: `https://doi.org/10.1017/CB09780511801136`

Eisner, Jason. 1996. Efficient normal-form parsing for Combinatory Categorial Grammar. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 79–86, Santa Cruz, CA. DOI: `https://doi.org/10.3115/981863.981874`

Giusti, Giuliana. 2006. Parallels in clausal and nominal periphery. In Mara Frascarelli, editor, *Phases of Interpretation*. de Gruyter, Berlin, pages 163–184.

Gunderson, David S. 2014. *Handbook of Mathematical Induction: Theory and Applications*. Chapman and Hall/CRC.

Hawkins, John. 1983. *Word Order Universals*. Academic, New York.

Hockenmaier, Julia and Yonatan Bisk. 2010. Normal-form parsing for Combinatory

Categorial Grammars with generalized composition and type-raising. In *Proceedings of the 23nd International Conference on Computational Linguistics*, pages 465–473, Beijing.

Hockenmaier, Julia and Peter Young. 2008. Non-local scrambling: The equivalence of TAG and CCG revisited. In *Proceedings of the 9th International Workshop on Tree-Adjoining Grammars and Related Formalisms (TAG+9)*, Tübingen.

Jensen, John Thayer. 1977. *Yapese Reference Grammar*. University of Hawaii Press.

Joshi, Aravind, Tilman Becker, and Owen Rambow. 2000. Complexity of scrambling: A new twist to the competence-performance distinction. In Anne Abeillé, and Owen Rambow, editors, *Tree-Adjoining Grammar*. CSLI Publications, Palo Alto, CA, pages 167–181.

Joshi, Aravind, K. Vijay-Shanker, and David Weir. 1991. The convergence of mildly context-sensitive formalisms. In Peter Sells, Stuart Shieber, and Tom Wasow, editors, *Processing of Linguistic Structure*. MIT Press, Cambridge, MA, pages 31–81.

Joshi, Aravind K. 1985. Tree adjoining grammars: How much context-sensitivity is required to provide reasonable structural descriptions? In *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*, Studies in Natural Language Processing, pages 206–250, Cambridge University Press. DOI: `https://doi.org/10.1017/CBO9780511597855.007`

Knuth, Donald E. 1968. *The Art of Computer Programming, Volume 1: Fundamental Algorithms*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA.

Koller, Alexander and Marco Kuhlmann. 2009. Dependency trees and the strong generative capacity of CCG. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 460–468, ACL, Athens. DOI: `https://doi.org/10.3115/1609067.1609118`

Kuhlmann, Marco, Alexander Koller, and Giorgio Satta. 2015. Lexicalization and generative power in CCG. *Computational Linguistics*, 41:187–219. DOI: `https://doi.org/10.1162/COLI_a_00219`

Liebeck, Martin. 2010. *A Concise Introduction to Pure Mathematics*. CRC Press. DOI: `https://doi.org/10.3389/fpsyg.2018.00155`, PMID: 29497394, PMCID: PMC5818723

Medeiros, David P. 2018. ULTRA: Universal Grammar as a Universal Parser. *Frontiers in Psychology*, 9:155. DOI: `https://doi.org/10.3389/fpsyg.2018.00155`, PMID: 29497394, PMCID: PMC5818723

Merlo, Paola. 2015. Predicting word order universals. *Journal of Language Modelling*, 3:317–344. DOI: `https://doi.org/10.15398/jlm.v3i2.112`

Merlo, Paola and Sarah Ouwayda. 2018. Movement and structure effects on universal 20 word order frequencies: A quantitative study. *Glossa*, 3:1–35. DOI: `https://doi.org/10.5334/gjgl.149`

Mosel, Ulrike. 2017. Teop–an oceanic language with multifunctional verbs, nouns and adjectives. *Studies in Language*, 41:255–293. DOI: `https://doi.org/10.1075/sl.41.2.02mos`

Nchare, Abdoulaye Laziz. 2012. *The Grammar of Shupamem*. Ph.D. thesis, New York University.

Partee, Barbara and Mats Rooth. 1983. Generalised conjunction and type ambiguity. In Rainer Bäuerle, Christoph Schwarze, and Arnim von Stechow, editors, *Meaning, Use, and Interpretation of Language*. de Gruyter, Berlin, pages 361–383.

Senft, Gunter. 1986. *Kilivila: The Language of the Trobriand Islanders*. Mouton de Gruyter. DOI: `https://doi.org/10.1515/9783110861846`

Shapiro, Louis and A. B. Stephens. 1991. Bootstrap percolation, the Schroder numbers, and the N-kings problem. *SIAM Journal on Discrete Mathematics*, 4(2):275–280. DOI: `https://doi.org/10.1137/0404025`

Stabler, Edward. 1996. Derivational minimalism. In *International Conference on Logical Aspects of Computational Linguistics*, pages 68–95, Nancy. DOI: `https://doi.org/10.1007/BFb0052152`

Stabler, Edward. 2011. Computational perspectives on minimalism. In Boeckx, Cedric, editor, *Oxford Handbook of Linguistic Minimalism*. Oxford University Press, pages 617–641. DOI: `https://doi.org/10.1093/oxfordhb/9780199549368.013.0027`

Stanojević, Miloš and Khalil Sima'an. 2015. Reordering grammar induction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 44–54, Lisbon. DOI: `https://doi.org/10.18653/v1/D15-1005`

Steedman, Mark. 1985. Dependency and coordination in the grammar of Dutch and English. *Language*, 61:523–568. DOI:

https://doi.org/10.2307/414385

Steedman, Mark. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA. DOI: https://doi.org/10.7551/mitpress/6591.001.0001

Steedman, Mark. 2005. Interfaces and the grammar. In *Proceedings of the 24th West Coast Conference on Formal Linguistics*, pages 19–33, Vancouver.

Steedman, Mark. 2012. *Taking Scope: The Natural Semantics of Quantifiers*. MIT Press, Cambridge, MA. DOI: https://doi.org/10.7551/mitpress/9780262017077.001.0001, PMCID: PMC3082409

Steedman, Mark. 2014. The surface-compositional semantics of English intonation. *Language*, 90:2–57. DOI: https://doi.org/10.1353/lan.2014.0010, https://doi.org/10.1353/lan.2014.0022

Steedman, Mark. 2020. A formal universal of natural language grammar. *Language*, 96:618–660. DOI: https://doi.org/10.1353/lan.2020.0043

Svenonius, Peter. 2007. 1...3–2. In Ramchand and Charles Reiss, editors, *The Oxford Handbook of Linguistic Interfaces*. Oxford University Press, pages 239–287.

Szabolcsi, Anna. 1983. The possessor that ran away from home. *Linguistic Review*, 3:89–102. DOI: https://doi.org/10.1515/tlir.1983.3.1.89

Szabolcsi, Anna. 1994. The noun phrase. In Ferenc Kiefer and Katalin É. Kiss, editors, *Syntax and Semantics 27: The Syntactic Structure of Hungarian*. Academic Press, New York, pages 179–274. DOI: https://doi.org/10.1163/9789004373174_004

Tryon, Darrell. 1967. The Languages of the Loyalty Islands. Ph.D. thesis, *The Australian National University*. DOI: https://doi.org/10.1002/j.1834-4461.1967.tb00937.x

Weisstein, Eric. 2018. Schröder number. In *MathWorld*. A Wolfram Web Resource.

West, Julian. 1996. Generating trees and forbidden subsequences. *Discrete Mathematics*, 157(1-3):363–374. DOI:

https://doi.org/10.1016/S0012-365X(96)83023-8

Whitelock, Peter. 1992. Shake-and-bake translation. In *Proceedings of the 14th conference on Computational linguistics*, pages 784–791, Nantes. DOI: https://doi.org/10.3115/992133.992190

Williams, Edwin. 2003. *Representation Theory*. MIT Press. DOI: https://doi.org/10.7551/mitpress/5893.001.0001

Wu, Dekai. 1996. A polynomial-time algorithm for statistical machine translation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, ACL '96, pages 152–158, Stroudsburg, PA. DOI: https://doi.org/10.3115/981863.981884

Wurmbrand, Susi. 2004. West Germanic verb clusters: The empirical domain. In Katalin É. Kiss and Henk van Riemsdijk, editors, *Verb Clusters: A Study of Hungarian, German, and Dutch*. John Benjamins, Amsterdam, pages 43–85. DOI: https://doi.org/10.1075/la.69.05wur

Wurmbrand, Susi. 2006. Verb clusters, verb raising, and restructuring. In Martin Evaraert and Henk van Riemsdijk, editors, *The Blackwell Companion to Syntax*, volume V. Blackwell, Oxford, pages 229–343. DOI: https://doi.org/10.1002/9780470996591.ch75

Zhang, Hao and Daniel Gildea. 2005. Stochastic Lexicalized Inversion Transduction Grammar for alignment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 475–482, Ann Arbor, MI. DOI: https://doi.org/10.3115/1219840.1219899

Zhang, Hao and Daniel Gildea. 2007. Factorization of Synchronous Context-Free Grammars in linear time. In *NAACL Workshop on Syntax and Structure in Statistical Translation (SSST)*, pages 25–32, Rochester, NY. DOI: https://doi.org/10.3115/1626281.1626285