Dissociating functor-argument structure from surface phrase structure: the relationship of HPSG Order Domains to LFG

Abstract

Recent work on *order domains* and *linearization* in HPSG, and also conceptually related work in categorial grammar, is argued to mirror the key ideas that motivated the separation between c-structure and f-structure in the earliest work in LFG (Bresnan 1982). This paper argues that: (i) to be descriptively adequate, all constraint-based frameworks do indeed need a dissociation between functor-argument structure and surface phrase structure, (ii) while linearization HPSG can technically be regarded as a one-level/one-stratum architecture, the complex attribute-value matrix representations proposed are better thought of as encoding multiple 'virtual levels', and (iii) once this is noted, there is a general equivalence between the analyses that can be proposed in linearization HPSG and LFG, although certain differences are noted.

The modern linguistic frameworks Generalized Phrase Structure Grammar (GPSG: Gazdar et al. 1985), Head-driven Phrase Structure Grammar (HPSG: Pollard and Sag 1994) and Categorial Grammar (CG: Moortgat 1988) grew out of a structuralist tradition that assumed a rather close relationship between units of functor-argument structure and units of surface phrase structure – indeed they hewed to this line closely, since they denied the need for transformational operations, which much of modern linguistics has used to produce dissociations between the two (as in Chomsky (1957) and subsequent work). In contrast, Lexical Functional Grammar (LFG: Bresnan 1982) was designed to easily permit a much looser relationship between these two notions, and, recently, cases of marked deviation between these two levels have led researchers in HPSG to adopt an *order domain* or *linearization* framework (Reape 1993, Reape 1994, Reape forthcoming, Pollard et al. 1993, Kathol 1995b, Kathol and Pollard 1995a, Kathol and Pollard 1995b, Calcagno 1993). Similar considerations motivate related proposals by Gunji (1995) for JPSG and by Dowty (forthcoming), Moortgat (forthcoming), and Morrill (1995) for Categorial Grammar(-like) frameworks.[1]

When comparing different frameworks, the syntactic literature normally focuses on differences between them, while ignoring the great commonalities that are often present (if somewhat hidden by differences of notation). In contrast, this paper attempts to bring out the commonality between all this recent work, and the close connections with the motivations that led to the separation of constituent structure (c-structure) from functional structure (f-structure) in the earliest work on LFG – even though this parallel seems either to have been not noted or misunderstood by practitioners of linearization. It is shown

[1]This paper has benefited from thoughtful comments from Joan Bresnan, Bob Carpenter, Andreas Kathol, Gerald Penn, Ivan Sag, and attendees of a Syntax/Semantics reading group at Carnegie Mellon University. Special thanks to Ivan Sag for catching an analytical oversight in a previous version. I hope it is now somewhat clearer what I am trying to do.

that the main accounts that have been developed in HPSG using linearization, in particular accounts of German word order, can be modeled in much the same way utilizing the separation of c-structure from f-structure in LFG. Such LFG accounts do require certain theoretical innovations, such as functional uncertainty and functional precedence, that were not present in 1982 LFG, but we can compare HPSG accounts of German word order to the LFG account of Zaenen and Kaplan (Kaplan and Zaenen 1988, Zaenen and Kaplan 1995), which uses these tools. The LFG version arguably requires less machinery, since the need for separating c-structure from f-structure played a greater role in the development of the formalism. On the other hand, the addition of functional precedence to LFG can be seen as permitting tighter links between functor-argument structure and word order (which were arguably too dissociated in early LFG), and the HPSG account perhaps offers a neater statement of certain extraposition possibilities.

1 Reape's proposal and the basic comparison

While an account of English can work quite well by assuming an isomorphism between functor-argument units and units of surface phrase structure (modulo a few interesting extraposition phenomena), it has long been realized that such an isomorphism is untenable for many languages. For instance, elements of the same noun phrase may be discontinuous in a Warlpiri sentence (1a), and elements of embedded clauses are interleaved in the Dutch sentence (1b):

(1) a.

Karli	ka	*pirli-ngka*	nguna-mi	*wita-ngka*
boomerang-ABS	PRES	rock-LOC	lie-NPST	small-LOC

'A boomerang is lying on a *small rock*.'

b.

dat	ik	Marie	Henk	de	nijlpaarden	zag	helpen	voeren
that	I	Marie	Henk	the	hippos	saw	help	feed

'that I saw Marie help Henk feed the hippos'

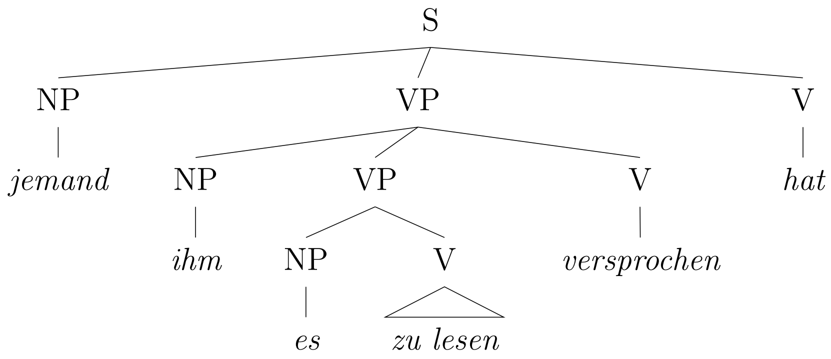
The essence of the "order domain" or "linearization" approach is to allow more flexible surface word order possibilities by separating out a level where surface word order is represented from another level of syntax where deeper relationships (grammatical functions, subcategorization or functor-argument structure) are represented. In HPSG work, the standard theory of HPSG becomes the deeper level, while a new attribute of signs, DOM(AIN), is introduced, which has as its value surface word order relationships. Certain confusions arise simply over the question of how to name these two levels. In (2), I show the preferred names that will be used in this paper, along with various other names that have appeared in the literature.

(2)	Surface phrase structure	Functor-argument structure
Reape	((word) order) domain (tree)	syntax tree
Curry, Dowty	phenogrammar	tectogrammar
Kathol, Pollard	order domain	composition structure
Vanilla HPSG		"syntax"
LFG	c-structure	f-structure
PTQ (Montague 1973)	output of syntactic rules	analysis tree
'Flexible' CG	prosodics	proof tree[2]

Modulo this kind of notational issue, there are close correspondences beteen order domain analyses and corresponding analyses that could be suggested in LFG. As an initial example, for the non-extraposed verbal complement sentence (3a), Reape (forthcoming) proposes the syntax tree in (3b) for which the top level domain is as in (3c):

(3) a. daß es ihm jemand zu lesen versprochen hat
that it.ACC him.DAT someone.NOM to read promised has
'that someone promised him to read it'

b.



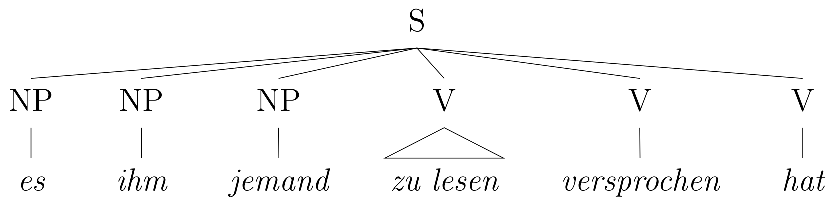
c. [$_{\text{S}}$ [$_{\text{NP}}$ *es*] [$_{\text{NP}}$ *ihm*] [$_{\text{NP}}$ *jemand*] [$_{\text{V}}$ *zu lesen*] [$_{\text{V}}$ *versprochen*] [$_{\text{V}}$ *hat*]]

Compare this with the outlines of a possible LFG analysis f-structure in (4a) and c-structure in (4b). Reape uses a tree notation for the functor-argument structure and square brackets for the surface phrase structure, where LFG has used the opposite convention, but nothing substantive follows from this. Assuming a similar linguistic approach to the syntax of German, Reape's structures are isomorphic to the ones that would be proposed in LFG.

(4) a.
$$\begin{bmatrix} \text{SUBJ} & \textit{jemand} \\ \text{XCOMP} & \begin{bmatrix} \text{OBJ} & \textit{ihm} \\ \text{XCOMP} & \begin{bmatrix} \text{OBJ} & \textit{es} \\ \text{PRED} & \textit{zu lesen} \end{bmatrix} \\ \text{PRED} & \textit{versprochen} \end{bmatrix} \\ \text{PRED} & \textit{hat} \end{bmatrix}$$

[2]CG differs from other frameworks by there not necessarily being only one proof tree for an (unambiguous) sentence. But the notion can perhaps be recaptured in terms of the canonical proof tree (Morrill 1994).

b.



Reape (1994, forthcoming) completely flattens both the sequence of NPs and the sequence of verbs in such a clause as a single domain. In contrast, Zaenen and Kaplan (1995) have a similar flat c-structure representation for the NPs but keep the verbs nested. In particular, Zaenen and Kaplan (1995:fn. 2) discuss but argue against, a flat structure for the verbs, while Reape (1994:164–165) suggests that there is little evidence for verb nesting in the order domain. But whether we accept either or neither of these possibilities is a question of our substantive linguistic theory and the empirical support there is for it. Given the separation of functor-argument structure from surface phrase structure, both of these accounts, and a variety of other options, are available in both frameworks.[3]

In certain respects, functor-argument structure and surface phrase structure become similar to deep and surface structures that have been posited in transformational grammar, but without postulating a transformational relationship between them. For instance, the structures just suggested for German are quite similar to the deep and surface structures suggested by Evers (1975). Looking more closely, the relationship between these two levels mimics the work commonly done by *scrambling* movements in GB or transformational grammar, while other classes of movements (such as passive, raising, and long distance dependencies) continue to be modeled in a quite different way.

2 Order domains and how they are built

Reape (1993, forthcoming) introduced the notion of "order domains" into HPSG.[4] Kathol (1995b) states the motivation for using order domains as "to allow for a certain degree of decoupling of the linear representation from the combinatorial structure of a phrase".[5] I would submit that this is the same motivation that led to the loose relationship between c-structure and f-structure in LFG. The idea was to have a largely uniform underlying functor-argument structure for diverse languages, at which constraints on grammatical relations could be stated, despite the vastly varying surface phrase structure of languages such as English, Malayalam, and Warlpiri that were explored in early LFG investigations. If there is a difference it is mainly one of theoretical vision: while this recent work in HPSG aims "for a

[3] LFG certainly does not require flat surface phrase structure trees: nested X′ trees, resembling those typical in GB work are posited for instance in King (1995) and Bresnan (2001:Ch. 7).

Kathol and much other work in HPSG does not accept Reape's analysis of these verb complexes, but rather adopts an 'argument composition' (division categories) analysis in the style of Hinrichs and Nakazawa (1994). Such an analysis should be compared with corresponding work on complex predicates in LFG (Alsina 1993, Butt 1995), but this further cross-framework comparison is beyond the scope of this paper.

[4] This idea is conceptually similar to the liberation metarules suggested for GPSG by Zwicky (1986), but differs by clearly introducing separate representations of functor-argument structure and surface phrase structure, whereas Zwicky's proposal uses metarules to generate flattened structures.

[5] Note, crucially, that by "a phrase", Kathol is refering to a unit in the functor-argument structure.

certain degree of decoupling", the LFG framework began by postulating a total decoupling, and some recent work can be seen as incorporating greater coupling back into the framework.

In the linearization approach, the functor-argument structure of a sentence is built up according to the predicate-argument and adjunct-modification relationships of the sentence. For a language like English, such a tree structure may be isomorphic to the surface phrase structure tree that can be motivated in terms of linear order and tests for surface constituency, but this is not true for languages such as German and Warlpiri where the two diverge. Nevertheless, in all cases the functor-argument-structure tree reflects functor-argument relationships, and a second structure, order domains, is introduced to describe surface phrase structure constituency.

The domain structure of a sentence is built up in parallel with the functor-argument structure, in a rule-to-rule approach, and is represented in an additional attribute of HPSG signs of sort *phrase*, namely the DOM(AIN) attribute. An example from Kathol (1995b) is shown in (5). At each level of combining partial trees in the functor-argument structure, a new order domain is created for the parent node, formed from the DOM values of the children. In Reape's approach, there are two possibilities for how the DOM value of the parent is formed from the DOM values of the children. One possibility is that the mother node should act in future as a single domain element. For example, a German NP gives rise to a single domain element, which is *opaque* in the sense that adjacency and order relations holding within it cannot be disturbed by subsequent addition of other domain objects while building up the order domain of the sentence. This is achieved by making a new domain object for the mother, which represents this surface constituent, an operation that is sometimes referred to as *freezing* or *compaction*.

(5) a. Sah Adam die Rose
saw Adam the rose
'Did Adam see the rose?'

b.

$$\begin{bmatrix} s[\textsc{subcat}\ \langle\ \rangle] \\ \textsc{dom} \quad \left\langle \begin{bmatrix} \textsc{phon}\langle \textit{sah}\rangle \\ v \end{bmatrix}, \begin{bmatrix} \textsc{phon}\ \langle \textit{Adam}\rangle \\ np[\textsc{nom}] \end{bmatrix}, \begin{bmatrix} \textsc{phon}\ \langle \textit{die Rose}\rangle \\ np[\textsc{acc}] \end{bmatrix} \right\rangle \end{bmatrix}$$

$$\begin{bmatrix} vp[\textsc{subcat}\ \langle \textsc{np}[\textsc{nom}]\rangle] \\ \textsc{dom} \left\langle \begin{bmatrix} \textsc{phon}\langle \textit{sah}\rangle \\ v \end{bmatrix}, \begin{bmatrix} \textsc{phon}\ \langle \textit{die Rose}\rangle \\ np[\textsc{acc}] \end{bmatrix} \right\rangle \end{bmatrix} \qquad \begin{bmatrix} np[\textsc{nom}] \\ \textsc{dom} \left\langle \begin{bmatrix} \textsc{phon}\ \langle \textit{Adam}\rangle \\ n \end{bmatrix} \right\rangle \end{bmatrix}$$

$$\begin{bmatrix} v[\textsc{sbct}\ \langle \textsc{np}[\textsc{n}],\ \textsc{np}[\textsc{a}]\rangle] \\ \textsc{dom} \left\langle \begin{bmatrix} \textsc{phon}\ \langle \textit{sah}\rangle \\ v \end{bmatrix} \right\rangle \end{bmatrix} \qquad \begin{bmatrix} np[\textsc{acc}] \\ \textsc{dom} \left\langle \begin{bmatrix} \textsc{phon}\ \langle \textit{die}\rangle \\ det \end{bmatrix}, \begin{bmatrix} \textsc{phon}\ \langle \textit{Rose}\rangle \\ n[\textsc{acc}] \end{bmatrix} \right\rangle \end{bmatrix}$$

The other possibility is that the domain of the parent node is simply a list of the domain elements of all the children, in which case these elements can be linearized independently in the next higher domain. In this way functor-argument structure units can become discontinuous in the surface phrase structure. For example, the functor-argument structure VP *sah die Rose* becomes discontinous in the top-level order domain in (5b). This operation is referred to as *domain union*, and it allows elements that are not sisters in functor-argument structure to be linearly ordered with respect to each other, contrary to standard HPSG. In general, it allows us to keep an invariant underlying structure while generating flatter surface structures with various word orders. Note, by way of comparison, that this possibility for functional units not to be units of surface phrase structure has always been present in LFG, perhaps most famously in the description of Warlpiri (Simpson 1983, Simpson 1991).

In the work of Reape, the internal structure of opaque constituents is still represented in the top level domain, and can be trivially mapped to an equivalent tree representation, as discussed above. However, in Kathol and Pollard's work, the internal structure of opaque domains, such as the NP *die Rose*, is hidden via the compaction operation. Pollard et al. (1993) and Dowty (forthcoming) write as if their surface level is simply a string of words on which linear precedence constraints are imposed, and hence is more superficial than LFG's c-structure – perhaps equivalent to the input string level mentioned in passing by Kaplan (1987). But in fact their work crucially employs mechanisms of *compaction* or *attachment* that clump multiple words into constituents and this is what makes order domains equivalent to LFG's c-structure. Indeed, I would say that opaque domains still effectively *have* internal structure – it is just that one has to look down to the node where compaction took place in order to see what that internal structure is. Thus the nature of, and the evidence for order domain constituency closely matches that for c-structure constituency in LFG (modulo certain minor questions of information architecture – e.g., the case feature appearing in the order domain in (4b) would not standardly appear in an LFG c-structure tree). LFG c-structures were designed to represent exactly the same surface word order information: c-structures show the "superficial arrangement of words and phrases in the sentence" and "are defined in terms of syntactic categories, terminal strings, and their dominance and precedence relationships" (Kaplan and Bresnan 1982:175).[6]

Moreover, evidence used to motivate a certain order domain structure in HPSG (versus evidence for functor-argument structure) also closely corresponds to evidence for c-structure in LFG. For instance, this would seem true of the data on coordination, stress, and Verum focus that Kathol and Pollard (1995b) mention. The data they discuss concerning the presence of agreement endings (clitics) on non-verbs would perhaps seem to be more of an f-structure issue in LFG, but perhaps this, too, can be seen to be related to c-structure if the agreement endings on complementizers are to be correctly regarded as a clitic edge inflection (Miller 1991, Halpern 1992).[7]

3 Levels

In GPSG (Gazdar et al. 1985), there is only one level of syntactic structure,[8] the surface constituency of a sentence, and standard HPSG (Pollard and Sag 1987, Pollard and Sag

[6]This close relationship to LFG work appears to be misunderstood in Kathol and Pollard (1995a:174) and Pollard et al. (1993:2).

[7]See Bresnan (2001) for a treatment of West Flemish inflected complementizers within LFG.

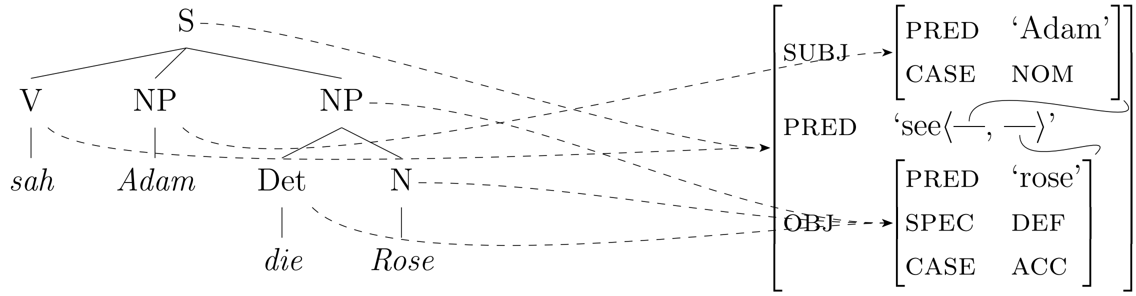
[8]And only one stratum, if we accept the distinction between levels and strata proposed by Ladusaw (1988).

1994) can be seen as inheriting this tradition. This stands in contrast not only to GB, but to a theory like LFG where there are multiple parallel levels of recursively generated syntactic structure, whose nodes cannot be mapped one-to-one onto each other.

However, the introduction of the DOMAIN feature creates an extra structure within an HPSG *sign* rich enough to be analogous to a flat c-structure, and, considering the entire derivation tree, these DOMAIN values represent something like a hierarchical c-structure. If one has rich enough model objects, any conception of a monostratal/monolevel grammar is almost impossible to define. In HPSG, it is always possible to 'hide' multiple levels within different parts of the architecture of the sign. While one could argue that adding order domains into HPSG is just a further elaboration of each node in the same one-level monostratal grammatical organization of early HPSG, I wish to argue that that is not an insightful way to look at things. Order domains should be looked at as a new 'virtual level' (even though coded up as an attribute of phrasal signs) and the introduction of order domains into HPSG can thus be seen as sending HPSG in the direction of LFG. Perhaps the key differentiation that makes a new attribute really a covert level is when that attribute imposes a hierarchical structure on a sentence that is different from other ones present in the formalism. And that is exactly what the DOMAIN attribute does.

A remaining difference between LFG and linearization HPSG is that in the former all the surface phrase structure information is placed in one structure (the c-structure) and all the functor-argument information is placed in another structure (the f-structure), and that nodes in these two are then related by the correspondence function ϕ (shown by dotted lines below). In contrast, in HPSG the surface phrase structure and functor-argument structure of a group of words is placed in a single node (the *sign*) and then there is a tree-structure over these signs. These differences can be seen by contrasting (5b) with the corresponding LFG representation in (6). Some possible advantages of the HPSG-style representation have been discussed in the LFG literature by Dalrymple et al. (1992) and by Andrews and Manning (1993). On the other hand, note that the intermediate order domains that appear for the VP and V nodes in the HPSG-style representation in (5b) appear to be epiphenomenal. The only justification for them is oiling the implementational wheels, whereas the LFG c-structure represents only what we would hope to be able to justify empirically about surface phrase structure. Nevertheless, these intermediate order domains are at present crucial to the theory, since it makes use of linear precedence constraints inherited from intermediate order domains, as is discussed in the next section.

(6)



4 Linear precedence

In linearization HPSG, the elements of each order domain are (partially) ordered according to the linear precedence (LP) rules of the language concerned. Additionally, this ordering must also respect any prior ordering constraints that are inherited monotonically from the children's order domains.[9] Reape's account of word order in German uses category information to order NPs in front of verbs (7a), features like INV and EXTRA to handle V2 and right-extraposed clauses, and finally a rule like (7b) to ensure the leftward embedding order of infinitives (unless extraposed, a verb precedes a verb that governs it). (7b) looks at the functor-argument structure of the sign and works because all verbs in the domain of a head verb are governed by it.[10] Observe crucially that this is an LP constraint that must be applied at each node of the functor-argument structure and the partial orderings imposed by it must then be inherited when ordering higher order domains; it isn't an ordering constraint that can just be applied to the top level sign.

(7) a. $_{sign}[\text{DOM}\ ] \;\Rightarrow\; _{sign}[\text{DOM}\quad \text{NP} \prec \text{V}]$

b. $_{sign}[\text{DTRS}\,|\,\text{HEAD-DTR}\quad \boxed{1}\text{V}[\text{INV}\quad -]] \;\Rightarrow\; _{sign}[\text{DOM}\quad \text{V} \prec \boxed{1}\ ]$

Dutch non-extraposed VPs also have all NPs preceding all verbs, but have the opposite ordering of verbs (1b). Reape proposes capturing this by replacing (7b) with (8) for Dutch:

(8) $_{sign}[\text{DTRS}\,|\,\text{HEAD-DTR}\quad \boxed{1}\text{V}[\text{INV}\quad -]] \;\Rightarrow\; _{sign}[\text{DOM}\quad \boxed{1} \prec \text{V}\ ]$

However, in contrast with the fairly free order of NPs within the German Mittelfeld, in Dutch the order of the object NPs must match the order of the corresponding verbs, generating the well-known cross-serial dependencies of Dutch (Evers 1975). The account of Reape (forthcoming) does not account for this NP ordering, although further linear precedence rules of the type of (7b) that refer to both functor-argument structure and domain structure could be added that would effect this ordering.[11]

Let us briefly contrast the above with an LFG account using the notion of functional precedence (Bresnan 1984, Bresnan 1994, Kaplan and Zaenen 1989a). An LFG c-structure rule can express ordering constraints, and they can even be written in the ID/LP format of GPSG (Gazdar et al. 1985), or in any other form that yields a finite state expansion of possible categories on the right-hand side of the rule. However, sometimes one wants to impose ordering constraints that may be longer distance than a single phrase structure rule or which depend on functional (i.e., f-structure) information. In order to do this, the notion of f-precedence was introduced. F-precedence allows statements to be made about the ordering of words depending on the f-structure that they map onto via the correspondence

[9] I will not dwell on the technicalities of how this is all implemented declaratively here. See Reape (1994) or Kathol and Pollard (1995a).

[10] Note that Reape's LP constraints are being defined over the whole *sign* structure.

[11] Reape (forthcoming) questions this characterization suggesting that "[in Dutch,] objects can in fact sometimes interchange ... where the semantics or pragmatics of the sentence makes it clear which object fulfills which object role" but the only example he gives is actually of exchanging the two objects of a ditransitive verb. Exchanging the order of objects of different verbs does not seem possible (thanks to Henriëtte de Swart for help with Dutch data).

function ϕ shown in (6). The first version of the proposal (Bresnan 1984, Kaplan 1987) was a global ordering constraint – it specified an ordering relationship over the entire string.

However, this does not allow one to order things in the German *Mittelfeld* while not ruling out grammatical sentences types. For example, one wants to specify that NPs precede verbs in the Mittelfeld, while still allowing a verb to appear in front of NPs if it is in the V2 position, or an NP to appear after the final verbs if it is extraposed. Zaenen and Kaplan (1995) propose as a solution *relativized f-precedence*, according to which f-precedence constraints can be relative to a certain c-structure node which gives the domain of application. In Zaenen and Kaplan (1995), the constraints are relative to VP (which represents their Mittelfeld). Giving one node a privileged status like this seems somewhat ad hoc. A natural generalization is to say that f-precedence statements can be relative to any node of the c-structure. If one adopts flat c-structure trees matching the flat domain trees of Reape's proposal, we can make f-precedence statements relative to *every* node, and this gives a system that allows pretty much identical linear precedence constraints to Reape's framework.[12]

The c-structure rules will generate a flat phrase structure, but the NPs and verbs in this flat structure may be deeply embedded complements in functor-argument structure. Use of the functional uncertainty notation of LFG will allow the necessary mapping to embedded functor-argument structures (Kaplan and Zaenen 1989b).[13] Functional uncertainty constraints will then interact with the functional precedence expressions to constrain legitimate mappings to be only ones that obey the ordering constraints discussed above. The LFG rule to order the NPs and verbs for Dutch in a flat Reape-style analysis is shown in (9) (cf. Zaenen and Kaplan (1995) for the corresponding rules if verbs are nested). The second bottom line of (9) orders the verbs, and the bottom line states the constraint on NP ordering by saying that a direct argument of a verb's complement clause cannot precede an direct argument belonging to its own clause. Note that the use of functional uncertainty obviates the need for applying LP rules to intermediate order domains and then monotonically preserving the resulting partial orders (the whole motivation for Reape's *sequence union* or *shuffle* operation). Instead all LP constraints can simply be applied to the final surface phrase structure of the sentence.

(9)
$$\begin{array}{lcc} \mathrm{VP} \rightarrow & \mathrm{NP}^* & \mathrm{V}^* \\ & (\uparrow\mathrm{XCOMP}^*\ \mathrm{GF}) = \downarrow & (\uparrow\mathrm{XCOMP}^*) = \downarrow \\ & & (\uparrow\mathrm{XCOMP}^+) \not<_f \uparrow \\ & & (\uparrow\mathrm{XCOMP}^+\ \mathrm{GF}) \not<_f (\uparrow\mathrm{GF}) \end{array}$$

5 Topological fields and extraposition

There are certain differences between the proposals of Reape and those of Kathol and Pollard. Reape uses n-ary branching functor-argument structures, whereas Kathol and Pollard (1995b) take functor argument structure to be binary. Reape uses complete *sign*s as domain

[12] A remaining difference is that functional precedence only allows ordering on the basis of f-structure attributes, whereas Reape can order constituents on the basis of any attributes, such as ones that would appear in the c-structure or σ-structure (semantics) of LFG.

[13] This notation allows the possible positions in functor argument structure of a phrase structure unit to be expressed as a regular set.

elements, which predicts that word order can be influenced by any part of the sign. Kathol and Pollard assume a sparse information structure for domain objects, which minimally contains categorial and phonological information. It seems clear that Reape's proposal is too unconstrained in this respect. A principled theory of word order should limit the features that are accessible to LP rules. Necessary features seem to include category, subcategorization relationships, a few features like INVERTED and EXTRAPOSED, the latter of which could be unified with discourse notions like FOCUS, and perhaps some notion of phonological weight. Many other attributes (like whether a verb is transitive or ditransitive, or whether an adjective is modified) do not seem to affect word order.

The most important difference, however, is that while in Reape's work the domain structure is nothing more than a tree structure (or its equivalent in nested lists), Kathol and Pollard (Kathol 1995b, Kathol and Pollard 1995b) introduce the traditional Germanic *topological fields*, which represent German sentences as shown in (10):

(10)

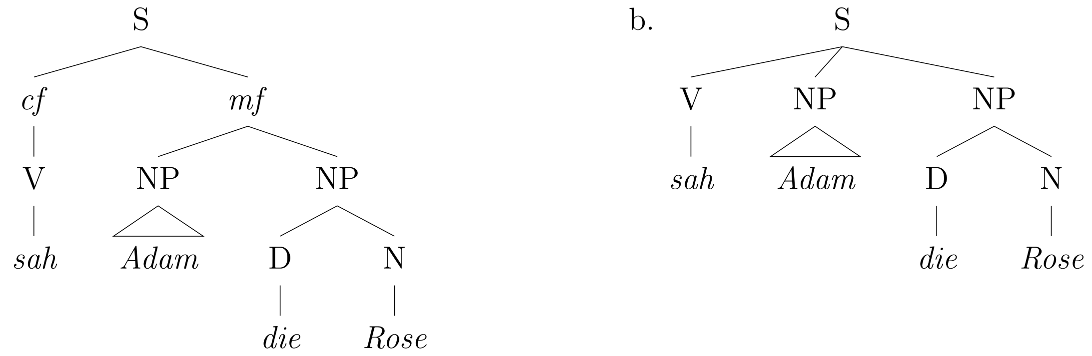
	Vorfeld	line Satzklammer	Mittelfeld	rechte Satzklammer	Nachfeld
	[Spec, CP]	Comp/Vfin		final verbs	extraposed
	vf	*cf*	*mf*	*vc*	*nf*
Vfinal		daß	Adam eine Rose	gesehen hat	
V2	eine Rose	hat	Adam	gesehen	
V1		hat	Adam eine Rose	gesehen	

The topological fields must appear linearly ordered as shown (although certain fields may be empty). We can directly model these topological fields in LFG by suggesting that German surface phrase structure (i.e., the c-structure) can be modeled by the top-level phrase structure rule:

(11) S $\rightarrow$ *vf* *cf* *mf* *vc* *nf*

If we then provided phrase structure rules for the *vf*, *mf* and so on in the usual way, we would capture the ordering constraints of topological fields satisfactorily, but the resulting phrase structure trees would look somewhat unusual, as there would be extra nodes for each field. For example, the tree for (5) would look like (12a):

(12) a.



Such an approach makes no distinction between topological fields, and the traditional notion of syntactic categories. As argued by Ahrenberg (1989), this is a serious defect:

the *Mittelfeld* is not a constituent in the sense in which an NP is. There is no sense in which it acts as a unit with respect to classical constituency tests such as topicalization or postposing. However, we can get rid of these unwanted nodes by regarding the topological field designators as metacategories (Kaplan and Maxwell 1993). The idea of metacategories is that a symbol may be introduced to express an important grouping of constituents, which is nevertheless not a node in the phrase structure tree. Metacategories are indicated by using an equals sign (=) to express their expansion rather than an arrow (→). Using metacategories, the final c-structure will be as in (12b).[14]

Kathol and Pollard also avoid introducing unwanted constituency into the order domain, but they do so by making TOPO(LOGY) an attribute of domain objects. Certain types in the lexicon are given a restricted assignment to topological fields, as shown in (13), and there are certain other restrictions (14).[15]

(13)

Vfin	*cf* or *vc*
Comp	*cf*
Filler (NPwh)	*vf* or *cf*

(14) a. [TOPO *vf*] $\prec$ [TOPO *vf*]

b. [TOPO *cf*] $\prec$ [TOPO *cf*]

c. Clause constraint: In every finite clause in German, the *cf* topological field is instantiated.

The equivalent in LFG is done simply by writing the appropriate rules for the metacategories:

(15) a. *vf* = (NP|NPwh)

b. *cf* = C|Vfin|NPwh

c. *mf* = NP*

d. *vc* = V*

Again, the degree of equivalence between these notations should be manifest, even though the factorization of information is somewhat different. The ordering of the topological fields was expressed in the rule expanding the S node. The fact that there can only be one thing in each of the *vf* and *cf* fields is coded in the expansions of those metacategories. The clause constraint is realized by making *cf* have to expand to something.[16]

[14]This proposal will be sufficient for the present paper, but see Ahrenberg (1989) for a more thoroughgoing attempt to combine a topological field organization of the surface phrase structure with an LFG account of functor-argument structure. In Ahrenberg's account, topological fields become first class members of the syntactic ontology.

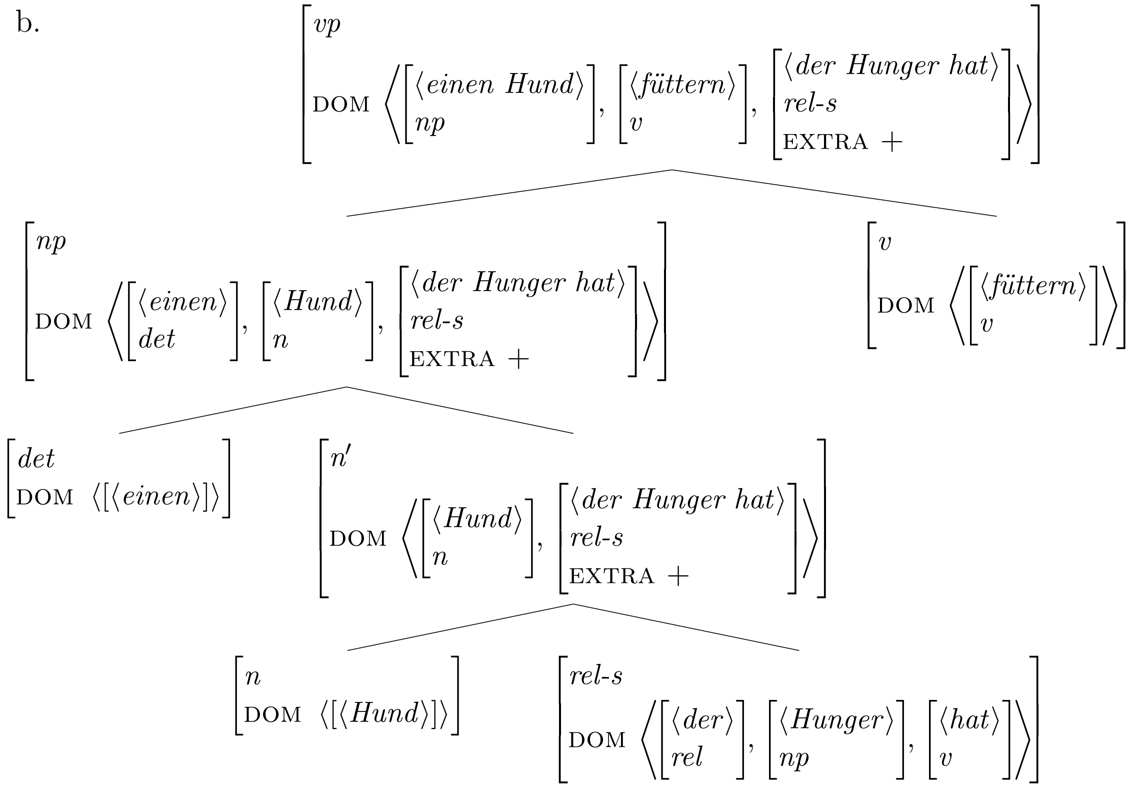
[15](14a) and (14b) are a baroque way of saying that the *vf* and *cf* fields may each contain at most one element.

[16]Note that nothing in this system prevents one from incorrectly placing the verb finally in root (as well as embedded) *Wh*-questions, providing that the *Wh*-word is placed in the *cf*: [*cf* was] [*mf* Adam] [*vc* sieht] 'Who did Adam see?'. This defect of the account presented in the cited papers is inherited by a direct translation of it into LFG. Kathol (1995a) avoids this overgeneration by placing further constraints on the realization of certain clause types such as *root* and *subord*.

Instead of allowing only *freezing* and *domain union* as operations to build the order domain of a parent node, a fourth innovation of Kathol and Pollard (1995a) is to propose a general operation of *partial compaction* which allows some children to become part of a domain object for the parent node, while others escape compaction and get to operate independently in higher order domains. They apply this notion to a treatment of extraposition. As shown in (16b), the extraposed phrase *der Hunger hat* is placed inside the NP in the functor-argument structure, but because it escapes compaction when the top domain structure is built, it is allowed to float to a rightwards position in the surface phrase structure.

(16) a. einen Hund füttern der Hunger hat
a dog feed that hunger has
'feed a dog that is hungry'

b.



An LFG account of extraposition might be somewhat more difficult to state. It appears that we would have to enumerate possible landing sites for the extraposed element and then, via a functional uncertainty expression, describe all the possible places that element could belong in the functor-argument structure. It is possible that this would not be too difficult, and much of the statement might follow from universal statements of phrase structure and long distance movement possibilities of the sort explored in Bresnan (2001), but it appears more difficult and laborious to control precisely than in the HPSG analysis just outlined.

I see this difference as stemming from the fact that LFG is built around a parsing metaphor, while HPSG, especially linearization approaches, is built around a generation metaphor. The natural statement of LFG is that one builds the c-structure for a sentence, and then derives the f-structure from it, while the natural statement of linearization HPSG is that one builds the functor-argument structure of the sentence and then derives the domain structure from it. Given that we both want to parse and generate with these syntactic frameworks, we hope that these processes are reversible (and indeed they are), but there is no guarantee that a certain relationship might not be more neatly statable, or more easily determinable in one direction than the other, and perhaps extraposition is an example of this. An idea underlying the development of both HPSG and LFG was that a declarative rather than a procedural characterization of grammar would provide a process-independent description of linguistic theory, not biased to either parsing or generation (Sag 1995). But, while a useful first step, providing a declarative description is not sufficient to ensure directional neutrality.[17]

6 Conclusions

This paper emphasizes the compatibility of ideas between the use of order domains in HPSG and related theories and the separation between c-structure and f-structure that has been developed in LFG. The major conclusions are:

1. All constraint-based theories have to develop mechanisms to dissociate word order and surface constituency from semantic or functor-argument structure units. The naive assumption of an isomorphism between the two is tenable for much of English, but clearly does not extend well to many other languages. Such mechanisms cover roughly the same empirical ground as the subcategory of movements sometimes labeled *scrambling* in transformational frameworks.[18]

2. Technically, a linearization HPSG analysis can be thought of as involving only one level and one stratum of syntactic description. However, when extremely articulated attribute-value matrix representations of linguistic units are proposed, as in HPSG, this technical statement no longer makes conceptual sense. One wants to ask what 'virtual levels' are present in a certain sign architecture. Functor-argument structure and linearization domains are best thought of as two virtual levels, and compared with the corresponding levels of f- and c-structure in LFG.

3. There is a general equivalence between possible analyses utilizing the separation of c-structure and f-structure in LFG and the separation between functor-argument structure and order domains in HPSG, although certain important qualifications have been noted.

This attempt to bring out the similarities and differences between certain HPSG and LFG proposals should be seen only as a beginning, however. Once we observe these close correspondences between the two theories, it is natural to ask in what ways insights can be

[17] Perhaps the simplest example of this is that most of modern cryptography is based on the fact that the declarative relationship "C is the product of the prime numbers A and B" is easy to calculate in one direction, but extremely difficult to calculate in the other direction (finding A and B given C).

[18] In another strand of work, the need for this dissociation is reflected by introducing tangled trees (McCawley 1982, Huck 1985, Ojeda 1987, Blevins 1990).

shared between them, and how various combined theories that develop ideas from both can be developed. On the other hand we might also wonder about some of the other differences. For instance, the nature of the functor argument structure is very different in HPSG versus LFG and we might wonder whether one representation or the other is better motivated. This appears to be a productive area for further research.

Bibliography

Ahrenberg, L. 1989. A formal field grammar. Technical Report LiTH-IDA-R-98-46, Institionen för datavetenskap, Linköping University, Sweden.

Alsina, A. 1993. *Predicate Composition: A Theory of Syntactic Function Alternations*. PhD thesis, Stanford.

Andrews, A. D., and C. D. Manning. 1993. Information spreading and levels of representation in LFG. Technical Report CSLI-93-176, CSLI, Stanford CA.

Blevins, J. 1990. *Syntactic Complexity: Evidence for Discontinuity and Multidomination*. PhD thesis, University of Massachusetts, Amherst.

Bresnan, J. (Ed.). 1982. *The Mental Representation of Grammatical Relations*. Cambridge, MA: MIT Press.

Bresnan, J. 1984. Bound anaphora on functional structures. Invited paper presented at the Tenth Annual Meeting of the Berkeley Linguistics Society.

Bresnan, J. 1994. Linear order vs. syntactic rank: Evidence from weak crossover. In K. Beals, J. Denton, B. Knippen, L. Melnar, H. Suzuki, and E. Zeinfeld (Eds.), *Papers from the 30th Regional Meeting of the Chicago Linguistics Society*, Vol. 1. Chicago Linguistics Society.

Bresnan, J. 2001. *Lexical-Functional Syntax*. Oxford: Blackwell.

Butt, M. 1995. *The Structure of Complex Predicates in Urdu*. Stanford: CSLI Publications.

Calcagno, M. 1993. Toward a linearization-based approach to word order variation in Japanese. In A. Kathol and C. Pollard (Eds.), *Papers in Syntax*, no. 42 in OSU Working Papers in Linguistics, 26–45. Ohio State University: Department of Linguistics.

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

Dalrymple, M., K. Halvorsen, R. Kaplan, C. Manning, J. Maxwell, J. Wedekind, and A. Zaenen. 1992. Relating projections. MS, Xerox PARC, Palo Alto.

Dowty, D. forthcoming. Towards a minimalist theory of syntactic structure. In W. Sijtsma and A. van Horck (Eds.), *Discontinuous Constituency*. Berlin: Mouton de Gruyter.

Evers, A. 1975. *The Transformational Cycle in Dutch and German*. PhD thesis, Rijksuniversiteit, Utrecht, Holland.

Gazdar, G., E. Klein, G. K. Pullum, and I. A. Sag. 1985. *Generalized phrase structure grammar: a theoretical synopsis*. Oxford: Basil Blackwell.

Gunji, T. 1995. On lexical treatment of Japanese causatives—a linearization approach. MS, Osaka University. To appear in G. Green and R. Levine (Eds.), *Readings in HPSG*.

Halpern, A. L. 1992. *Topics in the Placement and Morphology of Clitics*. PhD thesis, Stanford University.

Hinrichs, E., and T. Nakazawa. 1994. Linearizing AUXs in German verbal complexes. In J. Nerbonne, K. Netter, and C. Pollard (Eds.), *German in Head-Driven Phrase Structure Grammar*, no. 46 in Lecture Notes, 11–37. Stanford University: CSLI Publications.

Huck, G. 1985. Exclusivity and discontinuity in phrase structure grammar. In *West Coast Conference on Formal Linguistics*, Vol. 4, 92–98, Stanford University. CSLI Publications/SLA.

Kaplan, R. M. 1987. Three seductions of computational psycholinguistics. In P. Whitelock, H. Somers, P. Bennett, R. Johnson, and M. M. Wood (Eds.), *Linguistic Theory and Computer Applications*, 149–181. London: Academic Press.

Kaplan, R. M., and J. Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In J. Bresnan (Ed.), *The Mental Representation of Grammatical Relations*, 173–281. Cambridge, MA: MIT Press.

Kaplan, R. M., and J. T. Maxwell. 1993. LFG Grammar Writer's Workbench. Technical report, Xerox PARC.

Kaplan, R. M., and A. Zaenen. 1988. Functional uncertainty and functional precedence in Continental West Germanic. In *Proceedings of Wiener Workshop Wissensbasierte Sprachverarbeitung*, Vienna, Austria.

Kaplan, R. M., and A. Zaenen. 1989a. Functional precedence and constituent structure. In C. Huang and K. Chen (Eds.), *Proceedings of ROCLING II*, 19–40, Taipei, Republic of China.

Kaplan, R. M., and A. Zaenen. 1989b. Long-distance dependencies, constituent structure, and functional uncertainty. In M. Baltin and A. Kroch (Eds.), *Alternative Conceptions of Phrase Structure*. Chicago University Press.

Kathol, A. 1995a. *Linearization-based German Syntax*. PhD thesis, Ohio State University.

Kathol, A. 1995b. Verb-'movement' in German and topological fields. In *Papers from the 31st Regional Meeting of the Chicago Linguistic Society*, Vol. 31, Chicago, Illinois.

Kathol, A., and C. Pollard. 1995a. Extraposition via complex domain formation. In *Proceedings of the Thirty-Third Annual Meeting of the Association for Computational Linguistics*, 174–180.

Kathol, A., and C. Pollard. 1995b. On the left periphery of German subordinate clauses. In *West Coast Conference on Formal Linguistics*, Vol. 14, Stanford University. CSLI Publications/SLA.

King, T. H. 1995. *Configuring Topic and Focus in Russian*. Stanford, CA: CSLI Publications.

Ladusaw, W. A. 1988. A proposed distinction between *levels* and *strata*. In *Linguistics in the Morning Calm 2: Selected Papers from SICOL-1986*, 37–51, Seoul. The Linguistic Society of Korea, Hanshin.

McCawley, J. 1982. Parentheticals and discontinuous constituent structure. *Linguistic Inquiry* 13:91–106.

Miller, P. H. 1991. *Clitics and Constituents in Phrase Structure Grammar*. PhD thesis, Utrecht. Published by Garland, New York, 1992.

Montague, R. 1973. The proper treatment of quantification in ordinary English. In J. Hintikka, J. Moravcsik, and P. Suppes (Eds.), *Approaches to Natural Language*. Dordrecht: D. Reidel.

Moortgat, M. 1988. *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus*. Dordrecht: Foris.

Moortgat, M. forthcoming. Generalized quantifiers and discontinuous type constructors. In W. Sijtsma and A. van Horck (Eds.), *Discontinuous Constituency*. Berlin: Mouton de Gruyter.

Morrill, G. 1994. *Type-Logical Grammar*. Dordrecht: Kluwer Academic.

Morrill, G. 1995. Discontinuity in categorial grammar. *Linguistics and Philosophy* 18:175–219.

Ojeda, A. 1987. Discontinuity, multidominance and unbounded dependency in Generalized Phrase Structure Grammar. In A. E. Ojeda and G. J. Huck (Eds.), *Discontinuous Constituency*, Vol. 20 of *Syntax and Semantics*. New York: Academic Press.

Pollard, C., R. Kasper, and R. Levine. 1993. Studies in constituent ordering: Toward a theory of linearization in Head-Driven Phrase Structure Grammar. Grant Proposal to the National Science Foundation, Ohio State University.

Pollard, C., and I. A. Sag. 1987. *Information-Based Syntax and Semantics*. Vol. 1. Stanford, CA: Center for the Study of Language and Information.

Pollard, C., and I. A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Chicago, IL: University of Chicago Press.

Reape, M. 1993. *A Formal Theory of Word Order: A Case Study in West Germanic*. PhD thesis, University of Edinburgh.

Reape, M. 1994. Domain union and word order variation in German. In J. Nerbonne, K. Netter, and C. Pollard (Eds.), *German in Head-Driven Phrase Structure Grammar*, no. 46 in Lecture Notes, 151–197. Stanford University: CSLI Publications.

Reape, M. forthcoming. Getting things in order. In W. Sijtsma and A. van Horck (Eds.), *Discontinuous Constituency*. Berlin: Mouton de Gruyter.

Sag, I. A. 1995. Taking performance seriously. MS, Stanford University.

Simpson, J. H. 1983. *Aspects of Warlpiri morphology and syntax*. PhD thesis, MIT.

Simpson, J. H. 1991. *Warlpiri Morpho-Syntax: A Lexicalist Approach*. Dordrecht: Kluwer.

Zaenen, A., and R. M. Kaplan. 1995. Formal devices for linguistic generalizations: West Germanic word order in LFG. In M. Dalrymple, R. M. Kaplan, J. T. Maxwell III, and A. Zaenen (Eds.), *Formal Issues in Lexical-Functional Grammar*, 215–239. Stanford, CA: CSLI.

Zwicky, A. 1986. Concatenation and liberation. In *Papers from the 22nd Regional Meeting of the Chicago Linguistic Society*, 65–74, Chicago, Illinois. Chicago Linguistics Society.