

Chapter 10

Constituent order

Stefan Müller

Humboldt-Universität zu Berlin

This chapter discusses local ordering variants and how they can be analyzed in HPSG. So-called scrambling, the local reordering of arguments of a head, can be accounted for by assuming flat rules or binary branching rules with arbitrary order of saturation. The difference between SVO and SOV is explained by assuming different mappings between the argument structure list (a list containing all arguments of a head) and valence features for subjects and complements. The position of the finite verb in initial or final position in languages like German can be accounted for by flat rules and a separation between immediate dominance and linear precedence information or by something analogous to head-movement in transformational approaches. The chapter also addresses the analysis of languages allowing even more freedom than just scrambling arguments. It is shown how one such language, namely Warlpiri, can be analyzed with so-called constituent order domains allowing for discontinuous constituents. I discuss problems of domain-based approaches and provide an alternative account of Warlpiri that does not rely on discontinuous constituents.

1 Introduction

This chapter deals with constituent order, with a focus on local order variants. English is the language that is treated most thoroughly in theoretical linguistics but is probably also a rather uninteresting language as far as the possibilities of reordering constituents is concerned: the order of subject, verb, and object is fixed in sentences like (1):

- (1) Kim likes bagels.

Of course, there is the possibility to front the object as in (2) but this is a special, non-local construction that is not the topic of this chapter but is treated in **chapters/udc**, Chapter ?? of this volume.

- (2) Bagels, Kim likes.

This chapter deals with scrambling (the local reordering of arguments) and with alternative placements of heads (called *head movement* in some theories). Examples of the former are the subordinate clauses in (3) and an example of the latter is given in (4):

- (3) a. [weil] der Mann dem Kind das Buch gibt (German)
 because the.NOM man the.DAT child the.ACC book gives
 b. [weil] der Mann das Buch dem Kind gibt
 because the.NOM man the.ACC book the.DAT child gives
 c. [weil] das Buch der Mann dem Kind gibt
 because the.ACC book the.NOM man the.DAT child gives
 d. [weil] das Buch dem Kind der Mann gibt
 because the.ACC book the.DAT child the.NOM man gives
 e. [weil] dem Kind der Mann das Buch gibt
 because the.DAT child the.NOM man the.ACC book gives
 f. [weil] dem Kind das Buch der Mann gibt
 because the.DAT child the.ACC book the.NOM man gives
- (4) Gibt der Mann dem Kind das Buch? (German)
 gives the.NOM man the.DAT child the.ACC book
 ‘Does the man give the child the book?’

(3) shows that in addition to the unmarked order in (3a) (see **Hoehle82a** on the notion of unmarked order), five other argument orders are possible in sentences with three-place verbs. As with the examples just given, I will use German if a phenomenon does not exist in English. Section 6.2 discusses examples from Warlpiri, a language having even freer constituent order.

(4) shows that the verb is placed in initial position in questions in German. This contrasts with the verb-final order in the subordinate clause in (3a), which has the same order as far as the arguments are concerned. This alternation of verb placement is usually treated as head movement in the transformational literature (**Bach62a**; **Bierwisch63a**; **Reis74a**; **Thiersch78a**). Declarative main clauses in German are V2 clauses and the respective fronting of the preverbal constituent is usually treated as a non-local dependency (see **chapters/udc**, Chapter ?? of this volume). Hence, V2 sentences will not be handled here.

The following sections explore the theoretical options within the HPSG framework for dealing with these phenomena. I first discuss the separation of grammar

rules into an immediate dominance part and a linear precedence component in Section 2 and then flat vs. binary branching structures (Section 3). While flat structures allow verbs to be ordered clause-finally or clause-initially, this is not the case for binary branching structures, since only sisters can be ordered. So, for (3a) one would get the bracketing in (5a). If *das Buch* ‘the book’ and *gibt* ‘gives’ are ordered in a different order, (5b) results.

- (5) a. [weil] [der Mann [dem Kind [das Buch gibt]]]
 because the.NOM man the.DAT child the.ACC book gives
 b. * [weil] [der Mann [dem Kind [gibt das Buch]]]
 because the.NOM man the.DAT child gives the.ACC book

Hence, local reordering is not sufficient to get clause-initial verb order and therefore, proposals with binary branching structures are usually paired with HPSG’s analogue of what is head-movement in transformational theories. These are explained in Section 5. Section 6 introduces an extension to standard HPSG developed by Reape94a: constituent order domains. Such constituent order domains allow for discontinuous constituents and have been used to account for languages like Warlipri (DS99a). In contrast, Section 7 shows how such languages can be analyzed without admitting discontinuous constituents.

2 ID/LP format

HPSG was developed out of Generalized Phrase Structure Grammar (GPSG) and Categorical Grammar. The ideas concerning linearization of daughters in a local tree were taken over from GPSG (GKPS85a). In GPSG a separation between immediate dominance and linear precedence is assumed. So, while in classical phrase structure grammar, a phrase structure rule like (6) states that the NP[nom], NP[dat] and NP[acc] have to appear in exactly this order, this is not the case in GPSG and HPSG:

- (6) $S \rightarrow \text{NP}[\text{nom}] \text{NP}[\text{dat}] \text{NP}[\text{acc}] V$

The HPSG schemata corresponding to the immediate dominance rule (ID rule) in (6) do not express information about ordering. Instead, there are separate linear precedence (LP) rules (also called linearization rules). A schema like (6) licenses 24 different orders: the six permutations of the three arguments that were shown in (3) and all possible placements of the verb (to the right of NP[acc], between NP[dat] and NP[acc], between NP[nom] and NP[dat], to the left of NP[nom]). Orders like NP[nom], NP[dat], V, NP[acc] are not attested in German and hence

these orderings have to be filtered out. This is done by linearization rules, which can refer to features or to the function of a daughter in a schema. (7) shows some examples of linearization rules:

- (7) a. $X < V$
 b. $X < V[INI-]$
 c. $X < \text{Head } [INI-]$

The first rule says that all constituents have to precede a V in the local tree. The second rule says that all constituents have to precede a V that has the INITIAL value $-$. One option to analyze German would be the one that was suggested by **Uszkoreit87a** within the framework of GPSG: one could allow for two linearization variants of finite verbs. So in addition to the INI $-$ variant of verbs there could be a INI $+$ variant and this variant would be linearized initially. This reduces the number of permutations licensed by (6) and LP rules to 12: verb-initial placement and 6 permutations of the NPs and verb-final placement with 6 permutations of the arguments. The ID rule in (6) together with the two linearization rules linearizing the verb in initial or final position therefore licenses the same orders as the following twelve phrase structure rules would do:

- (8) a. $S \rightarrow \text{NP}[\text{nom}] \text{NP}[\text{acc}] \text{NP}[\text{dat}] V$
 $S \rightarrow \text{NP}[\text{nom}] \text{NP}[\text{dat}] \text{NP}[\text{acc}] V$
 $S \rightarrow \text{NP}[\text{acc}] \text{NP}[\text{nom}] \text{NP}[\text{dat}] V$
 $S \rightarrow \text{NP}[\text{acc}] \text{NP}[\text{dat}] \text{NP}[\text{nom}] V$
 $S \rightarrow \text{NP}[\text{dat}] \text{NP}[\text{nom}] \text{NP}[\text{acc}] V$
 $S \rightarrow \text{NP}[\text{dat}] \text{NP}[\text{acc}] \text{NP}[\text{nom}] V$
 b. $S \rightarrow V \text{NP}[\text{nom}] \text{NP}[\text{acc}] \text{NP}[\text{dat}]$
 $S \rightarrow V \text{NP}[\text{nom}] \text{NP}[\text{dat}] \text{NP}[\text{acc}]$
 $S \rightarrow V \text{NP}[\text{acc}] \text{NP}[\text{nom}] \text{NP}[\text{dat}]$
 $S \rightarrow V \text{NP}[\text{acc}] \text{NP}[\text{dat}] \text{NP}[\text{nom}]$
 $S \rightarrow V \text{NP}[\text{dat}] \text{NP}[\text{nom}] \text{NP}[\text{acc}]$
 $S \rightarrow V \text{NP}[\text{dat}] \text{NP}[\text{acc}] \text{NP}[\text{nom}]$

Note that we do not need a linearization rule for every ID rule. For example, in a grammar with rules for intransitive, transitive, and ditransitive verbs, head ordering is taken care of by general LP rules of the type in (7b) applying to the respective ID rules. The LP rule in (7c) is even more general than (7b) in that it does not mention the part of speech but instead refers to the function of the constituent. The rule says that a head that has the INI value ‘ $-$ ’ has to be lin-

earized to the right of all other elements in the local tree. Hence, it also applies to adjectives, nouns, and prepositions and their dependents.

This separation of linearization rules from phrase structure rules also makes it possible to capture other generalizations. For example, short elements tend to precede heavy constituents (Behaghel’s Law of Increasing Constituents, Behaghel109a)■
 Uszkoreit87a captured one aspect of this more general rule by formulating a linearization statement requiring that pronouns precede non-pronouns. The LP rules apply a large set of ID rules, for example for intransitive, transitive and ditransitive verbs. By factoring out the LP constraints, generalizations over the whole set of phrase structure rules are covered. Uszkoreit’s constraints on the order of arguments in the so-called *Mittelfeld* (that is, for rules like (8)) are assumed to be violable. While violable constraints are not part of the standard HPSG formalism, this is something desirable and something that is worked on. See also Abeillé & Godard’s work on weight-based linearization and the (reduced) mobility of various categories: bare nominals in various languages, certain pronouns (AG99a-u), certain adverbs (AG2001a-u), negation (AG97a-u; AG2004a-u), and attributive adjectives (AG99b-u). In various papers, Abeillé & Godard propose a three valued WEIGHT feature to account for the ordering of light, middle-weight and heavy constituents (AG2000a; AG2004a-u). See also chapters/complex-predicates■
 Chapter ?? of this volume on complex predicates and weight.

This treatment of constraints on linearization has an advantage that was already pointed out by researchers working in GPSG: it captures the generalizations regarding linearization. For instance, the order of verbs with respect to their arguments is the same in embedded sentences in German, independent of the finiteness of the verb:

- (9) a. dass er dem Mann das Buch gab
 that he.NOM the.DAT man the.ACC book gave
 ‘that he gave the man the book’
 b. dass er versucht, [dem Mann das Buch zu geben]
 that he.NOM tried the.DAT man the.ACC book to give
 ‘that he tried to give the man the book’

This is also true for the relative order of dative and accusative object in (9). The constraints regarding linearization hold across rules. By factoring these constraints out, the generalizations can be captured. See Uszkoreit87a for weighted constraints for the ordering of constituents in the *Mittelfeld*.

Furthermore, cross-linguistic generalizations about constituent structure can be captured. For example, the two phrase structure rules in (10) would be needed

for head-initial and head-final languages, respectively:

- (10) a. $VP \rightarrow V NP NP$
 b. $VP \rightarrow NP NP V$

In an ID/LP framework only one ID rule is needed to describe both sorts of languages. The linearization of the head is factored out of the rules.

Similarly, HPSG has just one schema for Head-Adjunct structures, although languages like English allow adjuncts to precede or follow the head they modify. Schema 1 does not say anything about the order of the daughters.

Schema 1 (Head-Adjunct Schema)

head-adjunct-phrase \Rightarrow

$$\left[\begin{array}{l} \text{HEAD-DTR} \quad \left[\text{SYNSEM } \boxed{1} \right] \\ \text{NON-HEAD-DTRS} \left\langle \left[\begin{array}{l} \text{SYNSEM} | \text{LOC} | \text{CAT} \quad \left[\begin{array}{l} \text{HEAD} | \text{MOD } \boxed{1} \\ \text{SPR } \langle \rangle \\ \text{COMPS } \langle \rangle \end{array} \right] \end{array} \right] \right\rangle \end{array} \right]$$

There is a head daughter and a list of non-head daughters. The respective daughters are specified as the value of a feature or as an element in a list but they are not ordered with respect to each other in the schema. Ordering is taken care of by two LP rules saying that adjuncts marked as pre-modifiers (e.g., attributive adjectives) have to precede their head while those that are marked as post-modifiers (noun-modifying prepositions) follow it:

- (11) a. $\text{Adjunct}[\text{PRE-MODIFIER } +] < \text{Head}$
 b. $\text{Head} < \text{Adjunct}[\text{PRE-MODIFIER } -]$

In general, there are two options for two daughters: head-initial and head-final order. Examples are given in (12):¹

- (12) a. head-initial: example:
- $$\left[\begin{array}{l} \text{PHON} \quad \boxed{1} \oplus \boxed{2} \\ \text{HEAD-DTR} \left[\text{PHON } \boxed{1} \right] \\ \text{NH-DTRS} \left\langle \left[\text{PHON } \boxed{2} \right] \right\rangle \end{array} \right] \quad \left[\begin{array}{l} \text{PHON} \quad \langle \text{ squirrel, from, America } \rangle \\ \text{HEAD-DTR} \left[\text{PHON } \langle \text{ squirrel } \rangle \right] \\ \text{NH-DTRS} \left\langle \left[\text{PHON } \langle \text{ from, America } \rangle \right] \right\rangle \end{array} \right]$$
- b. head-final: example:
- $$\left[\begin{array}{l} \text{PHON} \quad \boxed{2} \oplus \boxed{1} \\ \text{HEAD-DTR} \left[\text{PHON } \boxed{1} \right] \\ \text{NH-DTRS} \left\langle \left[\text{PHON } \boxed{2} \right] \right\rangle \end{array} \right] \quad \left[\begin{array}{l} \text{PHON} \quad \langle \text{ grey, squirrel } \rangle \\ \text{HEAD-DTR} \left[\text{PHON } \langle \text{ squirrel } \rangle \right] \\ \text{NH-DTRS} \left\langle \left[\text{PHON } \langle \text{ grey } \rangle \right] \right\rangle \end{array} \right]$$

¹ \oplus (append) is a relational constraint that concatenates two lists.

When linearization rules enforce head-initial order, as in the case of modification by a PP in English, the PHON value of the head daughter is concatenated with the PHON value of the non-head daughter, and if the order has to be the other way around as in the case of adjectives modifying nouns, the non-head daughter is concatenated with the head daughter. An adjective is specified as PRE-MODIFIER + and a preposition as PRE-MODIFIER -. Since these features are head-features (see **chapters/properties**, Chapter ?? of this volume on head features), they are also accessible at the level of adjective phrases and prepositional phrases.

check
chapter

For languages with free variation in head-adjunct order, it would suffice to not state any LP rule and one would get both orders with the same Head-Adjunct schema. So, the separation of immediate dominance and linear precedence allows for an underspecification of order. Therefore HPSG grammarians are not forced to assume several different constructions for attested patterns or derivational processes that derive one order from another more basic one.

3 Flat and binary branching structures

The previous section discussed LP rules and used flat phrase structure rules for illustration. The corresponding flat structures are also used in HPSG. Schema 2 shows a Head-Complement schema that combines a head with all the complements selected via the COMPS list.²

²GSag2000a-u assume a list called DTRS for all daughters including the head daughter. It is useful to be able to refer to specific non-head daughters without having to know a position in a list. For example in head-adjunct structures the adjunct is the selector. So I keep DTRS for a list of ordered daughters and HEAD-DTR and NON-HEAD-DTRS for material that is not necessarily ordered.

Schema 2 (Head-Complement Schema)

head-complement-phrase \Rightarrow

$$\left[\begin{array}{l} \text{SYNSEM|LOC|CAT|COMPS } \langle \rangle \\ \text{HEAD-DTR } \left[\text{SYNSEM|LOC|CAT|COMPS } \langle \underline{1} \rangle \right] \\ \text{NON-HEAD-DTRS } \text{synsem2signs}(\underline{1}) \end{array} \right]$$

synsem2signs is a relational constraint mapping a list of *synsem* objects as they are contained in the *COMPS* list onto a list of objects of type *sign* as they are contained in daughters (see **GSag2000a-u** for a similar proposal).³ How this schema can be used to analyze VPs like the one in (13) is shown in Figure 1.

(13) Kim gave Sandy a book.

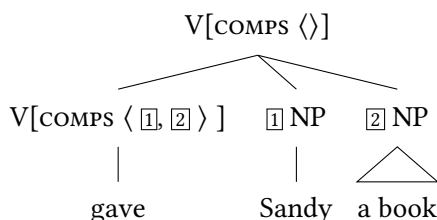


Figure 1: Analysis of the VP *gave Sandy a book* with a flat structure

HPSG differs from purely phrase structure-based approaches in that the form of a linguistic object is not simply the concatenation of the forms associated with the terminal symbols in a tree (words or morphemes). Every linguistic object has its own phonology representation. So in principle one could design theories in which the combination of *Mickey Mouse* and *sleeps* is pronounced as *Donald Duck laughs*. Of course, this is not done. The computation of the *PHON* value of the mother is dependent of the *PHON* values of the daughters. But the fact that the *PHON* values of a linguistic sign are not necessarily a strict concatenation of the *PHON* values of the daughters can be used to model languages having a less strict order than English. **ps** formulate the Constituent Order Principle, which is given as Principle 1 in adapted form:

³In Sign-Based Construction Grammar the objects in valence lists are of the same type as the daughters. A relational constraint would not be needed in this variant of the HPSG theory. Theories working with a binary branching Head-Complement Schema as Schema 3 on page **xi** would not need the relational constraint either, since the *synsem* object in the *COMPS* list can be shared with the *SYNSEM* value of the element in the list of non-head daughters directly.

Principle 1 (Constituent Order Principle)

$$phrase \Rightarrow \left[\begin{array}{l} \text{PHON order-constituents}(\boxed{1}) \\ \text{DTRS } \boxed{1} \end{array} \right]$$

DTRS is a list of all daughters including the head daughter (if there is one). This setting makes it possible to have the daughters in the order in which the elements are ordered in the COMPS list (primary object, secondary object, and obliques) and then compute a PHON value in which the secondary object precedes the primary object. French is a language with freer constituent order than English and such flat structures with appropriate reorderings are suggested by AG2000a. For English the function order-constituents would just return a concatenation of the PHON values of the daughters, but for other languages it would be much more complicated. In fact this function and its interaction with linear precedence constraints was never worked out in detail.

Researchers working on English and French usually assume a flat structure (ps2Sag97aGSag2000a-uAG2000a) but assuming binary branching structures would be possible as well, as is clear from analyses in Categorical Grammar, where binary combinatory rules are assumed (Ajdukiewicz35a-u; Steedman2000a-u). For languages like German it is usually assumed that structures are binary branching (but see Reape94a and BvN98a). The reason for this is that adverbials can be placed anywhere between the arguments, as the following example from Uszkoreit87a shows:

- (14) *Gestern hatte in der Mittagspause der Vorarbeiter in der*
 yesterday had during the lunch.break the foreman in the
Werkzeugkammer dem Lehrling aus Boshaftigkeit langsam zehn
 tool.shop the apprentice maliciously slowly ten
schmierige Gußeisenscheiben unbemerkt in die Hosentasche gesteckt.
 greasy cast.iron.disks unnoticed in the pocket put
 ‘Yesterday during the lunch break, the foreman maliciously put ten
 greasy cast iron disks slowly into the apprentice’s pocket unnoticed.’

A way to straightforwardly analyze adjunct placement in German and Dutch is to assume that adjuncts can attach to any verbal projection. For example, Figure 2 shows the analysis of (15):

- (15) *weil deshalb jemand gestern dem Kind schnell das Buch gab*
 because therefore somebody yesterday the child quickly the book gave
 ‘because somebody quickly gave the child the book yesterday’



Figure 2: Analysis of [weil] *deshalb jemand gestern dem Kind schnell das Buch gab* ‘because somebody quickly gave the child the book yesterday’ with binary branching structures

The adverbials *deshalb* ‘therefore’, *gestern* ‘yesterday’ and *schnell* ‘quickly’ may attach to any verbal projection. For example, *gestern* could also be placed at the other adjunct positions in the clause.

Binary branching structures with attachment of adjuncts to any verbal projection also account for recursion and hence the fact that arbitrarily many adjuncts can attach to a verbal projection. Of course it is possible to formulate analyses with flat structures that involve arbitrarily many adjuncts (Kasper94a; Noord94AG2000aBMS2001a), but these analyses involve relational constraints in schemata or in lexical items or an infinite lexicon. In Kasper’s analysis, the relational constraints walk through lists of daughters of unbounded length in order to compute the semantics. In the other three analyses, (some) adjuncts are treated as valents, which may be problematic because of scope issues. This cannot be dealt with in detail here, but see LH2006a and Chaves2009a for dis-

cussion.

The following schema licenses binary branching head-complement phrases:

Schema 3 (Head-Complement Schema (binary branching))

head-complement-phrase \Rightarrow

$$\left[\begin{array}{l} \text{SYNSEM|LOC|CAT|COMPS } \boxed{1} \oplus \boxed{2} \\ \text{HEAD-DTR} \quad \left[\text{SYNSEM|LOC|CAT|COMPS } \boxed{1} \oplus \langle \boxed{3} \rangle \oplus \boxed{2} \right] \\ \text{NON-HEAD-DTRS } \langle \left[\text{SYNSEM } \boxed{3} \right] \rangle \end{array} \right]$$

The COMPS list of the head daughter is split into three lists: a beginning ($\boxed{1}$), a list containing $\boxed{3}$ and a rest ($\boxed{2}$). $\boxed{3}$ is identified with the SYNSEM value of the non-head daughter. All other elements of the COMPS list of the head daughter are concatenated and the result of this concatenation ($\boxed{1} \oplus \boxed{2}$) is the COMPS list of the mother node. This schema is very general. It works for languages that allow for scrambling, since it allows an arbitrary element to be taken out of the COMPS list of the head daughter and realize it in a local tree. The schema can also be “parameterized” to account for languages with fixed word order. For head-final languages with fixed order, $\boxed{2}$ would be the empty list (= combination with the last element in the list) and for head-initial languages with fixed order (e.g., English), $\boxed{1}$ would be the empty list (= combination with the first element in the list). Since the elements in the COMPS list are ordered in the order of Obliqueness (KC77a; Pullum77a) and since this order corresponds to the order in which the complements are serialized in English, the example in (13) can be analyzed as in Figure 3.⁴ The second tree in the figure is the German counterpart of *gave Sandy a book*: the finite verb in final position with its two objects in normal order. Section 4 explains why SOV languages like German and Japanese contain their subject in the COMPS list while SVO languages like English and Romance languages do not.

The alternative to using relational constraints as in Schema 3 is to use sets rather than lists for the representation of valence information (Gunji86a; HN89a; Pollard90a-Eng; Oliva92b; EEU92a). The Head-Complement Schema would combine the head with one of its complements. Since the elements of a set are not ordered, any complement can be taken and hence all permutations of complements are accounted for.

⁴This structure may seem strange to those working in Mainstream Generative Grammar (MGG, GB/Minimalism). In MGG, different branchings are assumed, since the form of the tree plays a role in Binding Theory. This is not the case in HPSG: Binding is done on the ARG-ST list. See [chapters/binding](#), Chapter ?? of this volume for a discussion of HPSG’s Binding Theory and [chapters/minimalism](#), Chapter ?? of this volume for a comparison between HPSG and Minimalism.



Figure 3: Analysis of the English VP *gave Sandy a book* and the corresponding German verbal projection *Sandy ein Buch gab* with binary branching structures

The disadvantage of set-based approaches is that sets do not impose an order on their members, but an order is needed for various subtheories of HPSG (see **chapters/case**, Chapter ?? of this volume on case assignment, and **chapters/binding**, Chapter ?? of this volume on Binding Theory). In the approach proposed above and in Müller (Mueller2005c; MuellerHPSGHandbook; MuellerCoreGram), the valence lists are ordered but the schema allows for combination with any element of the list. For valence representation and the order of elements in valence lists see **chapters/arg-st**, Chapter ?? of this volume.

4 SVO vs. SOV

The careful reader will have noticed that the **COMPS** list of *gave* in Figure 3 contains the two objects, while its German counterpart *gab* has three elements in the **COMPS** list. The rationale behind this is explained in this section.

In principle, one could assume a rule like (6) for SVO languages like English as well. The SVO order would then be accounted for by linearization rules stating that $NP[nom]$ precedes the finite verb while other arguments follow it. This would get the facts about simple sentences like (16a) right but leaves the analysis of (16c) open.

- (16) a. Peter reads books.
b. Peter often read books.

The generalization about languages like English is that adverbials can attach to the left or to the right of verbs with their objects, that is, to the left or to the right

of the VP. Researchers like **Borsley87a** argued that subjects, specifiers, and complements differ in crucial ways and should be represented by special (valence) features. For example, the subject of the VP *to read more books* is not realized but is referred to in Control Theory (**chapters/control-raising**, Chapter ?? of this volume).

(17) Peter tries to read more books.

The subject in English main clauses is similar to the determiner in nominal structures, so one way of expressing this similarity is by using the same valence features and the same schema for subject-VP combinations as for determiner-noun combinations.⁵ The schema is given here as Schema 4:

Schema 4 (Specifier-Head Schema)

specifier-headphrase \Rightarrow

$$\left[\begin{array}{l} \text{SYNSEM|LOC|CAT|SPR } \boxed{1} \\ \text{HEAD-DTR|SYNSEM|LOC|CAT } \left[\begin{array}{l} \text{SPR } \boxed{1} \oplus \langle \boxed{2} \rangle \\ \text{COMPS } \langle \rangle \end{array} \right] \\ \text{NON-HEAD-DTRS } \langle [\text{SYNSEM } \boxed{2}] \rangle \end{array} \right]$$

The last element of the SPR list is realized as the non-head daughter. The remaining list is passed up to the mother node. Note that the non-head daughter is taken from the end of the SPR list. For heads that have exactly one specifier this difference is irrelevant, but in the analysis of object shift in Danish suggested by **MOe2013b**, the authors assume multiple specifiers and hence the difference in order of combination is relevant. The head-daughter must have an empty COMPS list. This way it is ensured that verbs form a unit with their objects (the VP) and the subject is combined with the VP, rather than the subject combining with a lexical verb and this combination combining with objects later.

The analysis of the sentence in (18) including the analysis of the NP is given in Figure 4.

(18) Kim gave Sandy a book.

For German, it is standardly assumed that the subjects of finite verbs are treated like complements (**Pollard90a-Eng**, **Kiss95a**) and hence are represented at the COMPS list (as in Figure 3). The assumption that arguments of finite verbs are

⁵This is non-standard in HPSG. Usually the **SUBJECT** feature is used for subjects and **SPR** for determiners. I follow the German HPSG tradition and use **SUBJ** for unexpressed subjects. See also **chapters/np**, Chapter ?? of this volume for alternative analyses of nominal structures that do not assume a selection of the determiner by the noun.

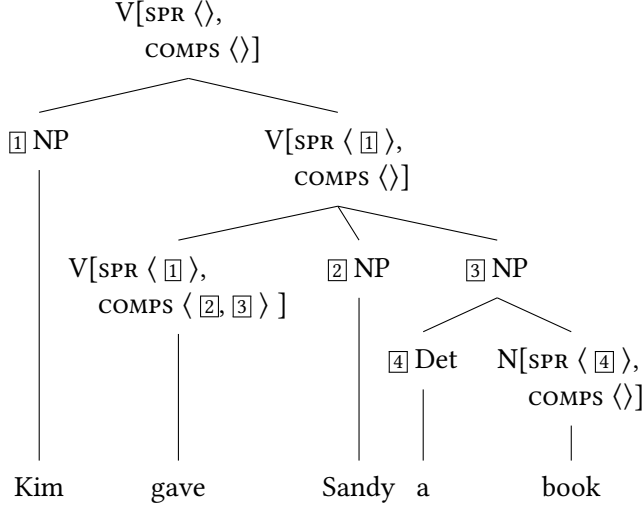


Figure 4: Analysis of *Kim gave Sandy a book* with SPR and COMPS feature and a flat VP structure

complements is also made by researchers working in different research traditions (Eisenberg94b). By assuming that the subject is listed among the complements of a verb it is explained why it can be placed in any position before, between, and after them. So in summary, German differs from English in the way the arguments are distributed on the valence lists, on the one hand in order to capture the similarity in English between combinations of subjects with VPs and determiners with nouns, and on the other hand to allow German the flexible constituent order it needs. However, HPSG has a more basic representation in which the languages do behave the same: the argument structure represented at the ARG-ST list. The ARG-ST list contains *synsem* objects and is used for linking (**chapters/arg-st**, Chapter ?? of this volume), case assignment (**chapters/case**, Chapter ?? of this volume), and binding (**chapters/binding**, Chapter ?? of this volume). Ditransitive verbs in German and English have three NP arguments on their ARG-ST and they are linked in the same way to the semantic representation (MuellerLFGphrasal). (19) shows the mapping from ARG-ST to SPR and COMPS:

- (19) *gives* (English, SVO language): $\left[\begin{array}{l} \text{SPR} \quad \langle [1] \rangle \\ \text{COMPS} \quad [2] \\ \text{ARG-ST} \quad \langle [1] \text{ NP} \rangle \oplus [2] \langle \text{NP}, \text{NP} \rangle \end{array} \right]$ *gibt* (German, SOV language): $\left[\begin{array}{l} \text{SPR} \quad \langle \rangle \\ \text{COMPS} \quad [1] \\ \text{ARG-ST} \quad [1] \langle \text{NP}, \text{NP}, \text{NP} \rangle \end{array} \right]$

In SVO languages, the first element of the ARG-ST list is mapped to SPR and all others to COMPS and in languages without designated subject positions all ARG-ST elements are mapped to COMPS.

Having explained scrambling in HPSG and the order of subjects in SVO languages, I now turn to “head movement”.

5 Head movement vs. constructional approaches assuming flat structures

The Germanic languages signal the clause type by verb position. All Germanic languages with the exception of English are V2 languages: the finite verb is in second position in declarative main clauses. The first position can be filled by any other constituent, for example a subject, objects, or adverbials. (20) shows an example from the V2 language German and its English translation.

- (20) Eigentlich mag ich Katzen sehr. (German)
 actually like I cats really
 ‘I actually really like cats.’

The fronted material is not necessarily from the matrix clause, clause boundary crossing non-local dependencies are possible. The same holds for questions with *w* phrases.

Yes/no questions are formed by putting the verb in initial position:

- (21) Magst du Katzen? (German)
 like you cats
 ‘Do you like cats?’

English is a so-called *residual V2 language*, that is, there are some constructions that are parallel to what is known from V2 languages. For example, while declarative clauses are in base order (SVO), questions follow the pattern that is known from other Germanic languages with the finite verb in second position.⁶

- (22) What_{*i*} will Kim read _ _{*i*}?

⁶SVO is not V2 although the verb is in second position in SVO sentences. Languages can be categorized into SOV, SVO, VSO, OSV, OVS, and VOS languages and into V2 or non-V2 languages. These two dimensions are independent. For example, Danish is an SVO language that is V2, while German is SOV and V2 (Haider2020a). See MuellerGermanic for discussion and the analysis of this variation in HPSG.

Analyses assuming flat structures (or flat linearization domains, see Section 6) usually treat alternative orders of verbs in Germanic languages as linearization variants (Reape94a; Kathol2001a; Mueller95c; Mueller2003a; TBjerre2006a), but this is not necessarily so, as Bouma and van Noord's analysis of Dutch clauses show (BvN98a). The alternative to verb placement as linearization is something that is similar to verb movement in Government & Binding: an empty element takes the position of the verb in its canonical position and the verb is realized in initial or – if something is realized before the finite verb – in second position. The following subsection deals with such approaches in more detail. Subsection 5.2 deals with a constructional approach.

5.1 Head movement approaches

Building on work by Jacobson87 in the framework of Categorical Grammar, Borsley89 showed that in addition to the analysis of auxiliary inversion in English that was suggested in GPSG (GKPS85a), an analysis that is similar to the movement-based analysis in GB is possible in HPSG as well. Head movement analyses in GPSG and HPSG are concerned with the verb placement in pairs such as the one in (23) rather than with adverb placement as in GB analyses by Pollock89a-u and Cinque99a-u.

- (23) a. Will Kim get the job?
 b. Kim will get the job.

The technique that is used in Borsley's analysis is basically the same that was developed by Gazdar81a for the treatment of nonlocal dependencies in GPSG. An empty category is assumed and the information about the missing element is passed up the tree until it is bound off at an appropriate place (that is, by the fronted verb). Note that the heading of this section contains the term *head movement* and I talk about traces, but it is not the case that something is actually moved. There is no underlying structure with a verb that is transformed into one with the verb fronted and a remaining trace in the original position of the verb. Instead, the empty element is a normal element in the lexicon and can function as the verb in the respective position. The analysis of (23a) is shown in Figure 5. A special variant of the auxiliary is licensed by a unary rule. The unary rule has as a daughter the auxiliary as it appears in canonical SVO order as in (23b). It licenses an auxiliary selecting a full clause in which the daughter auxiliary (with the LOCAL value [2]) is missing. The fact that the auxiliary is missing is represented as the value of DOUBLE SLASH (DSL). The value of DSL is a *local* object, that is, something that contains syntactic and semantic information ([2] in Figure 5). DSL

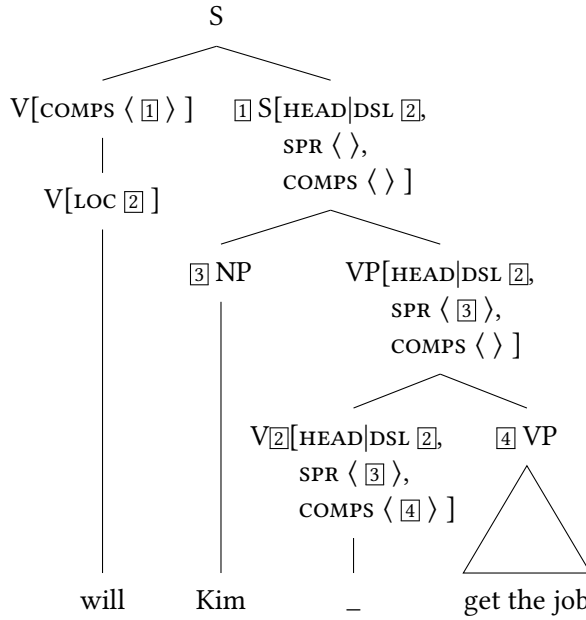


Figure 5: Analysis of English auxiliary constructions as head-movement following Borsley89

is a head feature and hence available everywhere along a projection path (see **chapters/properties**, Chapter ?? of this volume for the Head Feature Principle). The empty element for head movement is rather simple:

(24) empty element for head movement:

$$\left[\begin{array}{l} \text{word} \\ \text{PHON} \quad \langle \rangle \\ \text{SYNSEM|LOC } 1 \quad [\text{CAT|HEAD|DSL } 1] \end{array} \right]$$

It states that there is an empty element that has the local requirements that correspond to its DSL value. For cases of verb movement it says: I am a verb that is missing itself.

Such head-movement analyses are assumed by most researchers working on German (KW91a; Oliva92a; Netter92; Frank94; Kiss95a; Feldhaus97, Meurers2000b; Mueller2005c; MuellerGS) and also by BvN98a in their work on Dutch, by MOeDanish in their grammar of Danish and by MuellerGermanic for Germanic in general.

check reference to Chapter ??.

5.2 Constructional approaches

The alternative to head-movement-based approaches is a flat analysis with an alternative serialization of the verb. This was already discussed with respect to German, but I want to discuss English auxiliary constructions here, since they have figured prominently in linguistic discussions.⁷ In the analysis of (25) shown in Figure 6, the auxiliary *did* selects for the subject *Kim* and a VP *get the job*.

(25) Did Kim get the job?

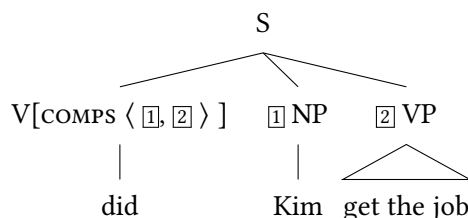


Figure 6: Analysis of English auxiliary constructions based on Sag2020a

The tree in Figure 6 is licensed by a schema combining a head with its subject ([1]) and its VP complement ([2]) in one go.⁸ As has been common in HPSG since the mid-1990s (Sag97a), phrasal schemata are organized in type hierarchies and the general schema for auxiliary-initial constructions has the type *aux-initial-cxt*. Fillmore99a and Sag2020a argue that there are various usages of auxiliary-initial constructions and assign the respective usages to subconstructions of the general auxiliary-initial construction. Technically this amounts to stating subtypes of *aux-initial-cxt*. For example, Sag2020a posit a subtype *polar-int-cl* for polar interrogatives like (26a) and another subtype *auxinitial-excl-cl* for exclamatives like (26b).

- (26) a. Are they crazy?
b. Are they crazy!

Chomsky2010a compared the various clause types used in HPSG with the – according to him – much simpler Merge-based analysis in Minimalism. Minimalism assumes just one very general schema for combination (External Merge is

⁷For a discussion including French verb placement see AG97a-u and KS2002a.

⁸An alternative is to assume a separate valence feature for the subject (SUBJ) and a schema that combines the head with the element in the SUBJ list and the elements in the COMPS list (GSag2000a-u).

basically equivalent to our Schema 3 above, see **MuellerUnifying**), so this rule for combining linguistic objects is very simple, but this does not help in any way when considering the facts: there are at least five different meanings associated with auxiliary initial clauses (polar interrogative, blessings/curses, negative imperative, exclamatives, conditionals) and these have to be captured somewhere in a grammar. One way is to state them in a type hierarchy as is done in some HPSG analyses and in Sign-Based Construction Grammar (SBCG), another way is to use implicational constraints that assign meaning with respect to actual configurations (see Section 5.3), and a third way is to do everything lexically. The only option for Minimalism is the lexical one. This means that Minimalism has to either assume as many lexical items for auxiliaries as there are types in HPSG or to assume empty heads that contribute the meaning that is contributed by the phrasal schemata in HPSG (Borsley2006a; **chapters/minimalism**, Chapter ?? of this volume). The latter proposal is generally assumed in Cartographic approaches (Rizzi97a-u). Since there is a fixed configuration of functional projections that contribute semantics, one could term these Rizzi-style analyses *Crypto-Constructional*.

Having discussed a lexical approach involving an empty element and a phrasal approach that can account for the various meanings of auxiliary inversion constructions, I turn now to a mixed approach in the next section and show how the various meanings associated with certain patterns can be integrated into accounts with rather abstract schemata for combinations like the one described in Section 5.1.

5.3 Mixed approaches

The situation with respect to clause types is similar in German. Verb first sentences can be yes/no questions (27a), imperatives (27b), conditional clauses (27c), and declarative sentences with topic drop (27d).

- (27) a. Kommt Peter? (German)
 comes Peter
 ‘Is Peter coming?’
 b. Komm!
 come
 c. Kommt Peter, komme ich nicht.
 comes Peter come I not
 ‘If Peter comes, I won’t come.’

- d. Kommt. (Was ist mit Peter?)
 comes what is with Peter
 ‘What about Peter?’ ‘He is coming.’

Verb second sentences can be questions (28a), declarative sentences (28b), or imperatives (28c).

- (28) a. Wer kommt? (German)
 who comes
 b. Peter kommt.
 Peter comes
 c. Jetzt komm!
 now come
 ‘Come now!’

While one could try and capture this situation by assuming surface order-related clause types, such approaches are rarely used in HPSG (but see **Kathol2001a** and **Wetta2011a**, and see Section 6.4.2 on why such approaches are doomed to failure). Rather, researchers assumed binary branching head-complement structures together with verb movement (for references see the end of Section 5.1).⁹

As was explained in Section 5.1, the head movement approaches are based on lexical rules or unary projections. These license new linguistic objects that could contribute the respective semantics. In analogy to what **Borsley2006a** has discussed with respect to extraction structures, this would mean that one needs seven versions of fronted verbs to handle the seven cases in (27) and (28), which would correspond to the seven phrasal types that would have to be stipulated in phrasal approaches. But there is a way out of this: one can assume one lexical item with underspecified semantics. HPSG makes it possible to use implicational constraints referring to a structure in which an item occurs. Depending on the context, the semantics contributed by a specific item can be further specified. Figure 7 shows the construction-based and the lexical-rule-based analysis in the abstract for comparison. In the construction-based analysis, the daughters contribute x and y as semantic values and the whole construction adds the construction meaning f . In the lexical-rule- or unary-projection-based analysis, the lexical rule/unary projection adds the f and the output of the rule is combined with the other daughter without any contribution by a specialized phrasal

⁹I assumed linearization domains (see Section 6) for ten years and then switched to the head-movement approach (**Mueller2005c**; **Mueller2005d**; **MuellerGS**). For a detailed discussion of all alternative proposals and a fully worked out analysis see **MuellerGS**.

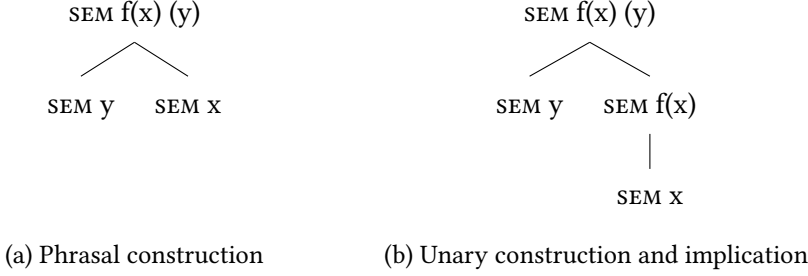


Figure 7: Construction-based, phrasal approach and approach with implicational constraint

construction. Now, implicational constraints can be used to determine the exact contribution of the lexical item (**MuellerSatztypen**). This is shown with the example of a question in Figure 8. The implication says: when the configuration has

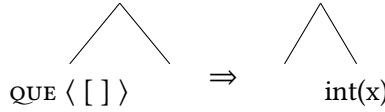


Figure 8: Implication for interrogative sentences

the form that there is a question pronoun in the left daughter, the output of the lexical rule gets question semantics. Since HPSG represents all linguistic information in the same attribute value matrix (AVM), such implicational constraints can refer to intonation as well and hence, implications for establishing the right semantics for V1 questions (27a) vs. V1 conditionals (27c) can be formulated.

6 Constituent order domains and linearization

There is an interesting extension to standard HPSG that opens up possibilities for analyses that are quite different from what is usually done in theoretical linguistics: Mike Reape (**Reape91; Reape92a; Reape94a**) working on German suggested formal tools that allow for the modeling of discontinuous constituents.¹⁰ His original motivation was to account for scrambling of arguments of verbs forming

¹⁰See also **Wells47a**, **Dowty90a-Eng**, and **Blevins94a-u** for proposals assuming discontinuous constituents in other frameworks.

verbal complexes, but this analysis was superseded by Hinrichs and Nakazawa’s analysis (HN89a; HN94a) since purely linearization-based approaches are unable to account for agreement and the so-called remote passive (Kathol98b; Mueller99a). Nevertheless, Reape’s work was taken up by others and was used for analyzing German (KP95a; Kathol2000a; Mueller95c; Babel; Mueller2004b; Wetta2011a; Wetta2014a-u). As will be discussed below in Section 6.4, there were reasons for abandoning linearization-based analyses of German assuming discontinuous constituents (Mueller2005dMuellerGS) but constituent order domains still play a major role in analyzing ellipsis (chapters/ellipsis, Chapter ?? of this volume) and coordination (chapters/coordination, Chapter ?? of this volume). BGM99a show that complex predicate formation does not account for subject-verb inversion in French and suggest a domain-based approach. BG2007b-u, also working on French, treat sentential adverbs in a domain-based approach.

6.1 A special representational layer for constituent order

The technique that is used to model discontinuous constituents in frameworks like HPSG goes back to Mike Reape’s work on German (Reape91; Reape92a; Reape94a). Reape uses a list called DOMAIN to represent the daughters of a sign in the order in which they appear at the surface of an utterance. (29) shows an example in which the DOM value of a headed-phrase is computed from the DOM value of the head and the list of non-head daughters.

$$(29) \text{ headed-phrase} \Rightarrow \left[\begin{array}{l} \text{HEAD-DTR} | \text{DOM} \quad \boxed{1} \\ \text{NON-HEAD-DTRS} \quad \boxed{2} \\ \text{DOM} \quad \boxed{1} \circ \boxed{2} \end{array} \right]$$

The symbol ‘ \circ ’ stands for the *shuffle* relation. *shuffle* relates three lists A, B and C iff C contains all elements from A and B and the order of the elements in A and the order of the elements of B is preserved in C. (30) shows the combination of two sets with two elements each:

$$(30) \quad \langle a, b \rangle \circ \langle c, d \rangle = \langle a, b, c, d \rangle \vee \\ \langle a, c, b, d \rangle \vee \\ \langle a, c, d, b \rangle \vee \\ \langle c, a, b, d \rangle \vee \\ \langle c, a, d, b \rangle \vee \\ \langle c, d, a, b \rangle$$

The result is a disjunction of six lists. *a* is ordered before *b* and *c* before *d* in all of these lists, since this is also the case in the two lists $\langle a, b \rangle$ and $\langle c, d \rangle$ that have been combined. But apart from this, *b* can be placed before, between, or after *c* and *d*.

On the linearization-based approach, every word comes with a domain value that is a list that contains the word itself:

(31) Domain contribution of single words, here *gibt* ‘gives’:

$$\boxed{1} \begin{bmatrix} \text{PHON} & \langle \textit{gibt} \rangle \\ \text{SYNSEM} & \dots \\ \text{DOM} & \langle \boxed{1} \rangle \end{bmatrix}$$

The description in (31) may seem strange at first glance, since it is cyclic, but it can be understood as a statement saying that *gibt* contributes itself to the items that occur in linearization domains.

The constraint in (32) is responsible for the determination of the PHON values of phrases:

$$(32) \textit{phrase} \Rightarrow \begin{bmatrix} \text{PHON } \boxed{1} \oplus \dots \oplus \boxed{n} \\ \text{DOM } \left\langle \begin{bmatrix} \textit{sign} \\ \text{PHON } \boxed{1} \end{bmatrix}, \dots, \begin{bmatrix} \textit{sign} \\ \text{PHON } \boxed{n} \end{bmatrix} \right\rangle \end{bmatrix}$$

It states that the PHON value of a sign is the concatenation of the PHON values of its DOMAIN elements. Since the order of the DOMAIN elements corresponds to their surface order, this is the obvious way to determine the PHON value of the whole linguistic object.

Figure 9 shows how this machinery can be used to license binary branching structures with discontinuous constituents in the sentence *dass dem Kind ein Mann das Buch gibt* ‘that a man gives the child the book’. Words or word sequences that are separated by commas stand for separate domain objects, that is, $\langle \textit{das}, \textit{Buch} \rangle$ contains the two objects *das* and *Buch* and $\langle \textit{das Buch}, \textit{gibt} \rangle$ contains the two objects *das Buch* and *gibt*. The important point to note here is that the arguments in the tree are combined with the head in the order accusative, dative, nominative, although the elements in the constituent order domain (i.e. in the list of DOMAIN elements and in the surface sentence) are realized in the order dative, nominative, accusative, rather than nominative, dative, accusative, which is what one might expect based on the order in which they are combined in the tree. This is possible since the formulation of the computation of the DOM value using the shuffle operator allows for discontinuous constituents. The node for *dem Kind das Buch gibt* ‘the child the book gives’ is discontinuous: *ein Mann*

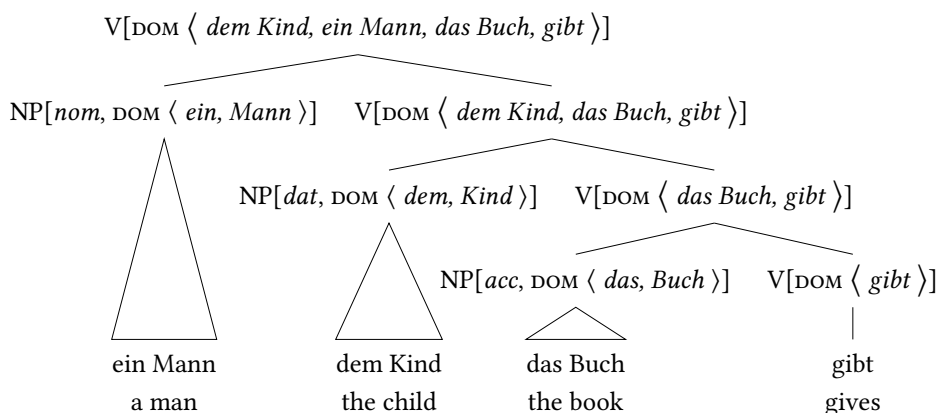


Figure 9: Analysis of *dass dem Kind ein Mann das Buch gibt* ‘that a man gives the child the book’ with binary branching structures and discontinuous constituents. The tree shows the order of combination, which does not correspond to the linearization of the DOMAIN objects.

‘a man’ is inserted into the domain between *dem Kind* ‘the child’ and *das Buch* ‘the book’. This is more obvious in Figure 10, which has a serialization of NPs that corresponds to their order.

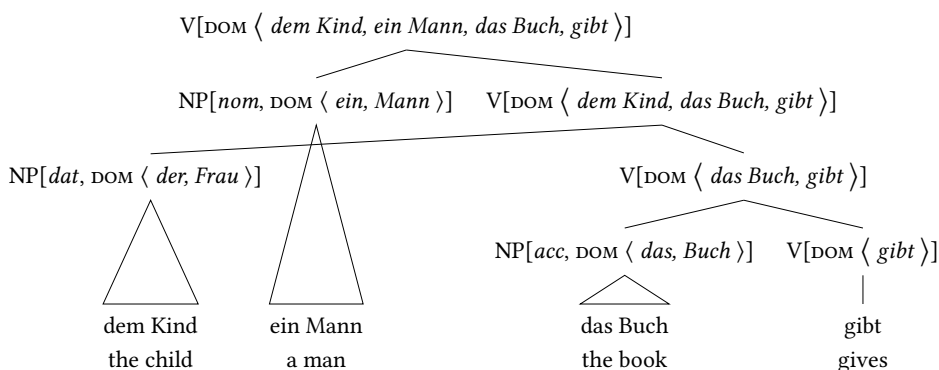


Figure 10: Analysis of *dass dem Kind ein Mann das Buch gibt* ‘that a man gives the child the book’ with binary branching structures and discontinuous constituents, more clearly showing the discontinuity

6.2 Absolutely free

While German is more striking than English in terms of constituent order, languages like Warlpiri are even more so, since they have much freer constituent order. In Warlpiri the auxiliary has to be in first or in second position (Laughren89a-uSimpson91a-u but apart from this, even parts of what are noun phrases in German and English can appear separated from each other. For example, the two parts of the NP *Kurdujarrarlu witajarrarlu* ‘child small’ may appear discontinuously since they are marked with the same case (Simpson91a-u):

- (33) Kurdu-jarra-rlu ka-pala maliki wajili.pi-nyi wita-jarra-rlu.
 child-DU-ERG PRS-3DU.SBJ dog.ABS chase-NPST small-DU-ERG
 ‘Two small children are chasing the dog.’ or
 ‘Two children are chasing the dog and they are small.’

DS99a develop an analysis for this that simply liberates domain elements and inserts them into the next higher domain. (34) shows how this is formalized:

- (34) *liberating-phrase* \Rightarrow
- $$\left[\begin{array}{l} \text{DOM} \quad \delta_0 \circ \delta_1 \circ \dots \circ \delta_n \\ \text{HEAD-DTR} \quad \left[\text{DOM } \delta_0 \right] \\ \text{NON-HEAD-DTRS} \quad \left\langle \left[\text{DOM } \delta_1 \right], \dots, \left[\text{DOM } \delta_n \right] \right\rangle \end{array} \right]$$

Rather than inserting the complete daughters into the domain of the mother as in (29), the DOM values of the daughters are shuffled into the domain of the mothers. So instead of having the NPs in the same domain as the verb as in the German example in the previous section, one has all the parts of NPs in the next higher domain. Hence, a single nominal element being placed in front of the auxiliary in second position is explained without difficulty. Figure 11 shows DS99a’s (DS99a) analysis of a version of (33) with the VP constituents *maliki wajilipinyi* ‘dog chase’ serialized after *witajarrarlu* ‘small’. Here *Kurdujarrarlu* ‘child’ and *witajarrarlu* ‘small’ form an NP. They contribute two independent domain objects ([1] and [3]) to the domain of the mother. The second element in this domain has to be the auxiliary ([2]), [1] is realized initially and [3] follows the auxiliary.

We have seen so far an analysis that inserts complete objects into the domain of the mother (the analysis of German) and an analysis that inserts all domain objects of objects into the domain of the mother (the analysis of Warlpiri), and in the next subsection I want to look at an intermediate case, so-called *partial compaction*.

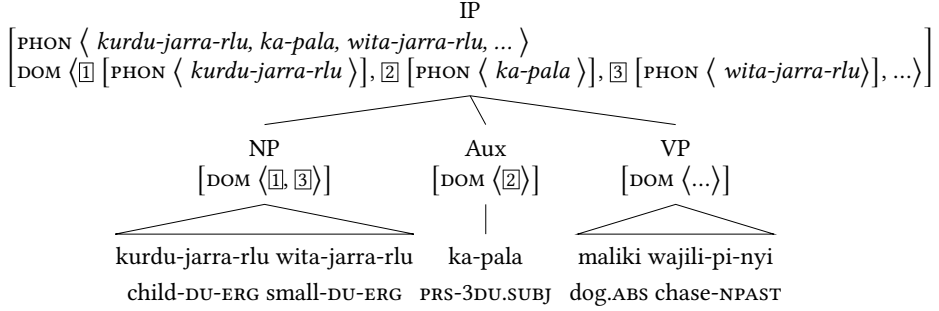


Figure 11: Analysis of free constituent order in Warlpiri according to DS99a

6.3 Partial compaction (extraposition)

KP95a develop an analysis of extraposition that is a mix of the strategies discussed in the two previous subsections: most of one NP object is inserted into the domain of the mother as a single object, while only those parts that are extraposed are liberated and inserted as individual domain objects into the domain of the mother.¹¹ KP95a's analysis of (35) is given in Figure 12.¹²

- (35) einen Hund füttern, der Hunger hat (German)
a dog feed that hunger has
‘feed a dog that is hungry’

einen Hund, der Hunger hat ‘a dog who is hungry’ consists of three domain objects: *einen* ‘a’, *Hund* ‘dog’, and *der Hunger hat* ‘who is hungry’. The two initial ones are inserted as one object (the NP *ein Hund* ‘a dog’) into the higher domain and the relative clause is liberated. While the formation of the new domain at the mother node is relatively straightforward in the cases discussed so far, a complex relational constraint is needed to split the relative clause ([3]) from the other domain objects and construct a new domain object that has the determiner and the

¹¹This analysis of extraposition is not the only option available in HPSG. I explain it here since it shows the flexibility of the domain approach. The more common analysis of extraposition is one that is parallel to the SLASH-based approach to extraction that is explained in chapters/udc, Chapter ?? of this volume. Since constraints regarding locality differ for fronting to the left and extraposition to the right, a different feature is used (EXTRA). See Keller95b and Mueller99a for discussion. More recent approaches assume the projection of semantic indices (Kiss2005a) to be able to solve puzzles like Link84a-u’s (Link84a-u) hydra sentences and even more recent proposals mix index projection and EXTRA projection (Crysmann2013a).

¹²The figure is taken over from KP95a. Words in italics are the object language. Part of speech or category labels are provided at the top of AVMS.

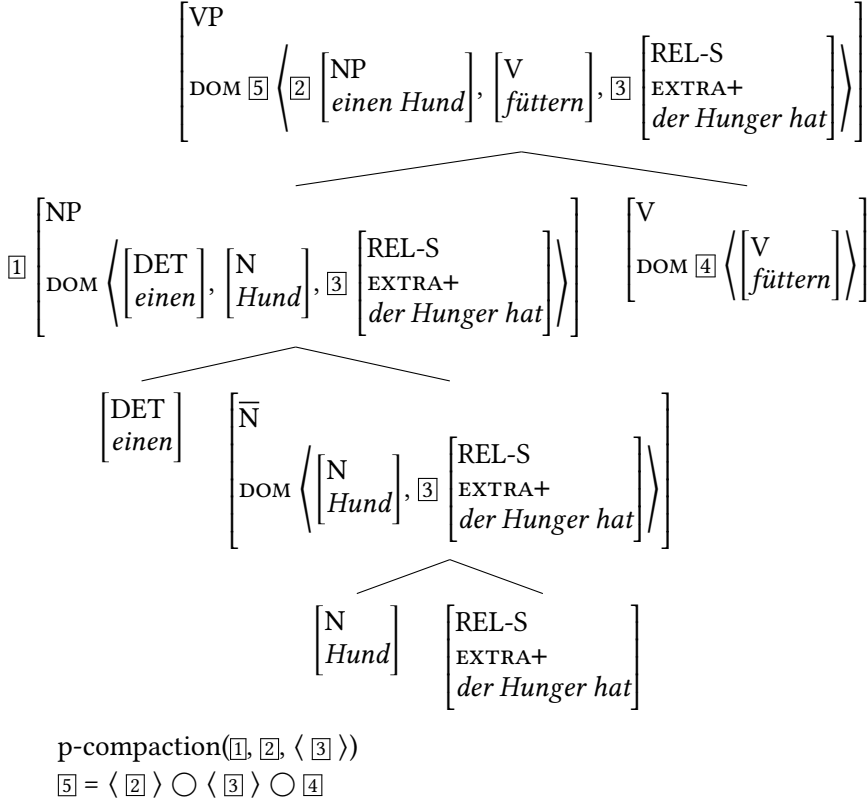


Figure 12: Analysis of extraposition via partial compaction of domain objects according to **KP95a**

noun as constituents ($[2]$). **KP95a** have a relational constraint called *compaction* that builds new domain objects for insertion into higher domains. *partial compaction* takes an initial part of a domain and forms a new domain object from this, returning the remaining domain objects for separate insertion into the higher domain. Due to space limitations, this constraint will not be discussed here, but see **Mueller99a** for a refined version of **KP95a**'s constraint. The effect of partial compaction in Figure 12 is that there is a new object $[2]$ and a list containing the remaining objects, in the example $\langle [3] \rangle$. A list containing the new object $\langle [2] \rangle$ and the list containing the remaining objects $\langle [3] \rangle$ are shuffled with the domain list of the head $[4]$. Since the relative clause is now in the same domain as the verb, it can be serialized to the right of the verb.

So, this subsection showed how examples like (35) can be analyzed by allowing

for a discontinuous constituent consisting of an NP and a relative clause. Rather than liberating all daughters and inserting them into the domain of the mother node as in the Warlpiri example, determiner and noun form a new object, an NP, and the newly created NP and the relative clause are inserted into the domain of the mother node. This explains why determiner and noun have to stay together while the relative clause may be serialized further to the right.

6.4 Problems with order domains

Constituent order domains may seem rather straightforward since linearization facts can be handled easily. I assumed constituent order domains and discontinuous constituents for German myself for over a decade (Mueller95c; Mueller2004b). However, there are some problems that seem to suggest that a traditional GB-like head-movement approach is the better alternative. In what follows I want to discuss just two problematic aspects of linearization approaches: spurious ambiguities and apparently multiple frontings.

6.4.1 Partial fronting and spurious ambiguities

Kathol2000a suggests an analysis with binary branching structures in which all arguments are inserted into a linearization domain and can be serialized there in any order, provided no LP rule is violated. Normally one would have the elements of the COMPS list in a fixed order, combine the head with one element from the COMPS list after another, and let the freedom in the DOM list be responsible for the various attested orders. So, both sentences in (36) would have analyses in which the verb *erzählt* ‘tells’ is combined with *Geschichten* ‘stories’ first and then *Geschichten erzählt* ‘stories tells’ is combined with *den Wählern* ‘the voters’. Since the verb and all its arguments are in the same linearization domain they can be ordered in any way, including the two possibilities in (36):

- (36) a. weil er den Wählern Geschichten erzählt (German)
 because he the voters stories tells
 ‘because he tells the voters stories’
 b. weil er Geschichten den Wählern erzählt
 because he stories the voters tells

The problem with this approach is that examples like (37) show that grammars have to account for fronted combinations of any of the verb and any of its objects to the exclusion of the other:

- (37) a. Geschichten erzählen sollte man den Wählern nicht. (German)
 stories tell should one the voters not
 ‘One should not tell the voters such stories.’
 b. Den Wählern erzählen sollte man diese Geschichten nicht.
 the voters tell should one these stories not

Kathol2000a accounts for examples like (37) by relaxing the order of the objects in the valence list. He uses the shuffle operator in the valence representation:

$$(38) \quad \langle \text{NP}[\textit{nom}] \rangle \oplus (\langle \text{NP}[\textit{dat}] \rangle \circ \langle \text{NP}[\textit{acc}] \rangle)$$

This solves the problem with examples like (37) but it introduces a new one: sentences like (36) now have two analyses each. One is the analysis we had before and another one is the one in which *den Wählern* ‘the voters’ is combined with *erzählt* ‘tells’ first and the result is then combined with *Geschichten* ‘stories’. Since both objects are inserted into the same linearization domain, both orders can be derived. So we have too much freedom: freedom in linearization and freedom in the order of combination. The proposal that I suggested in Müller (**Mueller2005c**; **MuellerGS**) and which is implemented in Schema 3 above has just the freedom in the order of combination and hence can account for both (36) and (37) without spurious ambiguities.

6.4.2 Surface order, clause types, fields within fields, and empty elements

Kathol2001a develops an analysis of German that uses constituent order domains and determines the clause types on the basis of the order of elements in such domains. He suggests the topological fields 1, 2, 3, and 4, which correspond to the traditional topological fields *Vorfeld* ‘prefield’, *linke Satzklammer* ‘left sentence bracket’, *Mittelfeld* ‘middle field’, *rechte Satzklammer* ‘right sentence bracket’. Domain objects may be assigned to these fields, and they are then ordered by linearization constraints stating that objects assigned to 1 have to precede objects of type 2, type 3, and type 4. Objects of type 2 have to precede type 3, and type 4 and so on. For the *Vorfeld* and the left sentence bracket, he stipulates uniqueness constraints saying that at most one constituent may be of this type. This can be stated in a nice way by using the linearization constraints in (39):

- (39) a. $1 < 1$
 b. $2 < 2$

This trick was first suggested by **GKPS85a** in the framework of GPSG and it works because, if there were two objects of type 1, then each one would be re-

quired to precede the other one, resulting in a violation of the linearization constraint. So in order to avoid such constraint violation there must not be more than one 1.

Kathol2001a assumes the following definition for V2 clauses:

$$(40) \quad V2\text{-clause} \Rightarrow \left[\begin{array}{c} S[fin] \\ \text{DOM} \left\langle [1], \left[\begin{array}{c} 2 \\ V[fin] \end{array} \right], \dots \right\rangle \end{array} \right]$$

This says that the constituent order domain starts with one element assigned to field 1, followed by another domain object assigned to field 2. While this is in accordance with general wisdom about German, which is a V2 language, there are problems for entirely surface-based theories: German allows for multiple constituents in front of the finite verb. (41) shows some examples:

- (41) a. [Zum zweiten Mal] [die Weltmeisterschaft] errang Clark 1965 ...¹³
 to.the second time the world.championship won Clark 1965
 ‘Clark won the world championship for the second time in 1965.’
 b. [Dem Saft] [eine kräftige Farbe] geben Blutorangen.¹⁴
 the.DAT juice a.ACC strong color give blood.oranges
 ‘Blood oranges give the juice a strong color.’

Mueller2003b extensively documents this phenomenon. The categories that can appear before the finite verb are almost unrestricted. Even subjects can be fronted together with other material (**BC2010a**; **Bildhauer2011a**). The empirical side of these apparent multiple frontings was further examined in the Collective Research Center 632, Project A6, and the claim that only constituents depending on the same verb can be fronted together (**Fanselow93aHoberg97a**) was confirmed (**MuellerGS**). A further insight is that the linearization properties of the fronted material (NPs, PPs, adverbs, adjectives) correspond to the linearization properties they would have in the *Mittelfeld*. The example in (42) is even more interesting. It shows that there can be a right sentence bracket (the particle *los*) and an extraposed constituent (something following the particle: *damit*) before the finite verb (*geht* ‘goes’):

¹³(Benes71)

¹⁴**BC2010a** found this example in the *Deutsches Referenzkorpus* (DeReKo), hosted at Institut für Deutsche Sprache, Mannheim: <http://www.ids-mannheim.de/kl/projekte/korpora>, 2018-09-13.

- (42) *Los damit geht es schon am 15. April.*¹⁵ (German)
 off there.with goes it PRT on 15. April
 4 5 2 3 3 3
 ‘The whole thing starts on April 15th.’

As far as topology is concerned, this sentence corresponds to sentences with VP fronting and extraposition like the one in (43) discussed in **Reis80a**.

- (43) [*Gewußt, daß du kommst,*] haben wir schon seit langem. (German)
 known that you come have we PART since long
 ‘We have known for a while that you are coming.’

In (43) *gewußt, daß du kommst* ‘known that you come’ forms a VP in which *gewußt* is the right sentence bracket and *daß du kommst* ‘that you come’ is extraposed. We have the same situation in (42) with *los* ‘off’ and *damit* ‘there.with’, except that one would not want to claim that *damit* ‘there.with’ depends on *los* ‘off’.

In Kathol’s system, *los* would be of type 4 and *damit* would have to be of type 5 (an additional type for extraposed items). Without any modification of the general system, we would get a 4 and a 5 ordered before a 2 (a right sentence bracket and a postfield preceding the left sentence bracket), something that is ruled out by Kathol’s linearization constraints.

Mueller2002c, still working in a domain-based framework, developed an analysis assuming an empty verbal head to explain the fact that the fronted constituents have to depend on the same verb and that there is a separate topological area that is independent of the remaining clause. So, *los* and *damit* are domain objects within a larger domain object placed in the prefield. **Wetta2011a** suggests an analysis in which two or more constituents are compacted into one domain object, so *los* and *damit* would form one object that is inserted into the domain containing the finite verb. However, this begs the question of what kind of object it is that is formed. Section 6.3 dealt with partial compaction of NPs. Some of the elements from an NP domain were liberated and other elements were fused into a new object that had the same category as the object containing all material, namely NP. But the situation with examples like (41) and (43) is quite different. We have a particle and a pronominal adverb in (43) and various other combinations of categories in the examples collected by **Mueller2003b** and **Bildhauer2011a**. It would not make sense to claim that the fronted object is a particle or a pronominal adverb. Note that it is not an option to leave the category of

¹⁵taz, 01.03.2002, p. 8.

the fronted object unspecified, since HPSG comes with the assumption that models of linguistic objects are total, that is, maximally specific (King99a-u, see also **chapters/formal-background**, Chapter ?? of this volume). Leaving the category and valence properties of the item in the prefield unspecified would make such sentences infinitely ambiguous. Of course Wetta could state that the newly created object is a verbal projection, but this would just be stating the effect of the empty verbal head within a relational constraint, which I consider less principled than stating the empty element.

However, the empty verbal head that I stated as part of a linearization grammar in 2002 comes as a stipulation, since its only purpose in the grammar of German was to account for apparent multiple frontings. **Mueller2005d**; **MuellerGS** drops the linearization approach and assumes head-movement instead. The empty head that is used for accounting for the verb position in German can also be used to account for apparent multiple frontings. The analysis is sketched in (44):

- (44) a. [_{VP} [Zum zweiten Mal] [die Weltmeisterschaft] _{-V}]_i errang_j
 to.the second time the world.championship won
 Clark 1965 _{-i} _{-j}.
 Clark 1965
- b. [_{VP} Los _{-V} damit]_i geht_j es schon am 15. April _{-i} _{-j}.
 off there.with goes it PRT on 15. April
 ‘The whole thing starts on the 15th April.’

Space precludes going into all the details here, but the analysis treats apparent multiple frontings parallel to partial verb phrase frontings. A lexical rule is used for multiple frontings which is a special case of the head-movement rule that was discussed in Section 5.1. So, apparent multiple frontings are analyzed with means that are available to the grammar anyway. This analysis allows us to keep the insight that German is a V2 language and it also gets the same-clause constraint and the linearization of elements right. As for (44b): *los damit* ‘off there.with’ forms a verbal constituent placed in the *Vorfeld* and within this verbal domain, we have the topological fields that are needed: the right sentence bracket for the verbal particle and the verbal trace and the *Nachfeld* for *damit* ‘there.with’. See **Mueller2005c**; **Mueller2005d**; **MuellerGS** for details.

This paper so far has discussed the tools that have been suggested in HPSG to account for constituent order: flat vs. binary branching structures, linearization domains, head-movement via DSL. I showed that analyses of German relying on discontinuous constituents and constituent order domains are not without problems and that head-movement approaches with binary branching and con-

tinuous constituents can account for the data. I also demonstrated in Section 6.2 that languages like Warlpiri that allow for much freer constituent order than German can be accounted for in models allowing for discontinuous constituents. The following section discusses a proposal by Bender2008a that shows that even languages like the Australian free constituent order languages can be handled without discontinuous constituents.

7 Free constituent order languages without order domains

Bender2008a discusses the Australian language Wambaya and shows how phenomena parallel to those treated by DS99a can be handled without discontinuous constituents. Bender assumes that all arguments of a head are projected to higher nodes even when they are combined with the head; that is, arguments are not canceled off from valence lists. See also Meurers99b; Prze99 and Mueller2008a for earlier non-cancellation approaches.¹⁶ The example (33) from Section 6.2 can be recast with continuous constituents as shown in Figure 13. The figure shows

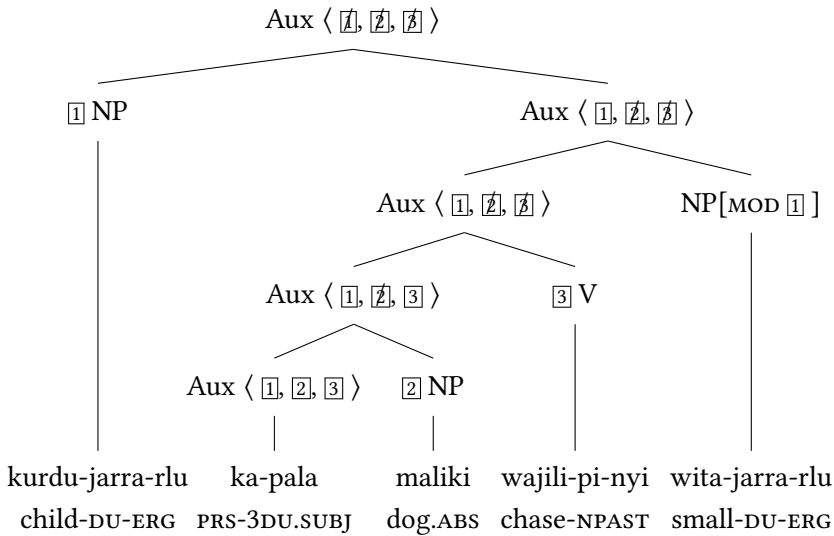


Figure 13: Analysis of free constituent order in Warlpiri using non-cancellation

that arguments are not removed from the valence representation after combina-

¹⁶Higginbotham85a and Winkler97a make similar suggestions with regard to the representation of theta roles.

tion with the head. Rather they are marked as satisfied: [4]. Since they are still in the representation, schemata may refer to them. Bender suggests a schema that identifies the MOD value of an element that could function as an adjunct in a normal head-adjunct structure with an element in the valence representation. In Figure 13, the MOD value of the second ergative nominal *wita-jarra-rlu* ‘small’ is identified with an argument of the auxiliary verb ([1]). The adjunct hence has access to the referential index of the argument and it is therefore guaranteed that both parts of the noun phrase refer to the same discourse referent. The NP for *kurdu-jarra-rlu* is combined with the projection of the auxiliary to yield a complete sentence. Since [1] does not only contain the semantic index and hence information about number (the dual) but also case information, it is ensured that distributed noun phrases have to bear the same case. Since information about all arguments are projected along the head path, [2] would also be available for an adjunct referring to it. So in the place of *wita-jarra-rlu* ‘small-DU-ERG’ we could also have another adjunct referring to *maliki* ‘dog.ABS’. This shows that even languages with constituent order as free as the Australian languages can be handled within HPSG without assuming discontinuous constituents.

8 Summary

A major feature of constraint-based analyses is that when no constraints are stated, there is freedom. The paper discussed the order of head and adjunct: if the order of head and adjunct is not constrained, both orders are admitted.

This paper explored general approaches to constituent order in HPSG. On the one hand, there are approaches to constituent order assuming flat constituent structure, allowing permutation of daughters as long as no LP constraint is violated. On the other hand, there are approaches assuming binary branching structures. Approaches that assume flat structures can serialize the head to the left or to the right or somewhere between other daughters in the structure. Approaches assuming binary branching have to use other means. One possibility is “head movement”, which is analyzed as a series of local dependencies by passing information about the missing head up along the head path. The alternative to head movement is linearization of elements in special linearization domains, allowing for discontinuous constituents. I showed that there are reasons for assuming head-movement for German and how even languages with extremely free constituent order can be analyzed without assuming discontinuous constituents.

Acknowledgements

I thank Anne Abeillé, Bob Borsley, and Doug Ball for very detailed and very useful comments. I thank Elizabeth Pankratz for comments and for proofreading.