THE BASIC PROPERTIES AND ELEMENTS OF HPSG

Anne Abeillé, Université Paris 7
Robert D Borsley, University of Essex

1. Introduction

Head-driven Phrase Structure Grammar (HPSG) dates back to early 1985 when Carl Pollard presented his Lectures on HPSG. It was often seen in the early days as a revised version of the earlier Generalized Phrase Structure Grammar (GPSG) framework (Gazdar *et al*. 1985), but it was also influenced by Categorial Grammar, and, as Pollard and Sag (1987) emphasized, by other frameworks like Lexical-Functional Grammar (LFG), as well. Naturally it has changed in various ways over the decades. This will be discussed in much more detail in the next chapter (Flickinger et al. chap 2), but it makes sense here to distinguish three versions of HPSG. Firstly, there is what might be called early HPSG, the framework presented in Pollard and Sag (1987) and Pollard and Sag (1994).[1] This has most of the properties of more recent versions but only exploits the analytic potential of type hierarchies to a limited degree. Next there is what is sometimes called Construction-based HPSG, the framework adopted in Sag (1997), Ginzburg and Sag (2000), and much other work. Unlike earlier work this utilizes type hierarchies not only in the lexicon but also in the syntax. It is this which makes it construction-based. Finally, in the 2000s, Sag developed a version of HPSG called Sign-Based Construction Grammar (SBCG) (Sag 2012). The fact that this approach has a new name suggests that it is very different from earlier work, but probably most researchers in HPSG would see it as a version of HPSG. Its central feature is the special status it assigns to constructions. In earlier work they are just types of sign, but for HPSG signs and constructions are quite different objects. In spite of this difference, most analyses in Construction-based HPSG could probably be translated into SBCG and vice versa. In this chapter we will concentrate on the ideas of Construction-based HPSG, which is probably the version of the framework that has been most widely assumed. We will comment briefly on SBCG in the final section.

The chapter is organized as follows. In section 2 we set out the properties that characterize the approach, the assumptions it makes about the nature of linguistic analyses and the conduct of linguistic research. Then in section 3 we consider the main elements of HPSG analyses: types, features, and constraints. In section 4 we look more closely at the HPSG approach to the lexicon, and in section 5 we outline the basic of the HPSG approach to syntax. In section 6 we look at some further syntactic structures, and in section 7, we consider some further issues, including SBCG. Finally, in section 8, we summarize the chapter.

2. Properties

Perhaps the first thing to say about HPSG is that it is a form of generative grammar. This means that it seeks to develop precise and explicit analyses of grammatical phenomena. But unlike many versions of generative grammar, it is a declarative or constraint-based approach to grammar. As such, it assumes that a linguistic analysis involves a set of constraints to which linguistic objects must conform, and that a linguistic object is well-formed if and only if it

[1] As discussed in Richer (2018) chapter 3 of this volume, the approaches that are developed in these two books have rather different formal foundations. However, they propose broadly similar syntactic analyses, and for this reason it seems reasonable to group them together as early HPSG.

conforms to all relevant constraints.[2] This includes linguistic objects of all kinds – words, phrases, phonological segments, etc. There are no procedures constructing representations such as the phrase structure and transformational rules of classical transformational grammar or the Merge and Agree operations of Minimalism. Of course, speakers and hearers do construct representations and must have procedures that enable them to do so, but this is a matter of performance, and there is no need to think that the knowledge that is used in performance has a procedural character. Rather, the fact that it used in both production and comprehension (and other activities, e.g. translation) suggests that it should be neutral between the two and hence declarative. For further discussion of the issues, see e.g. Pullum and Scholz (2001), Postal (2003) and Sag and Wasow (2011, 2015).

HPSG is also a monostratal approach, which assumes that linguistic expressions have a single constituent structure. This makes it quite different from transformational grammar, in which an expression can have a number of constituent structures. It means among other things that there is no possibility of saying that an expression occupies one position at one level of structure and another at another. Hence, HPSG has nothing like the movement processes of transformational grammar. The relations that are attributed to movement in transformational work are captured by constraints that require certain features to have the same value. For example, as discussed in section 4, a raising sentence is one with a verb which has the same value for the feature SUBJ(ECT) as its complement and hence combines with whatever kind of subject its complement requires.

HPSG is sometimes described as a concrete approach to syntax. This description refers not only to the fact that it assumes a single constituent structure but also to the fact that this structure is relatively simple, especially compared with the structures that are postulated within Minimalism. Unlike Minimalism, HPSG does not assume that all branching is binary. This inevitably leads to simpler flatter structures. Also unlike Minimalism, it makes limited use of phonologically empty elements. For example, it is not assumed, as in Minimalism, that because some clauses contain a complementizer they all do, an empty one if not an overt one. It is not assumed that because some languages like English (1a) have determiners, they all do, overt or covert. It is also not generally assumed that null subject sentences, such as (1b) from Polish, have a phonologically empty subject in their constituent structure. Thus, the constituent structure of the two following sentences are quite different, even if their semantics are similar:

(1) a. I read a book.
b. Czytałem książkę.
write.PAST.1SG book.ACC
'I read a book.'

It is also assumed in much HPSG work that there are no phonologically empty elements in the constituent structure of an unbounded dependency construction such as the following:

(2) Who did you talk to?

On this view, there is not an empty element but nothing at all following the preposition *to* in (2). There is, however, some debate here (Sag & Fodor 1994, Müller 1996, Borsley & Crysmann, this volume chapter 14).

[2] In classical HPSG all constraints are equal. Hence, there is no possibility as there is in Optimality Theory of violating one if it is the only way to satisfy another more important one. However, see Öpen et al 2004 for an HPSG parser with probabilities or weighted constraints.

A further important feature of HPSG is the rejection of any core-periphery distinction. This means that it is not only concerned with such core phenomena as *wh*-interrogatives, relative clauses, and passives but also with more peripheral phenomena such as the following:

(3) a. It's amazing the people you see here.
b. The more I read, the more I understand.
c. Chris lied his way into the meeting.

These exemplify the nominal extraposition construction (Michaelis & Lambrecht. 1996), the comparative correlative construction (Abeillé et al. 2006, Abeillé & Borsley 2008, Borsley 2011), and the *X's Way* construction (Sag 2012). As we will see, HPSG is an approach which is able to accommodate broad linguistic generalizations and highly idiosyncratic facts and everything in between.

Another notable feature of the framework since the earliest work is a concern with Semantics as well as syntax. More generally, it does not try to reduce either semantics or morphology to syntax. We will comment further on this in the following sections.

We turn now to some assumptions which are more about the conduct of linguistic research than the nature of linguistic analyses. Firstly, HPSG emphasizes the importance of firm empirical foundations in the form of detailed formal analyses of the kind advocated by Chomsky in *Syntactic Structures* (Chomsky 1957). Whereas transformational work typically offers sketches of analyses which might be fleshed out one day, HPSG commonly provides detailed analyses which can be set out in an appendix. A notable example is Ginzburg and Sag (2000), which sets out its analysis of English interrogatives in a 50 page appendix. Arguably, one can only be fully confident that a complex analysis works if it is incorporated into a computer implementation. Hence, computer implementations of HPSG analyses are also quite common (see e.g. Müller 2015, Bender *et al.* 2010, Bender 2016, and Bender and Emerson 2018, chapter 29 of this volume).

Another feature of the framework is a rejection of abstract analyses with tenuous links to the observable data. As we noted above, phonologically empty elements are only assumed if there is compelling evidence for them.[3] Similarly, overt elements are only assumed to have properties for which there is clear evidence. For example, words are only assumed to have case or agreement features if there is some concrete morphological evidence for them, as in Polish. This feature of HPSG stems largely from considerations about acquisition (Müller 2016: chapter 19, Ginzburg this volume, chap 27). Every element or property which is postulated for which there is no clear evidence in the data increases the complexity of the acquisition task and hence necessitates more complex innate machinery. This suggests that such elements and properties should be avoided as much as possible. It has important implications both for the analysis of individual languages and for how differences between languages are viewed.

A related feature of the framework is a rejection of the idea that it is reasonable to assume that a language has some element or property if some other languages do. Many languages have case and many languages have agreement, but for HPSG it does not follow that they all do. As Müller (2015: 25) puts it, 'grammars should be motivated on a language-specific basis.' Does this mean that other languages are irrelevant when one investigating a specific language? Clearly not. As Müller also puts it, 'In situations where more than one analysis would be compatible with a given dataset for language X, the evidence from language Y with similar constructs is most welcome and can be used as evidence in favour of one of the two analyses for language X.' (2015: 43)

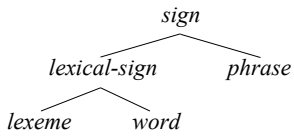
[3] There may be compelling evidence for some empty elements in some languages. Thus, Borsley (2009: section 8) argues that Welsh has phonologically empty pronouns. For general discussion of empty elements, see Müller (2016: chapter 19.2).

3. Elements

For HPSG, a linguistic analysis is a system of types, features, and constraints. Types provide a complex classification of linguistic objects, features identify their basic properties, and constraints impose further restrictions. In this section, we will explain these three elements.

There are many different kinds of types, but particularly important is the type *sign* and its various subtypes. For Ginzburg and Sag (2000), this type has the subtypes *lexical-sign* and *phrase*, and *lexical-sign* has the subtypes *lexeme* and *word*. (Types are written in lower case italics.) Thus, we have the following type hierarchy:

(4)



Both *lexeme* and *phrase* have a complex system of subtypes. The type *lexeme* and its subtypes and the associated constraints constitute the lexicon of a language, while the type *phrase* and its subtypes and the associated constraints constitute the syntax of a language. In both cases, complex hierarchies mean that the framework is able to deal with broad, general facts, very idiosyncratic facts, and everything in between. We will say more about this below.

Signs are obviously complex objects with (at least) phonological, syntactic and semantic properties. Hence, the type *sign* must have features that encode these properties. For much work in HPSG, phonological properties are encoded as the value of a feature PHON(OLOGY), whose value is an object of type *phon*, while syntactic and semantic properties are grouped together as the value of a feature SYNSEM, whose value is an object of type *synsem*. (Feature names or attributes are written in upper case.) A type and its features is a typed feature structure, and it can be represented by an attribute-value-matrix (AVM) with the type name at the top on the left hand side and the feature names below followed by their values. Thus, signs can be represented as follows:

(5)
$$\begin{bmatrix} \textit{sign} \\ \text{PHON } \textit{phon} \\ \text{SYNSEM } \textit{synsem} \end{bmatrix}$$

Specific signs will obviously have specific values for the two features. For example, we might have the following simplified AVM for the phrase *the cat*:

(6)
$$\begin{bmatrix} \textit{phrase} \\ \text{PHON 'the cat'} \\ \text{SYNSEM NP} \end{bmatrix}$$

Here, following a widespread practice, we use standard orthography instead of a real *phon* object, and we use the traditional label NP as an abbreviation for the relevant *synsem* object. We will say more about *synsem* objects shortly. First, however, we must say something about the difference between words and phrases.

In much HPSG work it is assumed that the main difference between words and phrases is that only the latter have internal structures. More precisely, they have daughters, i.e. immediate constituents, one of which may be the head. This information is encoded by further features, for Ginzburg and Sag (2000) the features DAUGHTERS (DTRS) and HEAD-DAUGHTER (HD-DTR). The value of the former is *sign*, and the value of the latter is a list of *signs*, and the value of the former is part of the list which is the value of the latter.[4] Thus, phrases have representations of the following form:

(7)
$$\begin{bmatrix} \textit{phrase} \\ \text{PHON } \textit{phon} \\ \text{SYNSEM } \textit{synsem} \\ \text{DTRS } \textit{list}(\textit{sign}) \\ \text{HD-DTR } \textit{sign} \end{bmatrix}$$

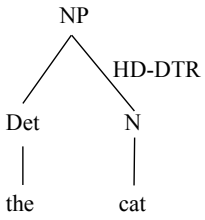
To take a concrete example the phrase *the cat* might have the fuller AVM in (8).

(8)
$$\begin{bmatrix} \textit{phrase} \\ \text{PHON 'the cat'} \\ \text{SYNSEM NP} \\ \\ \text{DTRS} < \begin{bmatrix} \text{PHON 'the'} \\ \text{SYNSEM Det} \end{bmatrix}, [1] \begin{bmatrix} \text{PHON 'cat'} \\ \text{SYNSEM N} \end{bmatrix} > \\ \text{HD\_DTR } [1] \end{bmatrix}$$

Here the two instances of the tag [1] indicate that the *synsem* object which is the second member of the DTRS list is also the value of HD-DTR. Thus, the word *cat* is the head of the phrase *the cat*. As we will see below, an object occupying more than position in a representation is a pervasive feature of HPSG. This is known as re-entrancy or structure-sharing.

AVMs like (8) can be quite hard to look at. Hence, it is common to use traditional tree diagrams instead. Thus, we might have (9) instead of (8). But one should bear in mind that AVMs are equivalent to (rooted) graphs and thus more expressive than trees.

(9)



[4] Some HPSG work, e.g. Sag (1997) has a HEAD-DAUGHTER feature and a NON-HEAD-DAUGHTERS feature, and the value of the former is not part of the value of the latter.

If the head is either obvious or unimportant, the HD-DTR annotation might be omitted. This is a convenient informal notation, but it is important to remember that it is just that and has no status within the theory.

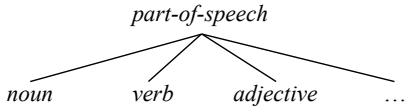
We return now to *synsem* objects. Standardly these have two features: LOCAL, whose value is a *local* object and NONLOCAL, which we will ignore for now. A *local* object has the features CAT(EGORY) and CONT(ENT), whose values are objects of type *category* and *content,* respectively, and the feature CONTEXT. In much work, a *category* object has the features, HEAD, SUBJ and COMP(LEMENT)S. HEAD takes as its value a *part-of-speech* object, while SUBJ and COMPS have a list of *synsem* objects as their value. The former indicates what sort of subject a sign requires and the latter indicates what complements it takes. In both cases the value is the empty list if nothing is required. SUBJ and COMPS are often called VALENCE features. Thus, the following AVM provides a fuller representation of signs:

(10)

$$\begin{bmatrix} \textit{sign} \\ \text{PHON } \textit{phon} \\ \text{SYNSEM} \begin{bmatrix} \textit{synsem} \\ \text{LOCAL} \begin{bmatrix} \textit{local} \\ \text{CATEGORY} \begin{bmatrix} \textit{category} \\ \text{HEAD } \textit{part - of - speech} \\ \text{SUBJ } \textit{list}(\textit{sysnem}) \\ \text{COMPS } \textit{list}(\textit{sysnem}) \end{bmatrix} \\ \text{CONTENT} \\ \text{CONTEXT} \end{bmatrix} \\ \text{NONLOCAL…} \end{bmatrix} \end{bmatrix}$$

The type *part-of-speech* has subtypes such as *noun*, *verb*, and *adjective*. In other words, we have a type hierarchy of the following form:

(11)



A particular word or phrase is well formed if it belongs to the ontology of grammar types, i.e. if it inherits from of some of the predefined types. If it is the case, the information associated with it is the result of the unification of the information associated with its various supertypes.[5] Thus, we might have the following *synsem* object for the phrase *the cat*:

[5] Unification of two AVM A and B is the smallest AVM which subsumes A and B, if it exists and A subsumes B if it contains all information contained in B (see Richter, Formal foundations chapter). Otherwise, we say that unification fails.

(12)

$$\begin{bmatrix} \textit{synsem} & \\ \text{LOCAL} & \begin{bmatrix} \textit{local} & \\ \text{CATEGORY} & \begin{bmatrix} \textit{category} \\ \text{HEAD } \textit{noun} \\ \text{SUBJ } \langle\rangle \\ \text{COMPS } \langle\rangle \end{bmatrix} \\ \text{CONTENT} & \\ \text{CONTEXT} & \end{bmatrix} \\ \text{NONLOCAL}\ldots & \end{bmatrix}$$

This ignores a number of matters including the value of CONTENT and NONLOCAL. It also ignores the fact that the type *noun* will have certain features, for example CASE, but it highlights some important aspects of HPSG analyses. Notice that (12) unifies with (10): it contains more specific information, such as HEAD noun, but no conflicting information: <> is the empty list, and it unifies with list(synsem). We should also note the linguistic objects described (be they lexemes, words or phrases) should be distinguished from their descriptions : although the description can be underspecified, it is usually assumed that the linguistic objects be totally well-typed and sort-resolved feature structures.[6]

Rather different from most of the features mentioned above are fairly traditional features like PERSON, NUMBER, GENDER, and CASE. In most HPSG work, these have as their value an atomic type, a type with no features. A simple treatment of person might have the types *first*, *second*, and *third*, and a simple treatment of number the types *sing*(*ular*) and *plur*(*al*).[7] There are also Boolean features with + and – as their values. An example is AUX used to distinguish auxiliary verbs ([AUX +]) from non-auxiliary verbs ([AUX –]).[8]

As the preceding makes clear, features in HPSG can have a number of types of value. They may have an atomic type (PERSON, NUMBER, GENDER, CASE, AUX), a feature structure (SYNSEM, LOCAL, CATEGORY, etc.), or a list of some kind (SUBJ, COMPS).[9] As we will see in section 5, HPSG also assumes features with a set as their value.

The CONTENT feature, whose value is a *content* object, highlights the importance of semantics within HPSG. But what exactly is a *content* object? Different views of semantics have been taken within the HPSG literature. Much HPSG work has assumed some version of Situation Semantics (Barwise & Perry 1983). But some work has employed so-called Minimal Recursion Semantics (Copestake et al 2005), and Sag (2010a) adopts a conventional, Montague-style possible-worlds semantics in his analysis of English filler-gap constructions. SBCG has generally employed a version of Frame Semantics. See Koenig and Richter (2018) chapter 23 of this volume for a discussion of the issues.

Finally, the CONTEXT feature is used for information structure, deixis, and more generally pragmatics (see de Kuthy this volume).

---

[6] This is relaxed in Sag 2002 analysis of unlike coordination (see Richter, this volume)

[7] In practice a more complex system of values may well be appropriate.

[8] In some recent work, e.g. Sag (2012), the feature is used to distinguish positions that only allow an auxiliary from positions that allow any verb. Within this approach auxiliaries are unspecified for AUX since they may appear in both [AUX +] and [AUX –] positions. Non-auxiliary verbs are [AUX –].

[9] A list can be analysed as a type of feature structure with the features FIRST and REST, the latter having a list as its value.

We will say more about types and features in the following sections. We turn now to constraints. These are implicational statements, saying that if a linguistic object has some property or properties then it must have some other property or properties. They take the following form:[10]

(13) $X \rightarrow Y$

Commonly X is a type and Y a feature structure, and this is the case in all the constraints that we discuss below. However, X may also be a feature structure. This is necessary, for example, in the constraints that constitute binding theory. See Branco (2018) chapter 20 of this volume Here is a very simple constraint :

(14) $\textit{phrase} \rightarrow \left[\text{COMPS } \langle\rangle\right]$

This says that a phrase has the empty list (< >) as the value of the COMPS feature, which means that it does not require any complements. Particularly important are constraints dealing with internal structure of various types of phrase. We will consider some constraints of this kind in section 5.

4. The lexicon

As noted above, the type *lexeme* and its subtypes and the associated constraints constitute the lexicon of a language. Some of the earliest work in HPSG focused on the organization of the lexicon and the question of how lexical generalizations can be captured. As we will see, the framework uses two devices here: complex hierarchies of lexical types and lexical rules.

In some frameworks, the lexicon contains not lexemes but morphemes, i.e. roots and affixes of various kinds. But almost all work in HPSG has assumed a realizational approach to morphology, in which affixes are just bits of phonology realizing certain morphosyntactic features. Within this approach, there are no morphemes, just lexemes and the words that realize them (Stump 2001, Anderson 1992). One consequence of this is that HPSG has no syntactic elements like the T(ense) and Num(ber) functional heads of Minimalism, which are mainly realized by affixes. See Crysmann (2018) chapter 22 of this volume and Davis and Koenig chapter 4 of this volume for discussion of the issues.

Probably the most important properties of any lexeme are its part of speech and its combinatorial properties. As we saw in the last section, the HEAD feature encodes part of speech information while the SUBJ and COMPS features encode combinatorial information. As we also noted in the last section, HEAD takes as its value a *part-of-speech* object and the type *part-of-speech* has subtypes such as *noun*, *verb*, and *adjective*. At least some of the subtypes have certain features. For example, in many languages the type *noun* has the feature CASE with values like *nom*(*inative*), *acc*(*usative*), and *gen*(*itive*). Thus, nominative pronouns like *I* might have a *part-of-speech* of the form in (15).

(15) $\begin{bmatrix} \textit{noun} \\ \text{CASE } \textit{nom} \end{bmatrix}$

[10] Sometimes a double-shafted arrow (⇒) is used in constraints instead of the single shafted arrow, but the double shafted arrow is also often used in lexical rules, so it seems best to use the single shafted arrow.

Similarly, in many languages the *type* verb has the feature VFORM with values like *fin*(*ite*) and *inf*(*initive*). Thus, the *part-of-speech* of the word form *be* might be (16).

(16) $\begin{bmatrix} verb \\ \text{VFORM}\ inf \end{bmatrix}$

In much the same way, the type *adjective* will have a feature distinguishing between positive, comparative, and superlative forms, in English and many other languages.

We must now say more about combinatorial properties. In much HPSG work it is assumed that SUBJ and COMPS encode what might regarded as superficial combinatorial information and more basic combinatorial information is encoded by a feature ARG(UMENT)-ST(RUCTURE). Normally the value of ARG-ST is the concatenation of the values of SUBJ and COMPS. In other words, we normally have the following situation (notice the use of re-entrancy or structure-sharing):

(17) $\begin{bmatrix} \text{ARG-ST}\ [1] \oplus [2] \\ \text{SUBJ}\ [1] \\ \text{COMPS}\ [2] \end{bmatrix}$

The appropriate features for *read* in (1a) for example would be the following:

(18) $\begin{bmatrix} \text{ARG-ST} < [1]\text{NP}, [2]\text{NP} > \\ \text{SUBJ} < [1] > \\ \text{COMPS} < [2] > \end{bmatrix}$

Feldfunktion geändert

Under some circumstances, however, we have something different. For example, it is generally assumed that null subject sentences have an element representing the understood subject in the ARG-ST list of the main verb but nothing in the SUBJ list. Thus, the verb in (1b), repeated here as (19) has the features in (20).

(19) Czytałem książkę.
write.PAST.1SG book.ACC
'I read a book.'

(20) $\begin{bmatrix} \text{ARG-ST} < \text{NP}, [1]\text{NP} > \\ \text{SUBJ} <> \\ \text{COMPS} < [1] > \end{bmatrix}$

A similar analysis is widely assumed for unbounded dependency gaps. On this analysis, the preposition *to* in (21) (which appears earlier as (2)) has the features in (22).

(21) Who did you talk to?

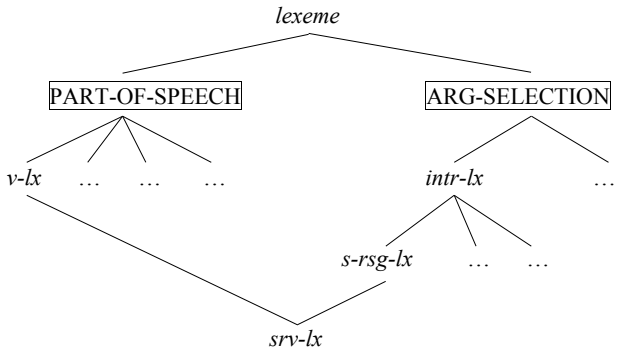
(22) $\begin{bmatrix} \text{ARG-ST} < \text{NP} > \\ \text{SUBJ} <> \\ \text{COMPS} <> \end{bmatrix}$

Other differences between SUBJ and COMPS and ARG-ST may be found in other languages. See Manning and Sag (1999) for discussion.[11]

A variety of HPSG work assumes the SUBJ and COMPs features, but some work assumes a SPR (SPECIFIER) feature instead of or in addition to the SUBJ. Where it replaces SUBJ, the idea is that subjects are one of a number of types of specifier (Sag et al. 2003). Where it is an additional feature the idea is that there are number of types of specifier but subjects are not specifiers (Abeillé & Godard 2003). There are other positions in the HPSG community. Much early work has a single feature called SUBCAT instead of SUBJ and COMPS (Pollard & Sag 1987). Essentially the same position has been adopted within Sign Based Construction Grammar, which has a single feature called VALENCE instead of SUBJ and COMPS. Obviously, there are some important issues here.

It is an important feature of lexical items that part of speech and combinatorial properties are separate matters. Members of the same part of speech can have different combinatorial properties and members of different parts of speech. Much HPSG work captures this fact by proposing that the type *lexeme* has two distinct sets of subtypes, one dealing with part-of-speech information and one dealing with argument selection information. This is why HPSG employs cross-classification, i.e. multiple classifications along different dimensions. Here is a simple illustration based on Ginzburg and Sag (2000: 20):

(23)



Upper case letters are used for the two dimensions of classification, and *v-lx*, *intr-lx*, *s-rsg-lx*, and *srv-lx* abbreviate *verb-lexeme*, *intransitive-lexeme*, *subject-raising-lexeme*, and *subject-raising-verb-lexeme*, respectively. All these types will be subject to specific constraints, For example, *v-lx* will be subject to something like the following constraint:

(24) $\textit{v-lx} \rightarrow \begin{bmatrix} \text{HEAD } \textit{verb} \\ \text{SUBJ} < \text{XP} > \end{bmatrix}$

This says that a verb lexeme has a verbal part of speech and requires a phrase of some kind as its subject. Similarly, we will have something like the following constraint for *s-rsg-lx*:

(25) $\textit{s-rsg-lx} \rightarrow \left[\text{ARG-ST} < [1], [\text{SUBJ} < [1] >] >\right]$

[11] ARG-ST is also crucial for binding theory, which takes the form of a number of constraints on ARG-ST lists See Branco (2018) chapter 20 of this volume.

This says that a subject-raising-lexeme has two arguments, a subject and a complement, and that the subject is whatever the complement requires as a subject, noted by [1]. Most of the properties of any lexeme will be inherited from its supertypes. Thus, very little information needs to be associated with specific lexemes in a system like this.

For example, for a subject-raising verb like *seem*, its PHON and CONTENT features are the following, using Minimal Recursion semantics : RELS is the attribute for the set of elementary predications associated with a word or a phrase, THEME is one of the semantic roles (see Wechsler, Koenig and Davies, this volume). *Seem* takes an infinitival VP complement.[12] Notice that no semantic role is assigned to the first argument (see Abeillé, this volume).

(26) *seem*→ *s-rsg-lx* &
$$\begin{bmatrix} \text{PHON } seem \\ \text{CAT}\left[\text{ARG-ST} < [], \text{VP}\begin{bmatrix}\text{HEAD}[\text{VFORM } inf] \\ \text{INDEX } s1\end{bmatrix} >\right] \\ \text{CONT}\begin{bmatrix}\text{INDEX s} \\ \text{RELS}\left\{\begin{bmatrix} seem\text{-}rel \\ \text{THEME } s1\end{bmatrix}\right\}\end{bmatrix}\end{bmatrix}$$

Once these more specific features are unified with features from type *s-rsg-lxm*, we get a more complete AVM like the following :

(27) *seem* →
$$\begin{bmatrix} \text{PHON } seem \\ \text{CAT}\begin{bmatrix}\text{ARG-ST} < [1], [2]\text{VP}\begin{bmatrix}\text{HEAD}[\text{VFORM } inf] \\ \text{SUBJ} < [1] > \\ \text{INDEX } s1\end{bmatrix} > \\ \text{SUBJ} < [1] > \\ \text{COMPS} < [2] >\end{bmatrix} \\ \text{CONT}\begin{bmatrix}\text{INDEX s} \\ \text{RELS}\left\{\begin{bmatrix} seem\text{-}rel \\ \text{THEME } s1\end{bmatrix}\right\}\end{bmatrix}\end{bmatrix}$$

Notice that the SUBJ feature is underspecified. Thus, *seem* combines with an infinitival complement and with any subject (nominal, verbal, sentential), provided this subject is appropriate for its complement :

[12] The entry can be modified to allow predicative complements as well : John seems tired/ in a good mood.

(28) a. John seems to be sleeping.
b. Working seems to be tiring.
c. *Working seems to be sleeping

The hierarchy of lexical types provides one way of capturing lexical generalizations. Lexical rules provide another. They are relevant especially to valence alternations such as that illustrated in the following:

(29) a. That Kim was late annoyed Lee.
b. That Sandy was there is unimportant.
c. That Lee won impressed everyone.

(30) a. It annoyed Lee that Kim was late.
b. It is unimportant that Sandy was there.
c. It impressed everyone that Lee won.

These show that verbs and adjectives which allow a clausal subject generally also allow an expletive *it* subject and a clause as an extra complement. The lexical descriptions required for the latter use can be derived from the lexical descriptions required for the former use by a lexical rule of the following form:[13]

(31) [ARG-ST <CP> ⊕ [2]] ⇒ [ARG-ST <NP[*it*] ⊕ [2] ⊕ <CP>]

The active-passive relation can be captured by a similar lexical rule. Since these rules do not change the CONTENT feature, these alternations will preserve the meaning of the verb or adjective lexeme. Thus sentences in (25) will have a different syntactic structure than their counterparts in (26) but may have the same semantic representation (they will probably have different information structures, thus different CONTEXT features).

5. Syntax

As noted above, the type *phrase* and its subtypes and the associated constraints constitute the syntax of a language. A simple hierarchy of phrase types is implicit in early HPSG but what we have called Construction-based HPSG employs complex hierarchies of phrase types comparable the complex hierarchies of lexical types employed in the lexicon.

Like Mainstream generative grammar, HPSG takes from X-bar theory the idea that the local trees that make up syntactic structures fall into a limited number of types. Unlike Mainstream generative grammar, HPSG assumes that not all phrases are headed, even if many are, and does not asume that a head is necessarily lexical. Thus among phrases there is a basic distinction between non-headed-phrases and headed-phrases. There are various kinds of headed phrase. We will consider three here. First there are head-complement-phrases,

[13] This rule assumes that clauses introduced by *that* are CPs, which, as noted in section 7, is not assumed in all HPSG work. A satisfactory analysis of this domain needs to accommodate pairs of examples which involve not a *that*-clause but an infinitive, e.g. the following:

(i) To annoy Lee is easy.
(ii) It is easy to annoy Lee.

Verbs and comlementizers are two subtypes of verbals in much HPSG work. Hence similarities between CPs and VPs are expected.

combinations of a head and its complements. These can be headed by various parts of speech, verbs, positions, adjectives, nouns, and others, and may have one complement or more than one. Next there are head-subject-phrases. Typically the head of such a phrase is a VP. However, the bracketed material in the following may well be head-subject-phrases with a non-verbal head.

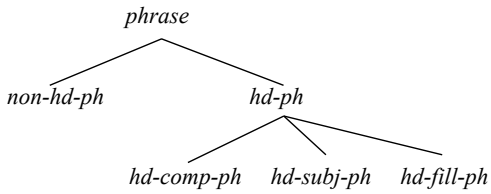
(32) With [Kim ill/in London/a candidate], anything is possible.

Finally, there are head-filler-phrases, clauses in which an initial constituent is associated with a following gap. *Wh*-interrogatives and *wh*-relatives, such as the bracketed material in the following, are typical examples.

(33) a. I'm wondering [who I talked to].
b. This is the official [who I talked to].

All this points to the following simple type hierarchy:

(34)



Each of these types is associated with a constraint capturing its distinctive properties.

Consider first the type *hd-ph*. Here we need a constraint capturing what all headed-phrases have in common. This is essentially that they have a head, with which they share certain features.[14] But what features? One view developed in Ginzburg and Sag (2000) is that they share all features except where some other constraint requires a difference. This is captured by the Generalized Head Feature Principle, which takes the following form:

(35) $\textit{hd-ph} \rightarrow \begin{bmatrix} \text{SYNSEM} /[1] \\ \text{HD-DTR}\,[\text{SYNSEM} /[1]] \end{bmatrix}$

The slashes (/) here indicate that this is a default constraint. Thus, it says that a headed-phrase and its head daughter have the same SYNSEM values unless some other constraint requires something different.

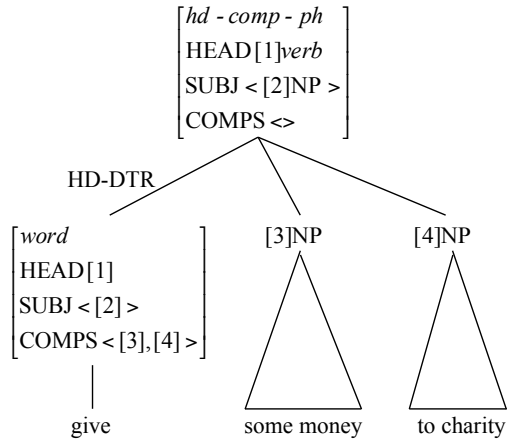
Each of the three subtypes of *hd-ph*, is subject to a constraint that requires something different. Here is a constraint on the type *hd-comp-ph* (with SYNSEM abbreviated as SS):

(36) $\textit{hd-comp-ph} \rightarrow \begin{bmatrix} \text{HD-DTR}\,[1]\begin{bmatrix} \textit{word} \\ \text{COMPS} < [2],\ldots,[n] > \end{bmatrix} \\ \text{DTRS} < [[1], \text{SS}[2]], \ldots, [\text{SS}[n]] > \end{bmatrix}$

[14] The term head is used in much the same way in HPSG as in Jackendoff (1977). In Minimalism it is only applied to lexical elements which share features with their mother and not to all expressions that have this property.

This ensures that a head-complement-phrase has a word as a head daughter and non-head daughters with the *synsem* properties that appear in the head's COMPS list. Notice that nothing is said about the SYNSEM value of the phrase. This will be identical to that of the head daughter except that it will be [COMPS <>], as required by the constraint in (14). Among other things, it will have the same value for SUBJ as the head daughter. Notice also that there is no assumption in HPSG that all branching is binary. Hence, where a head takes two complements, both will be its sisters. Here is an example of the sort of structures that the analysis licenses:

(37)

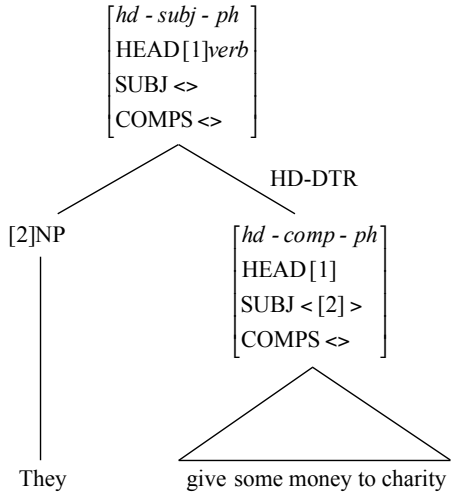


We turn now to the type *hd-subj-ph*. Here we need a constraint which mentions the SYNSEM value of the phrase as follows:

(38) *hd-subj-ph* → [SS | LOC | CAT | SUBJ <>; HD - DTR [1][*phrase*; SUBJ < [2] >]; DTRS < [SS[2]], [1] >]

This ensures that a head-subject-phrase is SUBJ <> and has a phrase as a head daughter and a non-daughter with the *synsem* properties that appear in the head's SUBJ list. It licenses structures like the following:

(39)



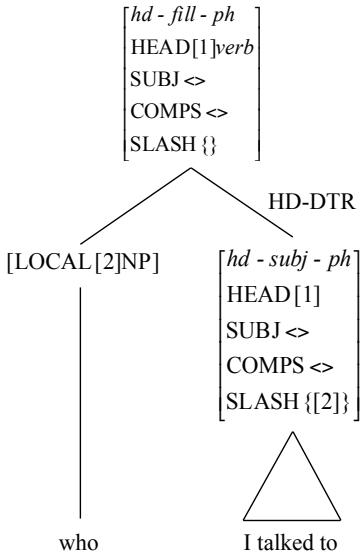
Finally we consider the type *hd-fill-ph*. This involves the feature SLASH, one of the features which makes up the value of the feature NONLOCAL introduced earlier (12). Its value is a set of *local* feature structures and it encodes information about unbounded dependency gaps (see Borsley & Crysmann, this volume chapter 14). Here is the relevant constraint:[15]

(40) $$\textit{hd-fill-ph} \rightarrow \begin{bmatrix} \text{SLASH}\,[1] \\ \text{HD - DTR}\,[2]\begin{bmatrix} \textit{phrase} \\ \text{SLASH}\,\{[3]\} \cup [1] \end{bmatrix} \\ \text{DTRS} < [\text{LOCAL}\,[3]], [2] > \end{bmatrix}$$

This says that a head-filler-phrase has a head daughter, which is a phrase with a SLASH set which is the SLASH set of the head-filler-phrase plus one other *local* feature structure, and a non-head daughter, whose LOCAL value is the additional *local* feature structure of the head daughter. [1] is normally the empty set. Here is a typical head-filler-phrase:

[15] Notice that the root category does not have to have an empty SLASH list, thus allowing for multiple extractions (*Paul, who could we talk to about* ?).

(41)



Notice that the head daughter in a head-filler-phrase is not required to have an empty SUBJ list (it is not marked as [SUBJ <>]) and hence does not have to be a head-subject-phrase. It can also be a head-complement phrase (a VP), as in the following:

(42) I'm wondering [who [to talk to]].

The constraints that we have just discussed are rather like phrase structure rules. This led Ginzburg and Sag (2000) to use an informal notation which reflects this. This involves the phrase type on the first line followed by a colon, and information about the phrase itself and its daughters on the second line separated by an arrow and with the head daughter identified by '**H**'. Thus, instead of (43) one has (44).

(43) $$\textit{phrase} \rightarrow \begin{bmatrix} \text{SYNSEM X} \\ \text{DTRS} < [1]\text{Y}, \text{Z} > \\ \text{HD - DTR [1]} \end{bmatrix}$$

(44) *phrase*:

[X] → **H**[Y], [Z]

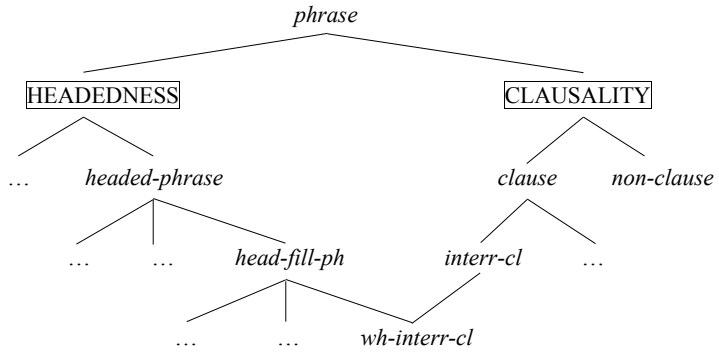
Notice that while the arrow in (43) has the normal 'if-then' interpretation, the arrow in (44) means 'can consist of'. In some circumstances this informal notation may be more convenent than the more formal notation used above.

In the preceding discussion, we have ignored the semantics of the phrase. Leaving aside quantification and other complex matters, the CONTENT of a headed phrase can be handled via two semantic principles, assuming INDEX and RELations as in MRS (26 above) : a

semantic inheritance principle (the INDEX of a headed-phrase is the INDEX of its HEAD-DTR) and a semantic compositionality principle (the RELS of a phrase is the set union of the RELS of its DTRS)(see Richter, this volume).

The type hierarchy in (34) is simplified in a number of respects. It includes no non-headed-phrases.[16] It also ignores various other subtypes of headed-phrase, some of which are discussed in the next section. Most importantly, it is widely assumed that the type *phrase* like the type *lexeme* has two distinct sets of subtypes, one dealing with head-dependent relations and the other dealing with the properties of various types of clauses. Here is a simplified illustration:

(45)



Here *wh-interr-cl* is identified as a subtype of *head-fill-ph* and a subtype of *interr*(*ogative*)*-cl*. As such it has both the properties required by the constraint in (36) and certain properties characteristic of interrogative clauses, most obviously interrogative semantics.

We must now say something about word order. In much HPSG work this is a matter of phonology, more precisely a matter of the relation between the PHON value of a phrase and the PHON values of its daughters. Consider, for example, a phrase with two daughters, each with its own PHON value. The PHON value of the phrase will be the concatenation of the PHON values of the daughters. Clearly they can be concatenated in two ways as follows, or their order may be left unspecified for 'free' word order':

(46)

$$\begin{bmatrix} \text{PHON } [1] \oplus [2] \\ \text{DTRS} < [\text{PHON } [1]], [\text{PHON } [2]] > \end{bmatrix} \quad \begin{bmatrix} \text{PHON } [2] \oplus [1] \\ \text{DTRS} < [\text{PHON } [1]], [\text{PHON } [2]] > \end{bmatrix}$$

Within this approach, the following English and Welsh examples might have exactly the same analysis (a head-adjunct phrase) except for their PHON values:

(47) a. black sheep
b. defaid du
sheep.PL black
'black sheep'

[16] The most important type of non-headed phrase is coordinate structure. See Abeillé and Chaves (2018) chapter 16 of this volume for discussion.

Similarly, a prepositional phrase in English and a postpositional phrase in Japanese might have the same analysis (a head-complement phrase) apart from their PHON values. Ordering rules are commonly written with < ('precedes'). Thus, languages with head-complement order might have the rule in (48a) and languages with complement-head order the rule in (48b).

(48) a. [COMPS <..., [1], ...>] < [SYNSEM [1]]
b. [SYNSEM [1]] < [COMPS <..., [1], ...>]

But it should be remembered that this is an abbreviation of a more complete notation as in (46).[17]

Not all pairs of expressions which might be seen as differing just in word order have the same analysis apart from their PHON values. Consider, for example, the following:

(49) a. Kim is late.
b. Is Kim late?

Here we have a declarative and a related interrogative. They differ semantically and in word order, but for most work in HPSG they also differ in their syntactic structures. (49a) is a head-subject-phrase much like that in (39). Clauses like (49b), on the other hand, are standardly seen as ternary branching phrases in which both the subject and the complement are a sister of the auxiliary. This requires an additional phrase type, which might be called a head-subject-complement-phrase.[18]

6. Further Syntactic Structures

Head-complement-phrases, head-subject-phrases, and head-filler-phrases are perhaps the most important types of syntactic structure, but there are others that are of considerable importance. Here we will say something about three: head-adjunct-phrases, head-specifier-phrases, and head-marker-phrases.

Adverbs and adverbial PPs within VPs and attributive adjectives and relative clauses within NPs are commonly viewed as adjuncts. Thus, the bracketed material in the following are head-adjunct-phrases:

(50) a. Kim [[slowly] [read the book]]
b. Kim [[met Lee] [in the pub]]
c. a [[new] [book about syntax]]
d. a [[book about syntax] [which impresses everyone]]

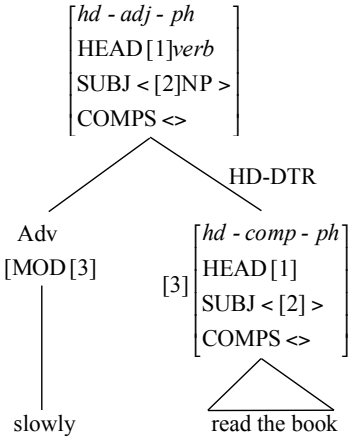
In much HPSG work, adjuncts select the heads they combine with through a feature MOD (modifies) whose value is a *synsem* object. Thus, (50a) involves a schematic structure of the following form:

---

[17] An alternative notation, provided different Daughters are distinguished with different names, could be :
a. HD-DTR < COMPS-DTRS
b. COMPS-DTRS < HD-DTR

[18] In Ginzburg and Sag (2000) it is called *sai-phrase*. In some HPSG work, e.g. Sag, Wasow, and Bender (2003), examples like (42b) are analysed as involving an auxiliary verb with two complements and no subject. This approach has no need for an additional phrase type, but it requires an extra set of descriptions for auxiliary verbs.

(51)



In the case of adverbs, adverbial PPs, and attributive adjectives, it is a simple matter to assign an appropriate value to MOD, and this value can be underspecified to account for the polymorphism of certain adverbs which can modify all (major) categories (Abeillé & Godard 2003). In the case of relative clauses, it is more complex because the value must match that of the wh-element if there is one or the gap if there isn't. In (50d) this is reflected in the fact that the verb in the relative clause is the singular *impresses* and not the plural *impress*. See the Borsley and Crysmann (2018) chapter 13 of this volume and Arnold and Godard (2018) chapter of 14 this volume for some discussion.

Notice also that in adjunct-head phrases, the adjunct is not a syntactic head but may well be the semantic predicate, or the semantic head. This is an example of the non isommorphism between syntax and semantics in HPSG.

Although an adjunct analysis of adverbial PPs seems quite natural, it has been argued in some HPSG work that they are in fact optional complements of verbs (see e.g. Bouma, Malouf and Sag 2001 and Ginzburg and Sag 2000: 168, fn.2). On this view, *in the pub* in (50b) is much like the same phrase in the following, where it is clearly a (predicative) complement:

(52) Kim is in the pub.

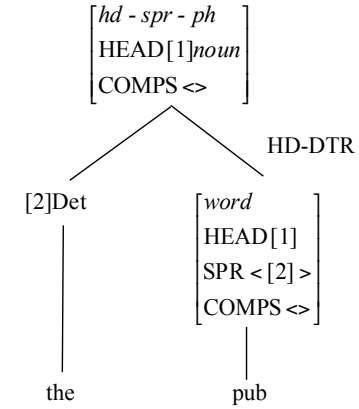
Various arguments have been advanced for this position, but it is controversial and it is rejected by Levine (2002) and Levine and Hukari (2006) and Chaves (2009). There is an unresolved issue here.[19]

As noted earlier, some HPSG work assumes a feature SPR (SPECIFIER) which is realized by various categories. In some work subjects are analysed as specifiers, but in other work they are realizations of a SUBJ(ECT) feature, as discussed in the last section. For some HPSG work, e.g. Pollard and Sag (1994: 9.4), Sag *et al.* (2003: 4.3), determiners within NPs are an important example of specifiers. On this view, (53) has the schematic structure in (54).

(53) the pub

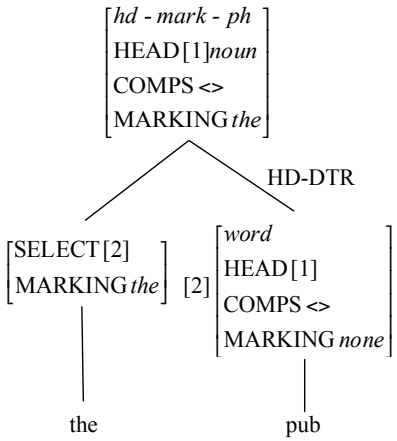
[19] It might be argued that some adverbials PPs are adjuncts and others are complements, depending on word order, case etc (see for example Abeillé & Hassamal 2014 for such a distinction among Mauritian adverbs).

(54)



Some recent work, e.g. Sag (2012) has adopted a rather different view of at least some determiners, that they are what are known as markers, a notion first introduced in Pollard and Sag (1994: 1.6). These are non-heads which select the head that they combine with through a SELECT feature but determine the MARKING value of their mother.[20] Within this approach (48) has the following schematic structure:

(55)



A marker analysis was originally proposed for complementizers. However, they have also been analysed as heads within HPSG, e.g. in Sag (1997) and Ginzburg and Sag (2000). Thus, there are some unresolved issues here.

[20] Work which assumes the SELECT feature also uses it instead of MOD for adjuncts, which may also be analysed as markers (Van Eynde 1998).

7. Further Issues

There are many other aspects of HPSG that could be discussed in this chapter, but we will focus on just two: what are known as order domains, and the distinguishing properties of the SBCG version of HPSG.

7.1. Order domains

We noted above that much HPSG work views word order as a matter of phonology, specifically a matter of the relation between the PHON value of a phrase and the PHON values of its daughters. Some work in HPSG argues that this is too simple in that it ties the observed order too closely to constituent structure. Consider the following examples:

(56) a. A man who looked like Churchill came into the room.
b. A man came into the room who looked like Churchill.

One might assume that these show different observed orders because they have different structures (Kiss 2005) but one might also want to claim that they have the same constituent structure. This is possible if the observed order is not a simple reflection of constituent structure. Much work in HPSG has proposed that the observed order is a reflection not of the constituent structure of an expression but of a separate system of order domains (see Reape 1992, Müller 1996, Kathol 2000). Within this approach, ordering rules may order non sister elements, as long as they belong to the same order domain: the constituent structure of an expression can be encoded as the value of a DTRS (DAUGHTERS) feature and the order domain as the value of a DOM(AIN) feature. Adopting it, one might propose that (56b) has the schematic analysis in (57).

(57)
$$\begin{bmatrix} \text{SYNSEM S} \\ \text{DTRS} < [\textit{a man who looked like Churchill}], [\textit{came into the room}] > \\ \text{DOM} < [\textit{a man}], [\textit{came into the room}], [\textit{who looked like Churchill}] > \end{bmatrix}$$

Here the clause has two daughters but three domain elements. The simpler example in (56a) will have two daughters and two domain elements.

It is worth noting that this approach allows a different analysis for interrogatives like (49b). It would be possible to propose that an analysis in which they have two daughters and three domain elements as follows:

(58)
$$\begin{bmatrix} \text{SYNSEM S} \\ \text{DTRS} < [\textit{Kim}], [\textit{is late}] > \\ \text{DOM} < [\textit{is}], [\textit{Kim}], [\textit{late}] > \end{bmatrix}$$

As far as we are aware, no one has proposed such an analysis for English interrogatives, since one wants to account for the fact that the two versions have different syntactic and semantic properties, but essentially this analysis is proposed for German interrogatives in Kathol (2000).

Order domains seem most plausible as an approach to the sorts of discontinuity that are found in so-called nonconfigurational languages such as Warlpiri (Donohue and Sag (1999).). However, they may well have a role to play in more familiar languages (Bonami et al. 1999). But exactly how much of a role they should play in syntax is an unresolved matter.

One might wonder whether a version of HPSG that includes order domains is still a monostratal framework. It remains a framework in which linguistic expressions have a single constituent structure. However, it does have a second important level of representation, which makes available a variety of analyses which would otherwise not be possible. Whether the framework is still monostratal depends on how exactly the term is used. We will not take a stand on this.

6.2. Sign-Based Construction Grammar

The SBCG version of HPSG will be discussed in some detail in the next chapter (Flickinger et al, this volume). Here we will just highlight the central difference between this approach and earlier work. The term *construction* is widely used in connection with the earlier Construction-based HPSG, but within that work constructions are just types of sign. In contrast, for SBCG signs and constructions are quite different objects.

For SBCG constructions are objects which associate a MOTHER sign with a list of DAUGHTER signs, one of which may be a HEAD-DAUGHTER. They take the following form:

(59)
$$\begin{bmatrix} cx \\ \text{MTR } sign \\ \text{DTRS } list(sign) \\ \text{HD - DTR } sign \end{bmatrix}$$

Constructions are utilized by the Sign Principle, which can be formulated as follows:[21]

(60) Signs are well formed if either (a) they match some lexical entry, or (b) they match the mother of some construction.

Constructions and the Sign Principle are properties of the SBCG which are lacking in earlier work. Essentially, then, they are complications. But they allow simplifications. In particular, they mean that signs do not need to have the features DTRS and HD-DTR. This in turn allows the framework to dispense with the feature SYNSEM and the type *synsem*. These elements are necessary in earlier HPSG because taking the value of COMPS to be a list of signs would incorrectly predict that heads may select complements not just with specific syntactic and semantic properties but also with specific kinds of internal structure. For example, it would allow a verb to select as its complement a phrase whose head has a specific type of complement. To exclude this possibility, earlier versions of HPSG seem to need SYNSEM and *synsem*. In SBCG it is excluded by the assumption that signs do not have the features DTRS and HD-DTR and so SYNSEM and *synsem* are unnecessary. Thus, SBCG is both more complex and simpler than earlier versions of the framework. This means that it is not easy to evaluate it.

[21] Lexical rules are analysed in SBCG as lexical constructions. Thus, (b) covers derived words as well as phrases.

REFERENCES

Abeillé A. and D. Godard, (2003), The syntactic flexibility of French degree adverbs, *Proceedings* International HPSG Conference, S. Muller (ed), CSLI Publications, 26-46.

Abeillé A. and R. D. Borsley (2008), Comparative correlatives and parameters , *Lingua* 118, 1139-1157.

Abeillé A., R. D. Borsley and M-T. Espinal (2006), The syntax of comparative correlatives in French and Spanish, *Proceedings* International HPSG Conference, S. Muller (ed), CSLI online Publications, 6-26.

Anderson S. (1991), A-morphous morphology, Cambridge University Press.Barwise J. and J. Perry (1983), *Situations and Attitudes*, Cambridge : MIT Press

Bender, E. M. (2016), Linguistic Typology in Natural Language Processing, *Linguistic Typology* 20(3):645-660.

Bender, E. M., S. Drellishak, A. Fokkens, L. Poulson, and S. Saleem. (2010), Grammar customization, *Research on Language and Computation* 8(1): 23-72.

Bonami, Olivier, Danièle Godard, and Jean-Marie Marandin. (1999), Constituency and word order in French subject inversion. In *Constraints and resources in natural language semantics*, edited by Gosse Bouma, Erhard Hinrichs, Geert-Jan Kruijff and Richard T. Oehrle, Stanford: CSLI Publications, 21-40

Borsley, R. D. (2009), On the superficiality of Welsh agreement, *Natural Language and Linguistic Theory* 27, 225-265.

Borsley, R. D. (2011), Constructions, functional heads and comparative correlatives, in O. Bonami and P. Cabredo Hofherr (eds.), *Empirical Issues in Syntax and Semantics* 8, 7-26.

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

Copestake, A., Flickinger, D., Pollard, C. et al. (2005), Minimal Recursion Semantics: An Introduction, *Research on Language and Computation* 3: 281-332.

Donohue, C. and I. A. Sag (1999), Domains in Warlpiri, in *Sixth International Conference on HPSG*, 101–106. Edinburgh.

Gazdar, G. E. Klein, G. K. Pullum and I. A. Sag (1985), *Generalised Phrase Structure Grammar*, Oxford: Blackwell.

Ginzburg, J. and I. A. Sag (2000), *Interrogative Investigations: The Form, Meaning and Use of English Interrogatives*, Stanford: CSLI Publications.

Jackendoff, R. S. (1977), *X Syntax: A Study of Phrase Structure*. Cambridge, MA: MIT Press.

Kathol, A. (2000), *Linear Syntax*, Oxford: Oxford University Press.

Kiss T. (2005), Semantic Constraints on Relative Clause Extraposition, Natural Language and Linguistic Theory 23(2):281-334

Manning, C., and I.A. Sag (1999), Dissociations between argument structure and grammatical relations, in A. Kathol, J.-P. Koenig, and G. Webelhuth (eds.), *Lexical and Constructional Aspects of Linguistic Explanation*, Stanford: CSLI Publications.

Michaelis, L. A. & K. Lambrecht. (1996). Toward a construction-based model of language function: The case of nominal extraposition. *Language* 72:215-247.

Müller, S. (1996), The Babel-System – An HPSG Fragment for German, a Parser and a Dialog Component, *Proceedings of the Fourth International Conference on the Practical Application of Prolog*, pages 263–277, London.

Müller, S. (2015), The CoreGram project: Theoretical linguistics, theory development and verification. *Journal of Language Modelling* 3, 21–86.

Müller, S. (2016), *Grammatical Theory: From Transformational Grammar to Constraint-Based Approaches* (Textbooks in Language Sciences 1). Berlin: Language Science Press.

Oepen, Stephan, Daniel P. Flickinger, Kristina Toutanova & Christopher D. Manning. (2004). LinGO Redwoods: A rich and dynamic treebank for HPSG. *Research*

on Language and Computation 2(4). 575–596.
Pollard, C. and I. A. Sag (1987), *Information-based Syntax and Semantics. Volume I: Fundamentals*, Stanford: CSLI publications.
Pollard, C. and I. A. Sag (1994), *Head-driven Phrase Structure Grammar* Chicago: University of Chicago Press.
Postal, P. M. (2003), '(Virtually) conceptually necessary', *Journal of Linguisics* 39, 599-620.
Pullum, G. K. and B. C. Scholz (2001), On the distinction between model-theoretic and generative-enumerative syntactic frameworks, in P. de Groote, G. Morrill, and C. Retoré (eds) *Logical Aspects of Computational Linguistics: 4th International Conference,* number 2099 in *Lecture Notes in Artificial Intelligence*, Berlin and New York: Springer, 17–43.
Reape, M. (1992), *A Formal Theory of Word Order: A Case Study in West Germanic*, PhD dissertation, University of Edinburgh.
Sag, I. A. (1997), English relative clause constructions, *Journal of Linguistics* 33, 431-484.
Sag I. A. (2002) Coordination and underspecification, in J-B. Kim, S. Wechsler (eds), Proceedings International HPSG Conference, Stanford : CSLI Publications, 267-291.
Sag, I. A. (2012), Sign-based construction grammar: an informal synopsis, in H. C. Boas and I. A. Sag (eds.), *Sign-based Construction grammar*, Stanford: CSLI Publications, pp. 69-202.
Sag I.A. and J. Fodor (1994), Extraction without traces, WCCFL 13, 365-384
Sag, I. A. and T. Wasow (2011), Performance-compatible competence grammar, in R. D. Borsley and K. Börjars (eds.), *Non-Transformational Syntax: Formal and Explicit Models of Grammar*, Oxford: Blackwell, 359-377.
Sag, I. A. and T. Wasow (2015), Flexible processing and the design of grammar, *Journal of Psycholinguistic Research*. 44: 47-63.
Sag, I. A., T. Wasow, and E. M. Bender (2003), *Syntactic Theory*, Second Edition, Stanford: CSLI Publications.
Stump G. (2001), *Inflectional morphology: A theory of paradigm structure*. Cambridge University Press.