## Chapter 1

# Basic properties and elements

Anne Abeillé

Université Paris Diderot

Robert D. Borsley

University of Essex

Head-driven Phrase Structure Grammar (HPSG) is a declarative and monostratal version of generative grammar, in which linguistic expressions have a single relatively simple constituent structure. It seeks to develop detailed formal analyses using a system of types, features, and constraints. Constraints on types of *lexical-sign* are central to the lexicon of a language and constraints on types of *phrase* are at the heart of the syntax, and both lexical and phrasal types include semantic and phonological information. Different versions of the framework have been developed, including versions in which constituent order is a reflection not of constituent structure but of a separate system of order domains and the Sign-Based Construction Grammar version, which make a fundamental distinction between signs of various kinds and the constructions which license them.

Fix the label for page-hfp

## 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 & Sag (1987: 1) emphasized, by other frameworks like Lexical-Functional Grammar (LFG) (Bresnan 1982), as well. Naturally it has changed in various ways over the decades. This is discussed in much more detail in the next

Anne Abeillé & Robert D. Borsley. 2020. Basic properties and elements. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 3–33. Berlin: Language Science Press. DOI: ??

chapter (Flickinger, Pollard & Wasow 2020, Chapter 2 of this volume), 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 & Sag (1987) and Pollard & 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 (Flickinger 1987; Flickinger et al. 1985). Next there is what is sometimes called Constructional HPSG, the framework adopted in Sag (1997); Ginzburg & Sag (2000), and much other work. Unlike earlier work this uses a rich hierarchy of phrase-types. This is why it is called constructional[2]. 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, and it was identified as such in Sag (2010: 486). Its central feature is the special status it assigns to constructions. In earlier work they are just types of sign, but for SBSG signs and constructions are quite different objects. In spite of this difference, most analyses in Constructional HPSG could probably be translated into SBCG and vice versa. In this chapter we will concentrate on the ideas of Constructional 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.

Add sec refs

The chapter is organized as follows. In section 2, we set out the properties that characterize the approach and 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 basics of the HPSG approach to syntax. In section 6, we look at some further syntactic structures, and in section 7, we consider some further topics, including SBCG. Finally, in section 8, we summarize the chapter.

[1] As discussed in Richter (2020), 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.

[2] As discussed below, HPSG has always assumed a rich hierarchy of lexical types. One might argue, therefore, that it has always been constructional.

## 2 Properties

Perhaps the first thing to say about HPSG is that it is a form of generative grammar in the sense of Chomsky (1957). 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, belonging to what Pullum & Scholz (2001) call 'Model Theoretic syntax'. 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 conforms to all relevant constraints.[3] 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 & Scholz (2001); Postal (2003) and Sag & 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. add sec ref

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

[3] In most HPSG work, 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 (Malouf 2003). However, see Müller & Kasper 2000; Oepen et al. 2004 for an HPSG parser with probabilities or weighted constraints.

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. Similarly, it is not assumed that because some languages like English 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.
read.PST.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) What did you say?

On this view, the verb *say* in (2) does not have an empty complement. There is, however, some debate here (Sag & Fodor 1994; Müller 2014; Borsley & Crysmann 2020: Section 3, Chapter 13 of this volume).

A further important feature of HPSG is a rejection of the Chomskyan idea that grammatical phenomena can be divided into a core, which merits serious investigation, and a periphery, which can be safely ignored.[4] 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 2006; Abeillé &

[4] This is not to deny that some constructions are more canonical and more frequent in use than others and that this may be important in various ways.

Borsley 2008; Borsley 2011), and the *X's Way* construction (Kay & Fillmore 1999; Sag 2012: 7.4). As we will see, HPSG is an approach which is able to accommodate broad linguistic generalizations and highly idiosyncratic facts and everything in between.[5]

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 (see Crysmann 2020, Chapter 21 of this volume, Koenig & Richter 2020, Chapter 22 of this volume). 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 and 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 & 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 1996; 2015; Copestake 2002; Bender et al. 2010; Bender 2016, and Bender & Emerson 2020, Chapter 25 of this volume).

Another property 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.[6] 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, illustrated in (1b). This feature of HPSG stems largely from considerations about acquisition (Müller 2016: Chapter 19; **chapters/acquisition**, Chapter **??** of this volume, Borsley & Müller (2020: Section 5.2), Chapter 29 of this volume). 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

[5] Idioms have also been an important focus of research in HPSG. See e.g. Richter & Sailer 2009; Kay & Michaelis 2017, and Sailer (2020), Chapter 17 of this volume.

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

of individual languages and for how differences between languages are viewed.

A related property 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 investigates 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 Elements

For HPSG, a linguistic analysis is a system of types (or sorts), 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. We note at the outset that HPSG distinguishes between the linguistic objects (lexemes, words phrases, etc.) and descriptions of such objects. Linguistic objects must have all relevant properties of their description and cannot be underspecified in any way. Descriptions in contrast can be underspecified and in fact always are.

There are many different kinds of types, but particularly important is the type sign and its various subtypes. For Ginzburg & Sag (2000: 19), 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 in Figure 1.

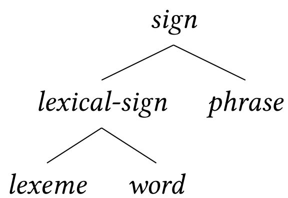


Figure 1: A hierarchy of types of signs

*Lexeme*, *word* and *phrase* have a complex system of subtypes. The type *lexical-sign*, its subtypes, and the constraints on them are central to the lexicon of a language, while the type *phrase*, its subtypes, and the constraints on them are at the heart of the syntax. 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 a list of objects 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 or attributes are written in small caps.) A type has certain features associated with it, and each feature has a value of some kind. A bundle of features can be represented by an attribute-value-matrix (AVM) with the type name at the top on the left hand side and the features below followed by their values. Thus, signs can be described as follows:

(4) $\begin{bmatrix} \textit{sign} & \\ \textsc{phon} & \text{list}(\textit{phon}) \\ \textsc{synsem} & \textit{synsem} \end{bmatrix}$

"list" is in textit in 6 tho

The descriptions of 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*:

(5) $\begin{bmatrix} \textit{phrase} & \\ \textsc{phon} & \langle \text{the,cat} \rangle \\ \textsc{synsem} & \text{NP} \end{bmatrix}$

Here, following a widespread practice, we use standard orthography instead of real *phon* objects[7],

Stefan: Why is information structure and coordination relevant to PHON?

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 phrases.

A central feature of phrases is that they have internal constituents. 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 & Sag (2000:

[7] See **BK94a-u**, Höhle (1999) and De Kuthy (2020), Chapter 23 of this volume, Abeillé & Chaves (2020), Chapter 16 of this volume.

29) the features DAUGHTERS (DTRS) and HEAD-DAUGHTER (HD-DTR). The value of the latter is a *sign*, and the value of the former is a list of *signs*, which includes the value of the latter.[8] Thus, phrases take the form in (6a), and headed-phrases the form in (6b):

(6) a. $\begin{bmatrix} \textit{phrase} & \\ \textsc{phon} & \textit{list(phon)} \\ \textsc{synsem} & \textit{synsem} \\ \textsc{dtrs} & \textit{list(sign)} \end{bmatrix}$ b. $\begin{bmatrix} \textit{headed-phrase} & \\ \textsc{phon} & \textit{list(phon)} \\ \textsc{synsem} & \textit{synsem} \\ \textsc{dtrs} & \textit{list(sign)} \\ \textsc{hd-dtr} & \textit{sign} \end{bmatrix}$

To take a concrete example the phrase *the cat* might have the fuller AVM in (7).

(7) $$\begin{bmatrix} \textit{phrase} & \\ \textsc{phon} & \langle \text{the,cat} \rangle \\ \textsc{synsem} & \text{NP} \\ \textsc{dtrs} & \left\langle \begin{bmatrix} \textsc{phon} & \langle \text{the} \rangle \\ \textsc{synsem} & \text{Det} \end{bmatrix}, \boxed{1} \begin{bmatrix} \textsc{phon} & \langle \text{cat} \rangle \\ \textsc{synsem} & \text{N} \end{bmatrix} \right\rangle \\ \textsc{hd-dtr} & \boxed{1} \end{bmatrix}$$

Here the two instances of the tag $\boxed{1}$ 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*. An object occupying more than one position in a representation, either as a feature value or as part of a feature value, for example $\boxed{1}$ in (7), is known as re-entrancy or structure-sharing. As we will see below, it is a pervasive feature of HPSG.

Most HPSG work on morphology has assumed a realizational approach, in which there are no morphemes (see **chapters/chap-morphology**, Chapter **??** of this volume). Hence, words do not have internal structures in the way that phrases do. However, it is widely assumed that lexemes and words that are derived through a lexical rule have the lexeme from which they are derived as a daughter (see Koenig (1999) and below section 4.3). Hence, the DTRS feature is relevant to words as well as phrases.

add sec ref

[8] 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. The sign that is the value of HD-DTR can be a word or a phrase. Within Minimalism, the term 'head' is only applied to words. On this usage, the value of HD-DTR is either the head or a phrase containing the head. But there are good reasons for not adopting this usage, for example the fact that the head can be an unheaded phrase for example a coordination (see Abeillé & Chaves 2020, Chapter 16 of

AVMs like (7) can be quite hard to look at. Hence, it is common to use traditional tree diagrams instead. Thus, we might have a tree-like representation in Figure 2 instead of (7). But one should bear in mind that AVMs correspond to (rooted) graphs and provide richer descriptions than traditional phrase structure trees, with richer node labels and edge labels, and with shared feature values between nodes. Thus, at each node, all kinds of information are available, not just syntax but also semantics and phonology.[9]

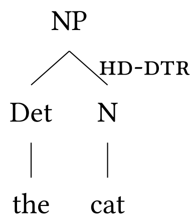


Figure 2: A simple tree for *the cat*

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 deal with in section 5. add sec ref 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.[10] 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. It is generally assumed that the subj list never has more than one member. subj and comps are often called *valence* features. Thus, the following AVM provides a fuller representation of signs:

this volume). So we will say that the value of hd-dtr is the head. See Jackendoff (1977: 30) for an early discussion of the term.

[9] This differs from Lexical Functional Grammar for instance, which distributes the information between different kinds of structures (see Wechsler & Asudeh 2020, Chapter V of this volume).

[10] Words also have a morph attribute that we ignore here (see Crysmann 2020, Chapter 21 of this volume).

(8)
$$\begin{bmatrix} \textit{sign} & \\ \text{PHON} & \textit{list(phon)} \\ \text{SYNSEM} & \begin{bmatrix} \textit{synsem} & \\ \text{LOCAL} & \begin{bmatrix} \textit{local} & \\ \text{CATEGORY} & \begin{bmatrix} \textit{category} & \\ \text{HEAD} & \textit{part-of-speech} \\ \text{SUBJ} & \textit{list(synsem)} \\ \text{COMPS} & \textit{list(synsem)} \end{bmatrix} \\ \text{CONTENT} & \\ \text{CONTEXT} & \end{bmatrix} \\ \text{NONLOCAL}\ldots & \end{bmatrix} \end{bmatrix}$$

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:

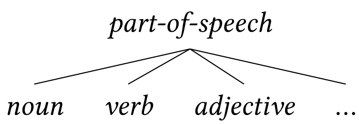


Figure 3: A hierarchy for part of speech

The type hierarchy can be viewed as an ontology of possible objects in the language. A particular word or phrase must instantiate one of the maximal (most specific) types and have the properties specified for it and all its supertypes.[11] We might have a *synsem* object of the following form for the phrase *the cat*:

add pages

(9)
$$\begin{bmatrix} \textit{synsem} & \\ \text{LOCAL} & \begin{bmatrix} \textit{local} & \\ \text{CATEGORY} & \begin{bmatrix} \textit{category} & \\ \text{HEAD} & \textit{noun} \\ \text{SUBJ} & \langle\rangle \\ \text{COMPS} & \langle\rangle \end{bmatrix} \\ \text{CONTENT} & \\ \text{CONTEXT} & \end{bmatrix} \\ \text{NONLOCAL}\ldots & \end{bmatrix}$$

[11] AVMs associated with types used to be combined by type unification (Pollard & Sag 1987). See Richter (2020), Chapter 3 of this volume.

This ignores a number of matters including the value of CONTENT, CONTEXT, 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 (9) is compatible with the SYNSEM feature in (8): it contains more specific information, such as HEAD noun, but no conflicting information: ⟨⟩ is the empty list, and is compatible with *list(synsem)*.

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)*.[12] 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 –]).[13]

As the preceding makes clear, features in HPSG can have a number of kinds 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).[14] As we will see in section 5, HPSG also assumes features with a set as their value. add sec ref

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), while some others use Logical Resource Semantics (Richter & Sailer 2004). Sag (2010: 501) adopts a conventional, Montague-style possible-worlds semantics in his analysis of English filler-gap constructions and SBCG (section 7.2) has generally employed a version of Frame Semantics. See Koenig & Richter (2020), Chapter 22 of this volume for a discussion of the issues. ref from hpsg-handbook-bibliography.b add secref

Finally, the CONTEXT feature is used for information structure, deixis, and more generally pragmatics (see de Kuthy, this volume, chapter 23).

We will say more about types and features in the following sections. We turn

[12] In practice a more complex system of values may well be appropriate (Flickinger 2000).

[13] In some recent work, e.g. Sag (2012: 157–162), Sag et al. (2020), the feature is used to distinguish positions that only allow an auxiliary from positions that allow any verb. Within this approach auxiliaries (except support do) are unspecified for AUX since they may appear in both [AUX +] and [AUX –] constructions. Non-auxiliary verbs are [AUX –], see Abeillé (2020), Chapter 12 of this volume.

[14] A list can be analysed as a type of feature structure with the features FIRST and REST, where the value of FIRST is the first element of the list.

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:[15]

(10) X ⇒ Y

Commonly X is a type and Y a feature description, and this is the case in all the constraints that we discuss below. However, X may also be a feature description with or without an associated type. This is necessary, for example, in the constraints that constitute binding theory. See Müller and Branco, this volume, chapter 20. Here is a very simple constraint:

(11) *phrase* ⇒ [COMPS ⟨⟩]

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.[16] As we will see below, most constraints are more complex than (11) and impose a number of restrictions on certain objects. For this reason, one might speak of a set of constraints. However, we will continue to use the term constraint for objects of the form in (10), no matter how many restrictions are imposed. Particularly important are constraints dealing with internal structure of various types of phrase. We will consider some constraints of this kind in section 5.

add secref

In most HPSG work, some shortcuts are used to abbreviate a feature path, for example in (11), COMPS stands for SYNSEM|LOC|CAT|COMPS. We use this practice in the rest of the chapter and it is used throughout the Handbook.

## 4 The lexicon

As noted above, the type *lexical-sign*, its subtypes, and the constraints on them are central to the lexicon of a language and the words it licenses.[17] Lexical rules

[15] The double-shafted arrow (⇒) is used in constraints, and a single shafted arrow |→ in lexical rules.

[16] The constraint in (11) is plausible for English, but it is too strong for some languages, especially for languages with complex predicates or partial VPs (see Godard & Samvelian 2020, Chapter 11 of this volume), and also for SOV languages if they are analysed in terms of binary branching (see Müller 2020, Chapter 10 of this volume).

[17] Other types of constraint are relevant to the form of lexemes and words, e.g. constraints on *synsem* objects and constraints on PHON values. These are also relevant to the form of phrases.

are also important. Some of the earliest work in HPSG focused on the organization of the lexicon and the question of how lexical generalizations can be captured, and detailed proposals have been developed.[18]

### 4.1 Lexemes and words

In some frameworks, the lexicon contains not lexemes but morphemes, i.e. roots and affixes of various kinds. But most work in HPSG has assumed a realizational approach to morphology. Within this approach, there are no morphemes, just lexemes and the words that realize them, and affixes are just bits of phonology realizing certain morphosyntactic features (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, this volume, chapter 21 and Davis and Koenig, this volume, chapter 4 for discussion.

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 (12) as its HEAD value.

(12) $\begin{bmatrix} \textit{noun} \\ \textsc{case}\ \textit{nom} \end{bmatrix}$

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 (13).

(13) $\begin{bmatrix} \textit{verb} \\ \textsc{vform}\ \textit{inf} \end{bmatrix}$

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.

[18] The lexicon is more important in HPSG than in some other constructional approaches, e.g. that of Goldberg (1995; 2006). See Müller & Wechsler (2014) and Müller, this volume, chapter 33 for discussion.

We must now say more about combinatorial properties. In much HPSG work it is assumed that SUBJ and COMPS encode what might be regarded as superficial combinatorial information and more basic combinatorial information is encoded by a feature ARG(UMENT)-ST(RUCTURE).[19] Normally the value of ARG-ST of a word is the concatenation of the values of SUBJ and COMPS, using ⊕ for list concatenation. In other words, we normally have the following situation (notice the use of re-entrancy or structure-sharing):

(14) $$\begin{bmatrix} \text{SUBJ} & \boxed{1} \\ \text{COMPS} & \boxed{2} \\ \text{ARG-ST} & \boxed{1} \oplus \boxed{2} \end{bmatrix}$$

As noted earlier, it is generally assumed that the SUBJ list never has more than one member. The appropriate features for the word *read* in (1a) for example would include the following, where the tags identify not lists but list members:

(15) $$\begin{bmatrix} \text{SUBJ} & \left\langle \boxed{1} \right\rangle \\ \text{COMPS} & \left\langle \boxed{2} \right\rangle \\ \text{ARG-ST} & \left\langle \boxed{1}\ \text{NP}, \boxed{2}\ \text{NP} \right\rangle \end{bmatrix}$$

ref from hpsg-handbook-bib

Under some circumstances, however, we have something different. For example, it has been proposed, e.g. in Manning & Sag (1999: 65), 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 *czytałem* in (1b), repeated here as (16), has the features in (17).

(16) Czytałem książkę.
write.PST.1SG book.ACC
'I read a book.'

(17) $$\begin{bmatrix} \text{SUBJ} & \langle\rangle \\ \text{COMPS} & \left\langle \boxed{1} \right\rangle \\ \text{ARG-ST} & \left\langle \text{NP}, \boxed{1}\text{NP} \right\rangle \end{bmatrix}$$

A similar analysis is widely assumed for unbounded dependency gaps. On this analysis, the verb *say* in (2), repeated here as (18), has the features in (19).

(18) What did you say?

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

(19) $$\begin{bmatrix} \textsc{subj} & \langle \boxed{1}\ \text{NP} \rangle \\ \textsc{comps} & \langle\rangle \\ \textsc{arg-st} & \langle \boxed{1}\ \text{NP, NP} \rangle \end{bmatrix}$$

It is also assumed that the arguments that are realised as pronominal affixes (traditionally known as clitics in Romance languages) are absent from COMPS lists (Miller & Sag 1997; Monachesi 2005), and other differences between SUBJ and COMPS and ARG-ST have been proposed for other languages (see Manning & Sag 1999, Wechsler, Koenig and Davies, this volume, chapter 9 for discussion. In much work, the relation ARG-ST and SUBJ and COMPS is regulated by a constraint called the Argument Realisation Principle (ARP). The following is a simplified version of the constraint proposed in Ginzburg & Sag (2000: 171) (see also Bouma et al. 2001: 12):

Monachesi from hpsg-hb.bib

from hpsg-bib

(20) $$\textit{word} \Rightarrow \begin{bmatrix} \textsc{subj} & \boxed{1} \\ \textsc{comps} & \boxed{2} \ominus \textit{list(non-canonical)} \\ \textsc{arg-st} & \boxed{1} \oplus \boxed{2} \end{bmatrix}$$

This ensures that non-canonical arguments, including gaps and arguments realized as a clitics do not appear in COMPS lists.[20] Notice, however, that it says nothing special about subjects and so does not allow an AVM like (17).[21] There are complex issues here, and the Principle will probably take a different form in different languages. So we will not try decide exactly what form it should take.

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 feature. Where it replaces SUBJ, the idea is that subjects are one of a number of types of specifiers, others being determiners within NPs and degree words like *so* and *too* within APs (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. Predicative nominals (e.g. *my cousin* in *Paul is my cousin*) may need both (Pollard & Sag 1994: 9.4.1; Ginzburg & Sag 2000: 409Abeillé & 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

[20] The sign ⊕ means concatenation of lists. In A ⊖ B=C, ⊖ stands for 'a relation where C is equal to A, iff B is the empty list. Otherwise C is the list that deletes the first part in A that is identical to B' (Müller 2001: 258).

[21] Ginzburg & Sag (2000: 177–183) explicitly allow gaps in SUBJ lists, but this is controversial, as discussed in Borsley and Crysmann, this volume, chapter 13.

feature called VALENCE instead of SUBJ, SPR and COMPS.[22] 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 can have the same combinatorial properties. Much HPSG work captures this fact by proposing that the type *lexeme* be cross-classified along two dimensions, one dealing with part-of-speech information and one dealing with argument selection information (Flickinger 1987). Figure 4 is a simple illustration based on Ginzburg & Sag (2000: 20).

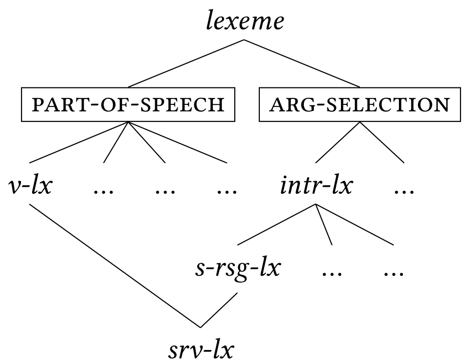


Figure 4: Cross-classification of lexemes

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, based on that in Ginzburg & Sag (2000: 22):

(21) $v\text{-}lx \Rightarrow \begin{bmatrix} \textsc{head} & verb \\ \textsc{arg-st} & \langle \text{XP} \ldots \rangle \end{bmatrix}$

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

[22] SBCG also has a feature X-ARG, which picks out subjects and other external arguments. But unlike the other features mentioned here, this always has the same value in a head and its mother. Its role is to make information about external arguments available outside the phrases in which they appear. See Sag (2007; 2012: 84, 149–151).

(22) $\textit{s-rsg-lx} \Rightarrow \left[\textsc{arg-st}\ \left\langle \boxed{1}, \left[\textsc{subj}\ \left\langle \boxed{1} \right\rangle\right] \ldots \right\rangle\right]$

This says that a subject-raising-lexeme has (at least) two (syntactic) arguments, a subject and a complement, and that the subject is whatever the complement requires as a subject, indicated by $\boxed{1}$. Most of the properties of any lexeme will be inherited from its supertypes. Thus, very little information needs to be listed for each specific lexeme, and the richness of the lexical description comes from the classification in a system like this.

For example, for a subject-raising verb like *seem*, its CAT and CONTENT features are the following, using Minimal Recursion semantics: RELS is the attribute for the list of elementary predications associated with a word, a lexeme or a phrase, SOA is for state-of-affair (see Koenig and Richter, this volume, chapter 22). *Seem* takes an infinitival VP complement.[23] Notice that no semantic role is assigned to the first argument (see Abeillé, this volume, chapter 12).

(23) $$\textit{seem-lx} \Rightarrow \textit{s-rsg-lx}\ \& \begin{bmatrix} \textsc{cat} & \begin{bmatrix} \textsc{arg-st} & \left\langle [\,], \mathrm{VP} \begin{bmatrix} \textsc{head} & \begin{bmatrix}\textsc{vform}\ \textit{inf}\end{bmatrix} \\ \textsc{index} & \textit{s1} \end{bmatrix} \right\rangle \end{bmatrix} \\ \textsc{cont} & \begin{bmatrix} \textsc{index} & \textit{s} \\ \textsc{rels} & \left\langle \begin{bmatrix} \textit{seems-rel} \\ \textsc{soa}\ \textit{s1} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

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

(24) $$\textit{seem-lx} \Rightarrow \begin{bmatrix} \textsc{cat} & \begin{bmatrix} \textsc{arg-st} & \left\langle \boxed{1}, \boxed{2}\ \mathrm{VP} \begin{bmatrix} \textsc{head} & \begin{bmatrix}\textsc{vform}\ \textit{inf}\end{bmatrix} \\ \textsc{subj} & \left\langle \boxed{1} \right\rangle \\ \textsc{index} & \textit{s1} \end{bmatrix} \right\rangle \\ \textsc{subj} & \left\langle \boxed{1} \right\rangle \\ \textsc{comps} & \left\langle \boxed{2} \right\rangle \end{bmatrix} \\ \textsc{cont} & \begin{bmatrix} \textsc{index} & \textit{s} \\ \textsc{rels} & \left\langle \begin{bmatrix} \textit{seems-rel} \\ \textsc{soa}\ \textit{s1} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

Notice that the SUBJ feature is underspecified. Thus, *seem* combines with an infinitival complement and with any subject (nominal or verbal, expletive or ref-

[23] The entry can be modified to allow predicative complements, as well as a second *to* complement (*John seems tired/ in a good mood to me*).

erential), provided this subject is appropriate for its complement (see Abeillé, this volume, chapter 12):

(25) a. John /*Working is sleeping.
b. John /* Working seems to be sleeping.
c. Working is/ seems to be tiring
d. It is raining / seems to be raining.

## 4.2 Lexical rules

The hierarchy of lexical types provides one way of capturing lexical generalizations. Lexical rules provide another.[24] They are used in morphology to relate lexemes to words (inflection) and lexemes to lexemes (derivation) (see Crysmann, this volume, chapter 21). For syntax, they are relevant especially to valence alternations such as that illustrated in the following (see Wechsler, Koenig and Davis, this volume, chapter 9):

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

(27) 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 (Pollard & Sag 1994: 150). The lexemes required for the latter use can be derived from the lexemes required for the former use by a lexical rule of the following form:[25]

[24] Lexical rules can be seen as a generative device, or alternatively, as a set of well-formedness conditions on the lexicon: if the lexicon contains items with description x, it must also contain items with description y (Meurers 2001). See Davis and Koenig, this volume chapter 4.

[25] Another representation of lexical rules is an AVM with features INPUT and OUTPUT, or with the left hand side as a daughter. As for (27), assuming both clauses and VPs have a verbal head, it easily extends to infinitival subjects, to accommodate pairs of examples like the following:

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

Clauses introduced by *that* are sometimes considered as CPs in HPSG (see section 7),
add secref

(28) $\left[\textsc{arg-st}\ \langle \mathrm{S} \rangle \oplus \boxed{2}\right] \mapsto \left[\textsc{arg-st}\ \langle \mathrm{NP}[\textit{it}] \rangle \oplus \boxed{2} \oplus \langle \mathrm{S} \rangle\right]$

The active-passive relation can be captured by a similar lexical rule (Flickinger 1987). Since these rules do not change the CONTENT feature, these alternations will preserve the meaning of the verb or adjective lexeme (see Davis and Koenig, this volume, chapter 4). Thus, the sentences in (27) will have a different syntactic structure from their counterparts in (26) but may have the same semantic representation (they will probably have different information structures, thus different CONTEXT features); see de Kuthy, this volume, chapter 23.

# 5 Syntax

As noted above, the type *phrase*, its subtypes, and the constraints on them are at the heart of the syntax of a language.[26] A simple hierarchy of phrase types was assumed in early HPSG, but what we have called Construction-based HPSG employs complex hierarchies of phrase types comparable to the complex hierarchies of lexical types employed in the lexicon.

## 5.1 A hierarchy of phrase types

Like much other work in syntax, HPSG takes from X-bar theory (Jackendoff 1977) the idea that the local trees that make up syntactic structures fall into a limited number of types. Like Jackendoff (1977), and unlike Minimalism, HPSG assumes that not all phrases are headed, even if many are, and does not limit the term 'head' to lexical elements. 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, combinations of a head and its complements. These can be headed by various parts of speech, verbs, prepositions, 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.

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

with verbs and complementizers as two subtypes of *verbal*.

[26] As noted in fn.18, constraints on *synsem* objects and PHON values are relevant to phrases as they are to lexemes and words.

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

(30) a. I'm wondering [who I talked to].

b. This is the official [who I talked to].

All this suggests the simple type hierarchy in Figure 5.

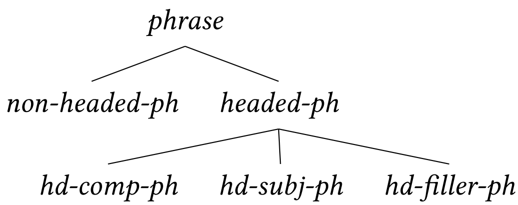


Figure 5: A hierarchy of types of phrases

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. But what features? One view is that the main features that are shared are those that are the value of HEAD. This is embodied in the following constraint, which is known as the Head Feature Principle:

(31) $$\textit{headed-ph} \Rightarrow \begin{bmatrix} \text{HEAD} & \boxed{1} \\ \text{HEAD-DTR} & \begin{bmatrix} \text{HEAD } \boxed{1} \end{bmatrix} \end{bmatrix}$$

Each of the three subtypes of *headed-ph* is subject to a constraint embodying its distinctive properties. Here is a constraint on the type *hd-comp-ph* (with SYNSEM abbreviated as SS):

(32) $$\textit{hd-comp-ph} \Rightarrow \begin{bmatrix} \text{HD-DTR } \boxed{1} & \begin{bmatrix} \textit{word} \\ \text{COMPS } \langle \boxed{2}, \ldots, \boxed{n} \rangle \end{bmatrix} \\ \text{DTRS} & \langle \boxed{1}, \begin{bmatrix}\text{SS } \boxed{2}\end{bmatrix}, \ldots, \begin{bmatrix}\text{SS } \boxed{n}\end{bmatrix} \rangle \end{bmatrix}$$

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.[27] Notice that nothing is said about the SYNSEM value of the phrase. It will be [COMPS <>], as required by the constraint in (14), and it will have the same value for HEAD as the Head-daughter as a consequence of the Head-Feature-Principle. It must also have the same value for SUBJ as the head daughter. One might add this to the constraint in (31), but that would miss a generalization. Head-complement-phrases are not the only phrases which have the same value for SUBJ as their head. This is also a feature of head-filler-phrases, as we will see below. It seems in fact that it is normal for a phrase to have the same value for any Valence feature as its head. This is often attributed to the Valence Principle, which can be stated informally as follows (cf. Sag & Wasow 1999: 86):

(33) Unless some constraint says otherwise, the mother's values for the Valence features are identical to those of the Head daughter.

There is no assumption in HPSG that all branching is binary.

Stefan: This has nothing to do with implementation. The implementation is not the reason for binary branching. There are theoretical reasons.

[28] Hence, where a head takes two complements, both may be its sisters. An example of the sort of structures that the analysis licenses is illustrated in Figure 6.

Instead of the Head Feature Principle and the Valence Principle, Ginzburg & Sag (2000: 33) propose the Generalized Head Feature Principle, which takes the following form:

(34) $$\textit{headed-ph} \Rightarrow \begin{bmatrix} \text{SYNSEM} & /\boxed{1} \\ \text{HD-DTR} & \begin{bmatrix} \text{SYNSEM} & /\boxed{1} \end{bmatrix} \end{bmatrix}$$

The slashes (/) here indicate that this is a default constraint (Lascarides & Copestake 1999). Thus, it says that a headed-phrase and its head daughter have the same SYNSEM value unless some other constraint requires something different. In versions of HPSG which assume this constraint, it is responsible among many other things for the fact that a head-complement-phrase has the same value for SUBJ as the head daughter.

We turn now to the type *hd-subj-ph*. Here we need a constraint which mentions the SYNSEM value of the phrase, and not just the daughters, as follows:

[27] The head could be identified as a [LEX +], [LIGHT +], or [WEIGHT *light*] phrase, to accomodate coordination of heads as in *John [knows and likes] this record.* (Abeillé 2006).

[28] Implemented HPSG grammars for various languages may assume binary branching structures (Flickinger 2000; Copestake 2002; Müller 2015), see Bender & Emerson (2020), Chapter 25 of this volume.

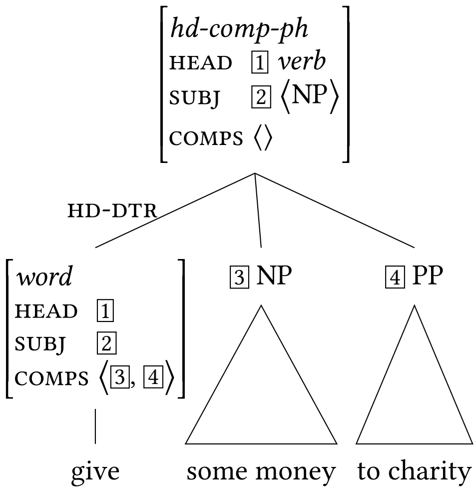


Figure 6: A tree for a head-complement phrase

(35) $$\textit{hd-subj-ph} \Rightarrow \begin{bmatrix} \textsc{subj} & \langle\rangle \\ \textsc{hd-dtr} & \boxed{1}\begin{bmatrix} \textsc{comps} & \langle\rangle \\ \textsc{subj} & \langle\boxed{2}\rangle \end{bmatrix} \\ \textsc{dtrs} & \left\langle \begin{bmatrix}\textsc{synsem}\ \boxed{2}\end{bmatrix}, \boxed{1} \right\rangle \end{bmatrix}$$

This ensures that a head-subject-phrase is [SUBJ <>] and has a head daughter which is [COMPS <>], and a non-head daughter with the *synsem* properties that appear in the head's SUBJ list.[29] It licenses structures like that in Figure 7.

Finally, we consider the type *hd-fill-ph*. This involves the feature SLASH, one of the features contained in the value of the feature NONLOCAL introduced earlier in (9). Its value is a set of local feature structures and it encodes information about unbounded dependency gaps (see Borsley & Crysmann, this volume, chapter 13). Here is the relevant constraint:[30]

[29] Instead of requiring the head to be [COMPS <>], one might require it to be a phrase (which would be required by (11) to be [COMPS <>]). However, this would require e.g. *laughed* in *Kim laughed* to be analysed as a phrase consisting of a single word. With (34) it can be analysed as just a word.

[30] We use ∪ for set union. 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?*).

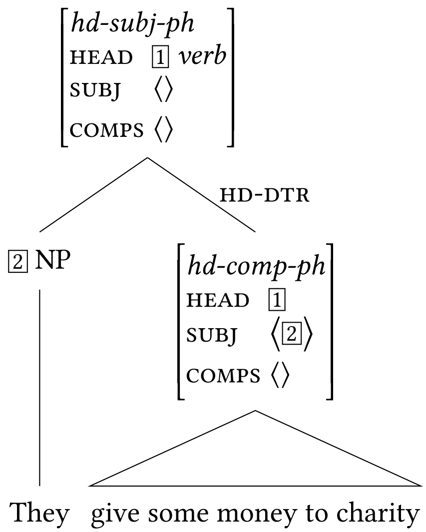


Figure 7: A tree for a head-subject phrase

(36) $$\textit{hd-filler-ph} \Rightarrow \begin{bmatrix} \text{SLASH} & \boxed{1} \\ \text{HD-DTR} & \boxed{2}\begin{bmatrix}\text{COMPS} & \langle\rangle \\ \text{SLASH} & \{\boxed{3}\} \cup \boxed{1}\end{bmatrix} \\ \text{DTRS} & \left\langle \left[\text{LOCAL}\ \boxed{3}\right], \boxed{2} \right\rangle \end{bmatrix}$$

This says that a head-filler-phrase has a head daughter, 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. $\boxed{1}$ is normally the empty set.[31] Figure 8 illustrates a typical head-filler-phrase.

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:

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

[31] As with (34), one might substitute phrase here for [COMPS <>]. But this would mean that *to* in *I would do it but I don't know how to* must be analysed as a phrase containing a single word. With (36) it can be just a word.

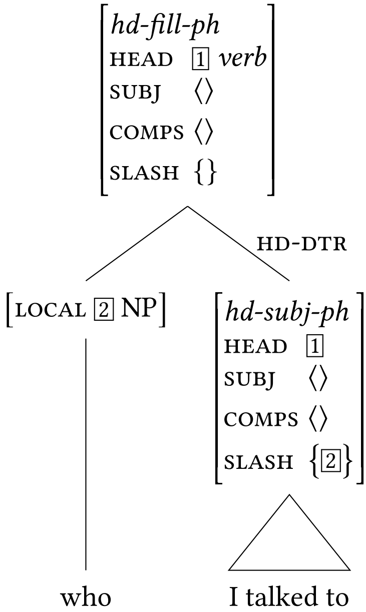


Figure 8: A tree for a head-filler phrase

Either the Valence Principle or the Generalized Head Feature Principle will ensure that a head-filler-phrase has the same value for SUBJ as its head daughter.

The constraints that we have just discussed are rather like phrase structure rules. This led Ginzburg & Sag (2000: 33) 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 (38a) one has (38b).

(38) a. *phrase* $\Rightarrow \begin{bmatrix} \text{SYNSEM} & \text{X} \\ \text{DTRS} & \langle \boxed{1}\ \text{Y, Z} \rangle \\ \text{HD-DTR} & \boxed{1} \end{bmatrix}$

b. *phrase:*
X → **H**[Y], Z

Notice that while the double arrow in (38a) has the normal 'if-then' interpretation, the single arrow in (38b) means 'consists of'. In some circumstances this informal notation may be more convenient 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 coindexing principle (the INDEX of a headed-phrase is the INDEX of its HEAD-DTR) and a 'compositionality' principle (the RELS of a phrase is the contatenation of the RELS of its DTRS) (Copestake et al. 2005; Koenig and Richter, this volume, chapter 6.1). This is supposed to be 36, right?

The type hierarchy in (34) is simplified in a number of respects. It includes no non-headed-phrases.[32] 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* can be cross-classified along two dimensions, one dealing with head-dependent relations and the other dealing with the properties of various types of clauses. A simplified illustration is given in Figure 9.

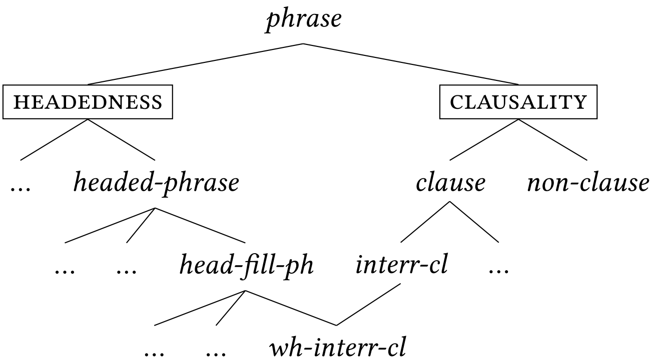


Figure 9: Cross-classification of phrases

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 (??) and certain properties characteristic of interrogative clauses, most obviously interrogative semantics. 42 isnt a constraint tho

[32] The most important type of non-headed phrase is coordinate structure. See Abeillé and Chaves, Chapter 16 of this volume, for discussion.

### 5.2 Constituency and constituent order

We must now say something about constituent 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.[33] 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, as in (39), or their order may be left unspecified for 'free' word order':

(39) $$\begin{bmatrix}\textsc{phon} & \boxed{1} \oplus \boxed{2} \\ \textsc{dtrs} & \left\langle \begin{bmatrix}\textsc{phon}\ \boxed{1}\end{bmatrix}, \begin{bmatrix}\textsc{phon}\ \boxed{2}\end{bmatrix} \right\rangle\end{bmatrix} \begin{bmatrix}\textsc{phon} & \boxed{2} \oplus \boxed{1} \\ \textsc{dtrs} & \left\langle \begin{bmatrix}\textsc{phon}\ \boxed{1}\end{bmatrix}, \begin{bmatrix}\textsc{phon}\ \boxed{2}\end{bmatrix} \right\rangle\end{bmatrix}$$

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

(40) a. black sheep

b. defaid du
sheep.PL black
'black sheep'

## Abbreviations

| | | | |
|---|---|---|---|
| COP | copula | NEG | negation |
| FV | final vowel | SM | subject marker |


## Acknowledgements

We are grateful to Stefan Müller, Jean-Pierre Koenig, and Frank Richter for many helpful comments on earlier versions of this chapter. We alone are responsible for what appears here.


[33] As discussed in section 7.1,
add secref
in some HPSG work, linear order is a property of so-called order domains, which essentially mediate constituent structure and phonology (see Müller, this volume, chapter 10).

## References


Abeillé, Anne. 2006. In defense of lexical coordination. In Olivier Bonami & Patricia Cabredo Hofherr (eds.), *Empirical issues in formal syntax and semantics*, vol. 6, 7–36. Paris: CNRS. http://www.cssp.cnrs.fr/eiss6/ (25 February, 2018).

Abeillé, Anne. 2020. Control and Raising. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 453–497. Berlin: Language Science Press. DOI: ??.

Abeillé, Anne & Robert D. Borsley. 2008. Comparative correlatives and parameters. *Lingua* 118(8). 1139–1157. DOI: 10.1016/j.lingua.2008.02.001.

Abeillé, Anne & Rui P. Chaves. 2020. Coordination. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 675–726. Berlin: Language Science Press. DOI: ??.

Abeillé, Anne & Danièle Godard. 2003. The syntactic flexibility of adverbs: French degree adverbs. In Stefan Müller (ed.), *Proceedings of the 10th International Conference on Head-Driven Phrase Structure Grammar, Michigan State University, East Lansing*, 26–46. Stanford, CA: CSLI Publications. http://csli-publications.stanford.edu/HPSG/2003/ (25 February, 2018).

Anderson, Stephen R. 1992. *A-morphous morphology* (Cambridge Studies in Linguistics 62). Cambridge: Cambridge University Press.

Barwise, Jon & John Perry. 1983. *Situations and attitudes*. Cambridge, MA: MIT Press.

Bender, Emily M. 2016. Linguistic typology in natural language processing. *Linguistic Typology* 20(3). 645–660. DOI: 10.1515/lingty-2016-0035.

Bender, Emily M., Scott Drellishak, Antske Fokkens, Laurie Poulson & Safiyyah Saleem. 2010. Grammar customization. *Research on Language & Computation* 8(1). 10.1007/s11168-010-9070-1, 23–72. http://dx.doi.org/10.1007/s11168-010-9070-1.

Bender, Emily M. & Guy Emerson. 2020. Computational linguistics and grammar engineering. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 1039–1083. Berlin: Language Science Press. DOI: ??.

Borsley, Robert D. 2009. On the superficiality of Welsh agreement. *Natural Language & Linguistic Theory* 27(2). 225–265. DOI: 10.1007/s11049-009-9067-3.

Borsley, Robert D. 2011. Constructions, functional heads, and comparative correlatives. In Olivier Bonami & Patricia Cabredo Hofherr (eds.), *Empirical issues in* *syntax and semantics*, vol. 8, 7–20. Paris: CNRS. http://www.cssp.cnrs.fr/eiss8/borsley-eiss8.pdf (23 August, 2018).

Borsley, Robert D. & Berthold Crysmann. 2020. Unbounded dependencies. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 499–553. Berlin: Language Science Press. DOI: ??.

Borsley, Robert D. & Stefan Müller. 2020. HPSG and Minimalism. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 1185–1257. Berlin: Language Science Press. DOI: ??.

Bouma, Gosse, Robert Malouf & Ivan A. Sag. 2001. Satisfying constraints on extraction and adjunction. *Natural Language & Linguistic Theory* 19(1). 1–65. DOI: 10.1023/A:1006473306778.

Bresnan, Joan (ed.). 1982. *The mental representation of grammatical relations* (MIT Press Series on Cognitive Theory and Mental Representation). Cambridge, MA: MIT Press.

Chomsky, Noam. 1957. *Syntactic structures* (Janua Linguarum / Series Minor 4). The Hague: Mouton.

Copestake, Ann. 2002. *Implementing typed feature structure grammars* (CSLI Lecture Notes 110). Stanford, CA: CSLI Publications.

Copestake, Ann, Daniel P. Flickinger, Carl J. Pollard & Ivan A. Sag. 2005. Minimal Recursion Semantics: An introduction. *Research on Language and Computation* 3(2–3). 281–332. DOI: 10.1007/s11168-006-6327-9.

Crysmann, Berthold. 2020. Morphology. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 883–934. Berlin: Language Science Press. DOI: ??.

De Kuthy, Kordula. 2020. Information structure. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 977–1013. Berlin: Language Science Press. DOI: ??.

Flickinger, Dan. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering* 6(1). Special Issue on Efficient Processing with HPSG: Methods, Systems, Evaluation, 15–28. DOI: 10.1017/S1351324900002370.

Flickinger, Dan, Carl Pollard & Tom Wasow. 2020. The evolution of HPSG. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 35–71. Berlin: Language Science Press. DOI: ??.

Flickinger, Daniel P. 1987. *Lexical rules in the hierarchical lexicon*. Stanford University. (Doctoral dissertation).

Flickinger, Daniel P., Carl J. Pollard & Thomas Wasow. 1985. Structure-sharing in lexical representation. In William C. Mann (ed.), *Proceedings of the Twenty-Third Annual Meeting of the Association for Computational Linguistics*, 262–267. Chicago, IL. DOI: 10.3115/981210.981242.

Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum & Ivan A. Sag. 1985. *Generalized Phrase Structure Grammar*. Cambridge, MA: Harvard University Press.

Ginzburg, Jonathan & Ivan A. Sag. 2000. *Interrogative investigations: The form, meaning, and use of English interrogatives* (CSLI Lecture Notes 123). Stanford, CA: CSLI Publications.

Godard, Danièle & Pollet Samvelian. 2020. Complex predicates. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 385–451. Berlin: Language Science Press. DOI: ??.

Goldberg, Adele E. 1995. *Constructions: A Construction Grammar approach to argument structure* (Cognitive Theory of Language and Culture). Chicago: The University of Chicago Press.

Goldberg, Adele E. 2006. *Constructions at work: The nature of generalization in language* (Oxford Linguistics). Oxford: Oxford University Press.

Höhle, Tilman N. 1999. An architecture for phonology. In Robert D. Borsley & Adam Przepiórkowski (eds.), *Slavic in Head-Driven Phrase Structure Grammar*, 61–90. Republished as **HoehlePhonology**. Stanford, CA: CSLI Publications.

Jackendoff, Ray S. 1977. *$\overline{X}$ syntax: A study of phrase structure*. Cambridge, MA: MIT Press.

Kay, Paul & Charles J. Fillmore. 1999. Grammatical constructions and linguistic generalizations: The What's X Doing Y? Construction. *Language* 75(1). 1–33.

Kay, Paul & Laura A. Michaelis. 2017. A few words to do with multiword expressions. In Cleo Condoravdi & Tracy Holloway King (eds.), *Tokens of meaning: Papers in honor of Lauri Karttunen*, 87–118. Stanford, CA: CSLI Publications.

Koenig, Jean-Pierre. 1999. *Lexical relations* (Stanford Monographs in Linguistics). Stanford, CA: CSLI Publications.

Koenig, Jean-Pierre & Frank Richter. 2020. Semantics. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 935–975. Berlin: Language Science Press. DOI: ??.

Lascarides, Alex & Ann Copestake. 1999. Default representation in constraint-based frameworks. *Computational Linguistics* 25(1). 55–105.

Malouf, Robert. 2003. Cooperating constructions. In Elaine J. Francis & Laura A. Michaelis (eds.), *Mismatch: form-function incongruity and the architecture of grammar* (CSLI Lecture Notes 163), 403–424. Stanford, CA: CSLI Publications.

Manning, Christopher & Ivan Sag. 1999. Dissociations between argument structure and grammatical relations. In Gert Webelhuth, Jean-Pierre Koenig & Andreas Kathol (eds.), *Lexical and constructional aspects of linguistic explanation*. Stanford, CA: CSLI Publications.

Meurers, Walt Detmar. 2001. On expressing lexical generalizations in HPSG. *Nordic Journal of Linguistics* 24(2). 161–217.

Michaelis, Laura A. & Knud Lambrecht. 1996. Toward a construction-based theory of language function: The case of nominal extraposition. *Language* 72(2). 215–247. DOI: 10.2307/416650.

Miller, Philip H. & Ivan A. Sag. 1997. French clitic movement without clitics or movement. *Natural Language & Linguistic Theory* 15(3). 573–639.

Monachesi, Paola. 2005. *The verbal complex in Romance: a case study in grammatical interfaces*. Oxford: Oxford University Press.

Müller, Stefan. 1996. The Babel-System: An HPSG fragment for German, a parser, and a dialogue component. In *Proceedings of the Fourth International Conference on the Practical Application of Prolog*, 263–277. London.

Müller, Stefan. 2001. The passive as a lexical rule. In Dan Flickinger & Andreas Kathol (eds.), *The proceedings of the 7th International Conference on Head-Driven Phrase Structure Grammar: University of California, Berkeley, 22–23 July, 2000*, 247–266. CSLI Publications. http://csli-publications.stanford.edu/HPSG/1/ (25 February, 2018).

Müller, Stefan. 2014. Elliptical constructions, multiple frontings, and surface-based syntax. In Paola Monachesi, Gerhard Jäger, Gerald Penn & Shuly Wintner (eds.), *Proceedings of Formal Grammar 2004, Nancy*, 91–109. Stanford, CA: CSLI Publications.

Müller, Stefan. 2015. The CoreGram project: Theoretical linguistics, theory development and verification. *Journal of Language Modelling* 3(1). 21–86. DOI: 10.15398/jlm.v3i1.91.

Müller, Stefan. 2016. *Grammatical theory: From Transformational Grammar to constraint-based approaches* (Textbooks in Language Sciences 1). Berlin: Language Science Press. DOI: 10.17169/langsci.b25.167.

Müller, Stefan. 2020. Constituent order. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 339–383. Berlin: Language Science Press. DOI: ??.

Müller, Stefan & Walter Kasper. 2000. HPSG analysis of German. In Wolfgang Wahlster (ed.), *Verbmobil: Foundations of speech-to-speech translation* (Artificial Intelligence), 238–253. Berlin: Springer Verlag.

Müller, Stefan & Stephen Wechsler. 2014. Lexical approaches to argument structure. *Theoretical Linguistics* 40(1–2). 1–76. DOI: 10.1515/tl-2014-0001.

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, Carl & Ivan A. Sag. 1987. *Information-based syntax and semantics* (CSLI Lecture Notes 13). Stanford, CA: CSLI Publications.

Pollard, Carl & Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar* (Studies in Contemporary Linguistics). Chicago: The University of Chicago Press.

Postal, Paul M. 2003. (Virtually) conceptually necessary. *Journal of Linguistics* 39(3). 599–620. DOI: 10.1017/S0022226703002111.

Pullum, Geoffrey K. & Barbara C. Scholz. 2001. On the distinction between model-theoretic and generative-enumerative syntactic frameworks. In P. de Groote, G. Morrill & C. Retoré (eds.), *Logical aspects of computational linguistics: 4th international conference* (Lecture Notes in Artificial Intelligence 2099), 17–43. Springer.

Richter, Frank. 2020. Formal background. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 73–108. Berlin: Language Science Press. DOI: ??.

Richter, Frank & Manfred Sailer. 2004. Polish negation and lexical resource semantics. In Lawrence S. Moss & Richard T. Oehrle (eds.), *Proceedings of the joint meeting of the 6th conference on formal grammar and the 7th conference on mathematics of language* (Electronic Notes in Theoretical Computer Science 53), 309–321. Elsevier Science.

Richter, Frank & Manfred Sailer. 2009. Phraseological clauses as Constructions in HPSG. In Stefan Müller (ed.), *Proceedings of the 16th International Conference on Head-Driven Phrase Structure Grammar, University of Göttingen, Germany*, 297–317. Stanford, CA: CSLI Publications. http://csli-publications.stanford.edu/HPSG/2009/ (25 February, 2018).

Sag, Ivan A. 1997. English relative clause constructions. *Journal of Linguistics* 33(2). 431–484. DOI: 10.1017/S002222679700652X.

Sag, Ivan A. 2007. Remarks on locality. In Stefan Müller (ed.), *Proceedings of the 14th International Conference on Head-Driven Phrase Structure Grammar*, 394–414. Stanford, CA: CSLI Publications. http://csli-publications.stanford.edu/HPSG/2007/ (25 February, 2018).

Sag, Ivan A. 2010. English filler-gap constructions. *Language* 86(3). 486–545.

Sag, Ivan A. 2012. Sign-Based Construction Grammar: An informal synopsis. In Hans C. Boas & Ivan A. Sag (eds.), *Sign-Based Construction Grammar* (CSLI Lecture Notes 193), 69–202. Stanford, CA: CSLI Publications.

Sag, Ivan A., Rui Chaves, Anne Abeillé, Bruno Estigarribia, Frank Van Eynde, Dan Flickinger, Paul Kay, Laura A. Michaelis-Cummings, Stefan Müller, Geoffrey K. Pullum & Tom Wasow. 2020. Lessons from the English auxiliary system. *Journal of Linguistics* 56(1). 87–155. DOI: 10.1017/S002222671800052X.

Sag, Ivan A. & Janet Dean Fodor. 1994. Extraction without traces. In Raul Aranovich, William Byrne, Susanne Preuss & Martha Senturia (eds.), *Proceedings of the Thirteenth West Coast Conference on Formal Linguistics*, 365–384. Stanford, CA: CSLI Publications/SLA.

Sag, Ivan A. & Thomas Wasow. 1999. *Syntactic theory: A formal introduction*. Stanford, CA: CSLI Publications.

Sag, Ivan A. & Thomas Wasow. 2011. Performance-compatible competence grammar. In Robert D. Borsley & Kersti Börjars (eds.), *Non-transformational syntax: Formal and explicit models of grammar: A guide to current models*, 359–377. Oxford, UK/Cambridge, MA: Blackwell Publishers Ltd.

Sag, Ivan A. & Thomas Wasow. 2015. Flexible processing and the design of grammar. *Journal of Psycholinguistic Research* 44(1). 47–63.

Sag, Ivan A., Thomas Wasow & Emily M. Bender. 2003. *Syntactic theory: A formal introduction*. 2nd edn. (CSLI Lecture Notes 152). Stanford, CA: CSLI Publications.

Sailer, Manfred. 2020. Idioms. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 727–759. Berlin: Language Science Press. DOI: ??.

Stump, Gregory T. 2001. *Inflectional morphology: A theory of paradigm structure* (Cambridge Studies in Linguistics 93). Cambridge University Press.

Wechsler, Stephen & Ash Asudeh. 2020. HPSG and Lexical Functional Grammar. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 1319–1369. Berlin: Language Science Press. DOI: ??.