Chapter 1

Basic properties and elements

Anne Abeillé
Université de Paris

Robert D. Borsley
University of Essex and Bangor University

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 makes a fundamental distinction between signs of various kinds and the constructions which license them.

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 Generalised Phrase Structure Grammar (GPSG) framework (Gazdar, Klein, Pullum & Sag 1985), but it was also influenced by Categorial Grammar (Ajdukiewicz 1935; Steedman 2000), and, as Pollard & Sag (1987: 1) emphasised, 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 chapter (Flickinger, Pollard

Anne Abeillé & Robert D. Borsley. 2021. Basic properties and elements. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 3–45. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599818

& Wasow 2021), 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, Pollard & Wasow 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 SBCG, 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 penultimate section.

The chapter is organised as follows. In Section 2, we set out the properties that characterise 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 summarise the chapter.

2 Properties

Perhaps the first thing to say about HPSG is that it is a form of Generative Grammar in the sense of Chomsky (1965: 4). 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 gram-

[1] As discussed in Richter (2021), 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.

mar, 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, and so on. 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 is 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), Sag & Wasow (2011; 2015), and Wasow (2021), Chapter 24 of this volume.

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 position at another level. 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 complementiser they all do, an empty one if not an overt one. Similarly, it is not assumed that because some languages like English have

[3] In most HPSG work, all constraints are equal. Hence, there is no possibility – as there is in Optimality Theory (Prince & Smolensky 2004) – of violating one if it is the only way to satisfy another more important one (Malouf 2003). However, see Müller & Kasper (2000) and Oepen et al. (2004) for an HPSG parser with probabilities or weighted constraints.

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 is 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 1995; Müller 2004; Borsley & Crysmann 2021: 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 HPSG 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 & Borsley 2008; Borsley 2011), and the *X's Way* construction (Sag 2012: Section 7.4). As we will see, HPSG is an approach which is able to accommodate broad linguistic generalisations, 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, HPSG does not try to reduce

[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.

[5]Idioms have also been an important focus of research in HPSG. See e.g. Sag (2007: Section 5.4), Richter & Sailer (2009), Kay & Michaelis (2017), and Sailer (2021), Chapter 17 of this volume.

either semantics or morphology to syntax (see Crysmann 2021, Chapter 21 of this volume on morphology in HPSG and Koenig & Richter 2021, Chapter 22 of this volume on semantics). 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 emphasises the importance of firm empirical foundations and detailed formal analyses of the kind advocated by Chomsky in *Syntactic Structures* (Chomsky 1957: 5). 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 fifty-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 2021, 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; Borsley & Müller 2021: Section 5.2, Chapter 28 of this volume). Every postulated element or property 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 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?

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

Clearly not. As Müller also states, "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" (Müller 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 the properties of their description and cannot be underspecified in any way.[7] 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 type hierarchy in Figure 1.

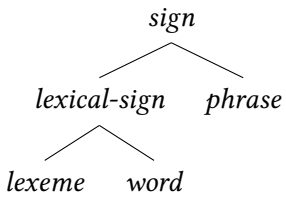

Figure 1: A hierarchy of types of signs

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 facts somewhere 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

[7] As pointed out by Pollard & Sag (1987: Chapter 2), HPSG grammars provide descriptions for models of linguistic objects rather than for linguistic objects per se. See also Richter (2021), Chapter 3 of this volume for a detailed discussion of the formal background of HPSG.

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*. (Features 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)
$$\begin{bmatrix} \textit{sign} & \\ \textsc{phon} & \textit{list}(\textit{phon}) \\ \textsc{synsem} & \textit{synsem} \end{bmatrix}$$

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

Here, following a widespread practice, we use standard orthography instead of real *phon* objects,[8] 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: 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.[9] Thus, phrases take the form in (6a), and headed phrases the form in (6b):

[8] See Bird & Klein (1994), Höhle (1999), and Walther (1999) for detailed approaches to phonology and structured PHON values, and De Kuthy (2021), Chapter 23 of this volume and Abeillé & Chaves (2021: 762–763), Chapter 16 of this volume for reference to structured PHON values.

[9] 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 HEAD-DTR can be a word or a phrase. Within Minimalism, the term *head* is only applied to words. On this usage, the value of HEAD-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 2021: Section 2, Chapter 16 of 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.

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

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

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

Here, the two instances of the tag $\boxed{1}$ indicate that the *sign* 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 (a member of a list or set), 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 Crysmann 2021, Chapter 21 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 Briscoe & Copestake 1999; Meurers 2001 and Section 4.2 below). Hence, the DTRS feature is relevant to words as well as phrases.

AVMs like (7) can be quite hard to look at. Hence, it is common to use traditional tree diagrams instead. Thus, we might have the tree-like representation in Figure 2 instead of (7). But one should bear in mind that AVMs correspond to (rooted) graphs and provide more detailed descriptions than traditional phrase structure trees, with richer node 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 phonology, semantics, and information structure.[10]

[10] This differs from Lexical Functional Grammar, for instance, which distributes the information between different kinds of structures (see Wechsler & Asudeh 2021, Chapter 30 of this volume).

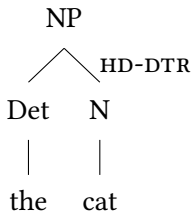

Figure 2: A simple tree for *the cat*

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. 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.[11] 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:

(8)
$$\begin{bmatrix} \textit{sign} \\ \textsc{phon} & \textit{list}(\textit{phon}) \\ \textsc{synsem} & \begin{bmatrix} \textit{synsem} \\ \textsc{local} & \begin{bmatrix} \textit{local} \\ \textsc{category} & \begin{bmatrix} \textit{category} \\ \textsc{head} & \textit{part-of-speech} \\ \textsc{subj} & \textit{list}(\textit{synsem}) \\ \textsc{comps} & \textit{list}(\textit{synsem}) \end{bmatrix} \\ \textsc{content} & \ldots \\ \textsc{context} & \ldots \end{bmatrix} \\ \textsc{nonlocal} & \ldots \end{bmatrix} \end{bmatrix}$$

[11] Words also have a MORPH (or INFL) attribute that we ignore here (see Crysmann 2021, Chapter 21 of this volume).

The type *part-of-speech* has subtypes such as *noun*, *verb*, and *adjective*. In other words, we have a type hierarchy of the form given in Figure 3.

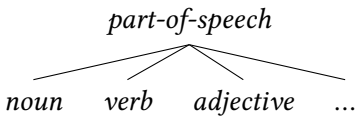

Figure 3: A hierarchy for part of speech

The type hierarchy in Figure 1 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.[12] We might have a *synsem* object of the following form for the phrase *the cat*:

(9)
$$\begin{bmatrix} \textit{synsem} & \\ \textsc{local} & \begin{bmatrix} \textit{local} & \\ \textsc{category} & \begin{bmatrix} \textit{category} & \\ \textsc{head} & \textit{noun} \\ \textsc{subj} & \langle\rangle \\ \textsc{comps} & \langle\rangle \end{bmatrix} \\ \textsc{content} & \ldots \\ \textsc{context} & \ldots \end{bmatrix} \\ \textsc{nonlocal} & \ldots \end{bmatrix}$$

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 *sg* (*singular*) and *pl* (*plural*).[13] There are also Boolean

[12] AVMs associated with types used to be combined by unification (Pollard & Sag 1987: Chapter 2). See Richter (2021: 90–91), Chapter 3 of this volume for discussion of the term "unification".

[13] In practice, a more complex system of values may well be appropriate (Flickinger 2000: Section 3).

features with + and − as their values. An example is AUX, used to distinguish auxiliary verbs ([AUX +]) from non-auxiliary verbs ([AUX −]).[14]

As the preceding discussion 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).[15] 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, Flickinger, Pollard & Sag 2005), while others use Lexical Resource Semantics (Richter & Sailer 2004). Sag (2010: 501) adopts a conventional, Montague-style possible-worlds semantics (Montague 1974) in his analysis of English filler-gap constructions, and SBCG (Section 7.2) has generally employed a version of Frame Semantics. See Koenig & Richter (2021), Chapter 22 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 2021, Chapter 23 of this volume).

We will say more about types and features in the following sections. We turn now to constraints. These are the machinery which imposes conditions on linguistic objects by saying that if an object has some property or properties, it must have some other property or properties. Constraints take the following form:[16]

(10) $X \Rightarrow Y$

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 2021a, Chapter 20 of this volume). Here is a very simple constraint:

[14] In some recent work, e.g. Sag (2012: 157–162) and 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 (2021: Section 4), Chapter 12 of this volume.

[15] A list can be represented as a feature description with the features FIRST and REST, where the value of FIRST is the first element of the list. See Richter (2021: 102), Chapter 3 of this volume for more on the encoding of lists.

[16] The double-shafted arrow ⇒ is used in implicational constraints, and a single shafted arrow ↦ in lexical rules.

(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.[17] 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 the internal structure of various types of phrases. We will consider some constraints of this kind in Section 5.

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.[18] Lexical rules are also important. Some of the earliest work in HPSG focused on the organisation of the lexicon and the question of how lexical generalisations can be captured, and detailed proposals have been developed.[19]

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 realise them, and affixes are just bits of phonology realising 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 realised by affixes.

[17] 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 2021, Chapter 11 of this volume), and also for SOV languages if they are analysed in terms of binary branching (see Müller 2021b, Chapter 10 of this volume).

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

[19] 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 (2021c: Section 2), Chapter 32 of this volume for discussion.

See Crysmann (2021: Section 3), Chapter 21 of this volume, Davis & Koenig (2021: Section 2), Chapter 4 of this volume, and Borsley & Müller (2021: Section 4.1.3), Chapter 28 of this volume 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 HEAD value 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* might 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 be regarded as superficial combinatorial information and that more basic combinatorial information is encoded by a feature ARG(UMENT)-ST(RUCTURE).[20] Normally the value of ARG-ST of a word is the concatenation of the values of SUBJ and COMPS, using $\oplus$ for list concatenation. In other words, we normally have the following situation (notice the use of re-entrancy or structure sharing):

(14) $\begin{bmatrix} \textsc{subj} & \boxed{1} \\ \textsc{comps} & \boxed{2} \\ \textsc{arg-st} & \boxed{1} \oplus \boxed{2} \end{bmatrix}$

[20] ARG-ST is also crucial for Binding Theory, which takes the form of a number of constraints on ARG-ST lists. See Müller (2021a), Chapter 20 of this volume.

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) Lexical item for *read*:
$$\begin{bmatrix} \textsc{subj} & \langle \boxed{1} \rangle \\ \textsc{comps} & \langle \boxed{2} \rangle \\ \textsc{arg-st} & \langle \boxed{1}\,\mathrm{NP}, \boxed{2}\,\mathrm{NP} \rangle \end{bmatrix}$$

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* 'read' in (1b), repeated here as (16), has the features in (17).

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

(17) Lexical item for *czytałem* 'read' with the subject dropped:
$$\begin{bmatrix} \textsc{subj} & \langle\rangle \\ \textsc{comps} & \langle \boxed{1} \rangle \\ \textsc{arg-st} & \langle \mathrm{NP}, \boxed{1}\,\mathrm{NP} \rangle \end{bmatrix}$$

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) Lexical item for *say* with the object extracted:
$$\begin{bmatrix} \textsc{subj} & \langle \boxed{1}\,\mathrm{NP} \rangle \\ \textsc{comps} & \langle\rangle \\ \textsc{arg-st} & \langle \boxed{1}\,\mathrm{NP}, \mathrm{NP} \rangle \end{bmatrix}$$

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: Section 3; Monachesi 2005), and other differences between SUBJ, COMPS, and ARG-ST have been proposed for other languages (see Manning & Sag 1999, Davis, Koenig & Wechsler 2021: Section 3, Chapter 9 of this volume for discussion). In much work, the relation between ARG-ST, 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):

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

This ensures that non-canonical arguments, including gaps and arguments realised as affixes, do not appear in COMPS lists.[21] Notice, however, that it says nothing special about subjects.[22] 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 (Pollard & Sag 1994: 358). Where it is an additional feature, the idea is that there are a 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: Section 9.4.1; Ginzburg & Sag 2000: 409; 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, SPR, and COMPS.[23] Obviously, there are some important issues here.

It is a central feature of lexical items that part of speech and combinatorial properties are separate matters. Members of the same part of speech can have

[21] As we saw above, the sign $\oplus$ means concatenation of lists. Ginzburg & Sag (2000: 170) state the following about $\ominus$: "Here '$\ominus$' designates a relation of contained list difference. If λ_2 is an ordering of a set σ_2 and λ_1 is a subordering of λ_2, then $\lambda_2 \ominus \lambda_1$ designates the list that results from removing all members of λ_1 from λ_2; if λ_1 is not a sublist of λ_2, then the contained list difference is not defined. For present purposes, $\ominus$ is interdefinable with the sequence union operator ($\bigcirc$) of Reape (1994) and Kathol (1995): $(A \ominus B = C) \Leftrightarrow (C \bigcirc B = A)$." The operator $\bigcirc$ is called *shuffle* and is also explained in Müller (2021b: 391), Chapter 10 of this volume.

[22] Ginzburg & Sag (2000: 177–183) explicitly allow gaps in SUBJ lists, but this is controversial, as discussed in Borsley & Crysmann (2021: 547–548), Chapter 13 of this volume.

[23] 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).

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: 20). Figure 4 is a simple illustration based on Ginzburg & Sag (2000: 20).

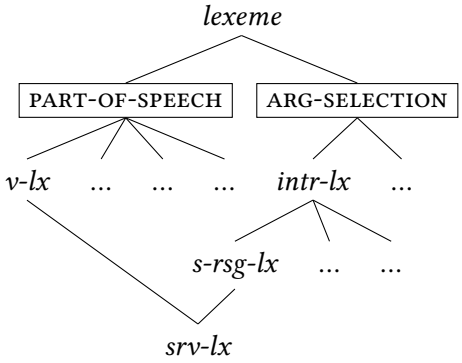

Figure 4: Cross-classification of lexemes

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

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

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 a simplified version of Minimal Recursion Semantics (MRS; Copestake et al. 2005): REL(ATION)s is the attribute for the list of elementary predications associated with a word, a lexeme, or a phrase, and SOA is for *state-of-affairs* (see Koenig & Richter 2021, Chapter 22 of this volume). *Seem* takes an infinitival VP complement.[24] Notice that the first syntactic argument (the subject) is not mentioned in the CONTENT, i.e. it is not assigned a semantic role by *seem* (see Abeillé 2021: Section 1, Chapter 12 of this volume).

(23) Constraints on type *seem-lx* in addition to those inherited from *srv-lx*:
seem-lx ⇒

$$\begin{bmatrix} \textsc{cat} & \begin{bmatrix} \textsc{arg-st} & \left\langle [], \text{VP}\begin{bmatrix} \textsc{head} & \begin{bmatrix} \textsc{vform} & \textit{inf} \end{bmatrix} \\ \textsc{index} & \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \\ \textsc{cont} & \begin{bmatrix} \textsc{index} & s \\ \textsc{rels} & \left\langle \begin{bmatrix} \textit{seem-rel} \\ \textsc{soa} & \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

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

(24) Constraints for the lexeme *seem*:

$$\begin{bmatrix} \textit{seem-lx} & \\ \textsc{cat} & \begin{bmatrix} \textsc{subj} & \langle \boxed{1} \rangle \\ \textsc{comps} & \langle \boxed{2} \rangle \\ \textsc{arg-st} & \left\langle \boxed{1}, \boxed{2}\ \text{VP}\begin{bmatrix} \textsc{head} & \begin{bmatrix} \textsc{vform} & \textit{inf} \end{bmatrix} \\ \textsc{subj} & \langle \boxed{1} \rangle \\ \textsc{index} & \boxed{3} \end{bmatrix} \right\rangle \end{bmatrix} \\ \textsc{cont} & \begin{bmatrix} \textsc{index} & s \\ \textsc{rels} & \left\langle \begin{bmatrix} \textit{seem-rel} \\ \textsc{soa} & \boxed{3} \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

Notice that the SUBJ value is underspecified. Thus, *seem* combines with an infinitival complement and with any subject (nominal or verbal, expletive or referential), provided this subject is appropriate for the infinitival complement (see Abeillé 2021: Section 2.1, Chapter 12 of this volume):

(25) a. Kim is/seems to be sleeping.
b. * Kim is/seems to be snowing.

[24] 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*).

c. That he is clever is/seems to be obvious.
d. * That he is clever is/seems to be obese.
e. There is/seems to be a problem.
f. * There is/seems to be in Paris.

4.2 Lexical rules

The hierarchy of lexical types provides one way of capturing lexical generalisations. Lexical rules provide another.[25] They are used in morphology to relate lexemes to words (inflection) and lexemes to lexemes (derivation) (see Crysmann 2021: Section 2, 3, Chapter 21 of this volume). For syntax, they are relevant especially to valence alternations such as that illustrated in the following (see Davis, Koenig & Wechsler 2021: Section 5.3, Chapter 9 of this volume):

(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:[26]

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

[25] 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 also Davis & Koenig (2021: Section 5), Chapter 4 of this volume.

[26] 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 that both clauses and VPs have a verbal head, it easily extends to infinitival subjects, to accommodate pairs of examples like the following:

(i) a. To annoy Lee is easy.
b. It is easy to annoy Lee.

Clauses introduced by *that* are sometimes considered as CPs in HPSG (see Section 7), with verbs and complementisers as two subtypes of *verbal*.

The active-passive relation can be captured by a similar lexical rule (Flickinger 1987: Section 5.1.1). Since these rules do not change the CONTENT feature, these alternations will preserve the meaning of the verb or adjective lexeme (see Davis & Koenig 2021, Chapter 4 of this volume). 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 (2021), Chapter 23 of this volume on information structure).

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.[27] A simple hierarchy of phrase types was assumed in early HPSG, but what we have called Constructional 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.

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.

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

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

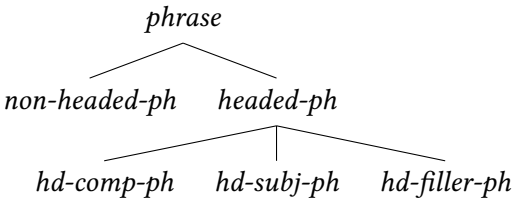

Figure 5: A hierarchy of types of phrases

Consider first the type *headed-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:[28]

(31) $$\textit{headed-ph} \Rightarrow \begin{bmatrix} \textsc{head} & \boxed{1} \\ \textsc{head-dtr} & \left[\textsc{head}\ \boxed{1}\right] \end{bmatrix}$$

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) $$\textit{hd-comp-ph} \Rightarrow \begin{bmatrix} \textsc{hd-dtr}\ \boxed{1}\begin{bmatrix} \textit{word} \\ \textsc{comps}\ \langle \boxed{2}, \ldots, \boxed{n} \rangle \end{bmatrix} \\ \textsc{dtrs}\quad \langle \boxed{1}, \left[\textsc{ss}\ \boxed{2}\right], \ldots, \left[\textsc{ss}\ \boxed{n}\right] \rangle \end{bmatrix}$$

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.[29] Notice that nothing is said about the SYNSEM value of the phrase. It will be

[28] HEAD here is an abbreviation for SYNSEM|LOC|CAT|HEAD. In later implicational constraints, we abbreviate SYNSEM|LOC|CAT|COMPS as COMPS and SYNSEM|LOC|CAT|SUBJ as SUBJ.

[29] The head could be identified as a [LEX +], [LIGHT +], or [WEIGHT *light*] phrase, to accommodate coordination of heads as in *John* [*knows and likes*] *this record* (Abeillé 2006: Section 5.1).

[COMPS ⟨⟩], as required by the constraint in (11), 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 (32), but that would miss a generalisation. 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.[30] 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.

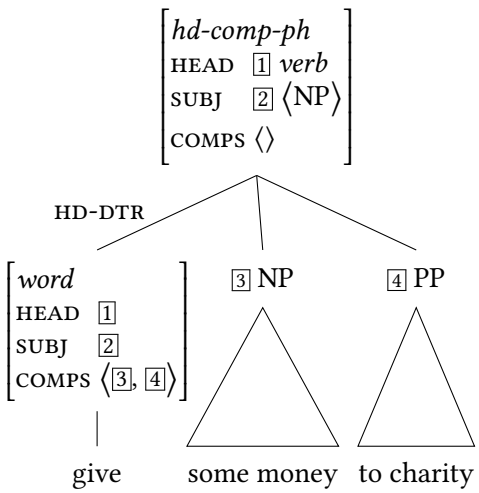

Figure 6: A tree for a head-complement phrase

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

[30] However, binary branching has been assumed in HPSG grammars for a number of languages. See Müller (2021b: Section 3), Chapter 10 of this volume.

(34) *headed-ph* ⇒ $\begin{bmatrix} \textsc{synsem} \; / \boxed{1} \\ \textsc{hd-dtr} \; \left[\textsc{synsem} \; / \boxed{1} \right] \end{bmatrix}$

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 for the fact that a head-complement phrase has the same value for SUBJ as the head daughter, among many other things.

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

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

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.[31] It licenses structures like that in Figure 7.

Finally, we consider the type *hd-filler-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* objects, and it encodes information about unbounded dependency gaps (see Borsley & Crysmann 2021, Chapter 13 of this volume). Here is the relevant constraint:[32]

(36) *hd-filler-ph* ⇒ $\begin{bmatrix} \textsc{slash} & \boxed{1} \\ \textsc{hd-dtr} \; \boxed{2} & \begin{bmatrix} \textsc{comps} & \langle\rangle \\ \textsc{slash} & \left\{ \boxed{3} \right\} \cup \boxed{1} \end{bmatrix} \\ \textsc{dtrs} & \left\langle \left[\textsc{local} \; \boxed{3} \right], \boxed{2} \right\rangle \end{bmatrix}$

[31] 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 (35), it can be analysed as just a word.

[32] We use ∪ for set union. Notice that the mother category does not have to have an empty SLASH list, thus allowing for multiple extractions (*Paul, who could we talk to about?* where *Paul* is understood as object of *about* and *who* as object of *to*).

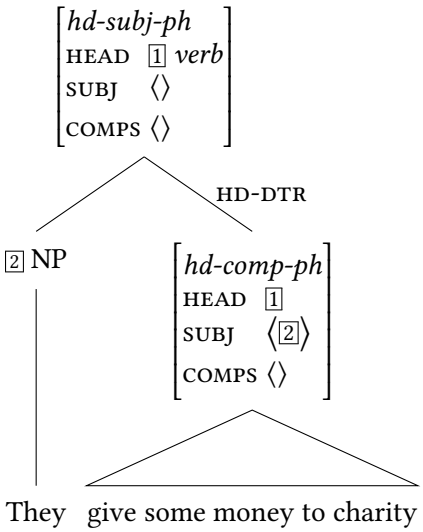

Figure 7: A tree for a head-subject phrase

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* object, and a non-head daughter, whose LOCAL value is the additional *local* object of the head daughter. 1 is normally the empty set.[33] 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]].

Either the Valence Principle or the Generalised 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 sep-

[33] As with (35), 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.

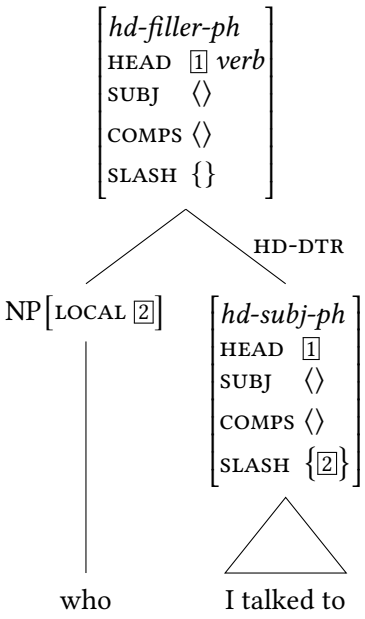

Figure 8: A tree for a head-filler phrase

arated by an arrow and with the head daughter identified by "**H**". Thus, instead of (38a), one has (38b).

(38) a. $\textit{phrase} \Rightarrow \begin{bmatrix} \textsc{synsem}\ X \\ \textsc{dtrs}\ \langle \boxed{1} Y, Z \rangle \\ \textsc{hd-dtr}\ \boxed{1} \end{bmatrix}$

b. *phrase*:
$X \rightarrow \mathbf{H}[Y], Z$

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 in (38a).

In the preceding discussion, we have ignored the semantics of the phrase. Leaving aside quantification and other complex matters, and assuming INDEX and REL(ATION)S as in MRS (as shown in (23) above), the CONTENT of a headed phrase can be handled via two semantic principles: 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 concatenation of the RELS of its DTRS; Copestake et al.

2005: Section 4.3.2, Section 5; Koenig & Richter 2021: Section 6.1, Chapter 22 of this volume).

The type hierarchy in Figure 5 is simplified in a number of respects. It includes no non-headed phrases.[34] 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.

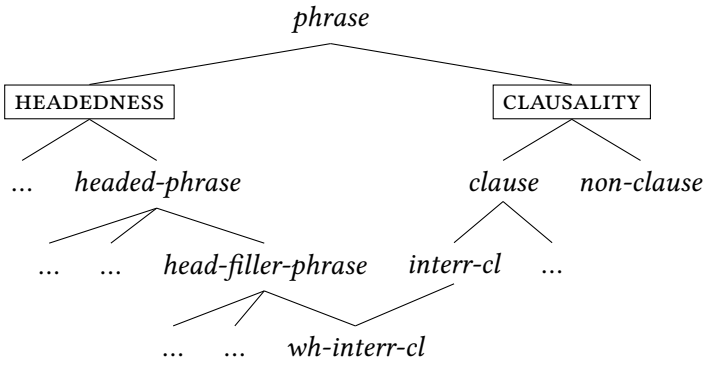

Figure 9: Cross-classification of phrases

Here *wh-interr-cl* is identified as a subtype of *head-filler-phrase* 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.

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.[35] 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.

[34] The most important type of non-headed phrase is coordinate structure. See Abeillé & Chaves (2021), Chapter 16 of this volume for discussion.

[35] As discussed in Section 7.1, in some HPSG work, linear order is a property of so-called order domains, which essentially mediate constituent structure and phonology (see Müller 2021b: Section 6, Chapter 10 of this volume).

Clearly, they can be concatenated in two ways as in (39), or their order may be left unspecified for "free" word order:[36]

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

b. $\begin{bmatrix} \textsc{phon} \boxed{2} \oplus \boxed{1} \\ \textsc{dtrs} \left\langle \left[\textsc{phon} \boxed{1}\right], \left[\textsc{phon} \boxed{2}\right] \right\rangle \end{bmatrix}$

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'

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 constraints on phrasal types. They are commonly written with < ("precedes"). Thus, languages with head-complement order might have the rule in (41a), and languages with complement-head order the rule in (41b).

(41) a. $\left[\textsc{comps} \left\langle \ldots, \boxed{1}, \ldots \right\rangle\right] < \left[\textsc{synsem} \boxed{1}\right]$

b. $\left[\textsc{synsem} \boxed{1}\right] < \left[\textsc{comps} \left\langle \ldots, \boxed{1}, \ldots \right\rangle\right]$

But it should be remembered that ordering rules are well-formedness constraints on structures built with certain concatenations of PHON values as in (39).[37]

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:

(42) a. Kim is late.

b. Is Kim late?

[36] Unspecified means any combination of $\boxed{1}$ and $\boxed{2}$ using the shuffle operation: $\boxed{1} \bigcirc \boxed{2}$. (see footnote 21)

[37] An alternative notation, provided different daughters are distinguished with different names, could be:

(i) a. HD-DTR < COMPS-DTRS

b. COMPS-DTRS < HD-DTR

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. (42a) is a head-subject phrase much like that in Figure 7. Clauses like (42b), 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 (Pollard & Sag 1994: 40). This requires an additional phrase type, which might be called *head-subject-complement-phrase*.[38]

6 Further syntactic structures

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

6.1 Adjuncts

Adverbs, adverbial PPs within VPs, attributive adjectives, and relative clauses within NPs are commonly viewed as adjuncts. Thus, the following illustrate head-adjunct phrases (with the head following the adjunct in (43a) and (43c) and preceding in (43b) and (43d)):

(43) 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(IFIES) whose value is a *synsem* object, while other signs are [MOD *none*]. Thus, (43a) involves the schematic structure in Figure 10.

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: 28–29). In the case of relative clauses, it is more complex because the value of MOD must be coindexed with the *wh*-element, if there is one, or the gap, if there isn't. In (43d), this is reflected in the

[38] In Ginzburg & Sag (2000: 36), it is called *sai-phrase*. In some HPSG work, e.g. Sag et al. (2003: 409–414), 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 alternative valence description for auxiliary verbs.

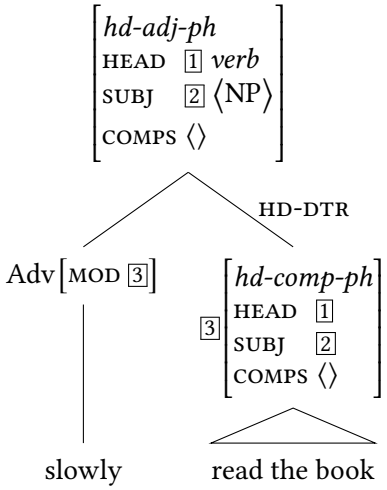

Figure 10: A tree for a head-adjunct phrase

fact that the verb in the relative clause is the singular *impresses* and not the plural *impress*. See Borsley & Crysmann (2021), Chapter 13 of this volume and Arnold & Godard (2021), Chapter 14 of this volume for some discussion.

Notice also that in head-adjunct phrases, the adjunct is not a syntactic head, but may well be the semantic head. This is an example of the difference between syntactic head and semantic head, and between syntactic argument and semantic argument 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. Abeillé & Godard 1997; Bouma et al. 2001: 4; Ginzburg & Sag 2000: 168, Footnote 2). On this view, *in the pub* in (43b) is much like the same phrase in (44), where it is clearly a (predicative) complement:

(44) Kim is in the pub.

Various arguments have been advanced for this position, but it is controversial and it is rejected by Levine (2003), Levine & Hukari (2006: Chapter 3), and Chaves (2009). There is an unresolved issue here.[39]

[39]It has been argued that some adverbs and PPs are adjuncts and others are complements, depending on word order, case, and so on. (see, for example, Przepiórkowski 1999, Hassamal & Abeillé 2014, and Kim 2021: Section 2.3, Chapter 18 of this volume).

6.2 Specifiers and markers

As noted earlier, some HPSG work assumes a feature SPR (SPECIFIER) which is realised by various categories. In some work, subjects are analysed as specifiers (Sag, Wasow & Bender 2003: 100–103), but in other approaches, they are realisations of a SUBJ(ECT) feature, as discussed in the last section. For some HPSG work, e.g. Pollard & Sag (1994: Section 9.4) and Sag et al. (2003: Section 4.3), determiners within NPs are an important example of specifiers. On this view, *the pub* has the schematic structure in Figure 11.

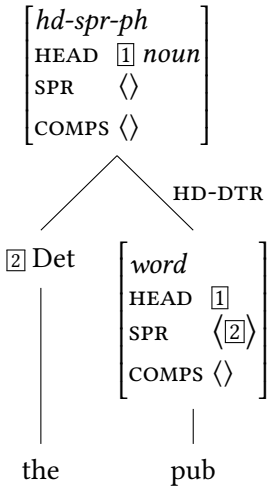

Figure 11: A tree for a head-specifier phrase

Some recent work, e.g. Sag (2012: 84), has adopted a rather different view of at least some determiners, namely that they are what are known as markers, a notion first introduced in Pollard & Sag (1994: Section 1.6). These are non-heads which select the head that they combine with through a SELECT feature (Van Eynde 1998; Van Eynde 2021: Section 2.3, Chapter 8 of this volume) but determine the MARKING value of their mother. Within this approach, *the pub* has the schematic structure in Figure 12.[40]

A marker analysis was originally proposed for complementisers. However, they have also been analysed as heads within HPSG, e.g. in Sag (1997: 456–458)

[40] Work which assumes the SELECT feature also uses it instead of MOD for adjuncts and considers both markers and adjuncts to be "functors" (Van Eynde 1998; Van Eynde 2021: Section 2.3.2, Chapter 8 of this volume).

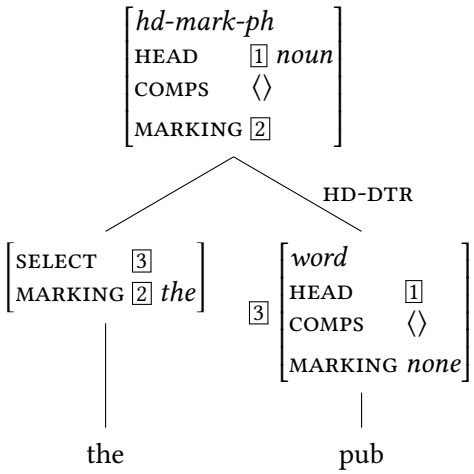

Figure 12: A tree for a head-functor phrase

and Ginzburg & Sag (2000: Section 2.8). There is no consensus here.

7 Further topics

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 (see Müller 2021b, Chapter 10 of this volume). 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:

(45) 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 (Kathol & Pollard 1995). 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 1994; 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 this position, one might propose that (45b) has the schematic analysis in (46).

(46)
$$\begin{bmatrix} \textsc{synsem}\ \mathrm{S} \\ \textsc{dtrs}\ \langle [\textit{a man who looked like Churchill}], [\textit{came into the room}] \rangle \\ \textsc{dom}\ \langle [\textit{a man}], [\textit{came into the room}], [\textit{who looked like Churchill}] \rangle \end{bmatrix}$$

Here the clause has two daughters but three domain elements. The simpler example in (45a) will have two daughters and two domain elements.

It is worth noting that this approach allows a different analysis for interrogatives like (42b). It would be possible to propose an analysis in which they have two daughters and three domain elements as follows:

(47)
$$\begin{bmatrix} \textsc{synsem}\ \mathrm{S} \\ \textsc{dtrs}\ \langle [\textit{Kim}], [\textit{is late}] \rangle \\ \textsc{dom}\ \langle [\textit{is}], [\textit{Kim}], [\textit{late}] \rangle \end{bmatrix}$$

As far as we are aware, no one has proposed such an analysis for English interrogatives, but essentially this analysis is proposed for German interrogatives in Kathol (2000: 81).[41]

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 & Sag 1999). However, they may well have a role to play in more familiar languages (Bonami et al. 1999; Chaves 2014). 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

[41] Kathol (2000) assumes that order domains are divided into topological fields and shows how this idea allows an interesting approach to various aspects of clausal word order. See Borsley (2006) for an application of this idea to negation.

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.

7.2 Sign-Based Construction Grammar

The SBCG version of HPSG will be discussed in some detail in the next chapter (Flickinger, Pollard & Wasow 2021: 68–70), in the chapter on unbounded dependencies (Borsley & Crysmann 2021: Section 10), and in the chapter on HPSG and Construction Grammar (Müller 2021c: Section 1.3.2). 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 Constructional 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 MTR (MOTHER) sign with a list of DAUGHTER signs, one of which is a HEAD-DAUGHTER in a headed construction. Thus, constructions take the form in (48a) and headed-constructions the form in (48b):

(48) a. $\begin{bmatrix} cx \\ \textsc{mtr} \;\; sign \\ \textsc{dtrs} \;\; list(sign) \end{bmatrix}$

b. $\begin{bmatrix} headed\text{-}cx \\ \textsc{mtr} \quad sign \\ \textsc{dtrs} \quad list(sign) \\ \textsc{hd-dtr} \;\; sign \end{bmatrix}$

Constructions are utilised by the Sign Principle, which can be formulated as follows:[42]

(49) 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 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

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

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* (Pollard & Sag 1994: 23). 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 considerations of simplicity do not obviously favour or disfavour the approach.

8 Concluding remarks

In the preceding pages, we have spelled out the basic properties of HPSG and the assumptions it makes about the nature of linguistic analyses and the conduct of linguistic research. We have looked at the types, features, and constraints that are the building blocks of HPSG analyses. We have also outlined the HPSG approach to the lexicon and the basics of its approach to syntax, and we have considered some of the main types of syntactic structure. Finally, we have discussed order domains and SBCG. More can be learned about all of these matters in the chapters that follow.

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.

References

Abeillé, Anne. 2006. In defense of lexical coordination. In Olivier Bonami & Patricia Cabredo Hofherr (eds.), *Empirical issues in syntax and semantics*, vol. 6, 7–36. Paris: CNRS. http://www.cssp.cnrs.fr/eiss6/ (10 February, 2021).

Abeillé, Anne. 2021. Control and Raising. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 489–535. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599840.

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. 2021. Coordination. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 725–776. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599848.

Abeillé, Anne & Danièle Godard. 1997. The syntax of French negative adverbs. In Danielle Forget, Paul Hirschbühler, France Martineau & María Luisa Rivero (eds.), *Negation and polarity: Syntax and semantics: Selected papers from the colloquium Negation: Syntax and Semantics. Ottawa, 11–13 May 1995* (Current Issues in Linguistic Theory 155), 1–28. Amsterdam: John Benjamins Publishing Co. DOI: 10.1075/cilt.155.02abe.

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*, 26–46. Stanford, CA: CSLI Publications. http://csli-publications.stanford.edu/HPSG/2003/abeille-godard.pdf (10 February, 2021).

Ajdukiewicz, Kazimierz. 1935. Die syntaktische Konnexität. *Studia Philosophica* 1. 1–27.

Anderson, Stephen R. 1992. *A-morphous morphology* (Cambridge Studies in Linguistics 62). Cambridge: Cambridge University Press. DOI: 10.1017/CBO9780511586262.

Arnold, Doug & Danièle Godard. 2021. Relative clauses in HPSG. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 595–663. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599844.

Barwise, Jon & John Perry. 1983. *Situations and attitudes*. Cambridge, MA: MIT Press. Reprint as: *Situations and attitudes* (The David Hume Series of Philosophy and Cognitive Science Reissues). Stanford, CA: CSLI Publications, 1999.

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). 23–72. DOI: 10.1007/s11168-010-9070-1.

Bender, Emily M. & Guy Emerson. 2021. Computational linguistics and grammar engineering. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre

Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 1105–1153. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599868.

Bird, Steven & Ewan Klein. 1994. Phonological analysis in typed feature systems. *Computational Linguistics* 20(3). 455–491.

Bonami, Olivier, Danièle Godard & Jean-Marie Marandin. 1999. Constituency and word order in French subject inversion. In Gosse Bouma, Erhard Hinrichs, Geert-Jan M. Kruijff & Richard Oehrle (eds.), *Constraints and resources in natural language syntax and semantics* (Studies in Constraint-Based Lexicalism 3), 21–40. Stanford, CA: CSLI Publications.

Borsley, Robert D. 2006. A linear approach to negative prominence. In Stefan Müller (ed.), *Proceedings of the 13th International Conference on Head-Driven Phrase Structure Grammar, Varna*, 60–80. Stanford, CA: CSLI Publications. http://csli-publications.stanford.edu/HPSG/2006/borsley.pdf (26 November, 2020).

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, 7–26. Paris: CNRS. http://www.cssp.cnrs.fr/eiss8/borsley-eiss8.pdf (10 February, 2021).

Borsley, Robert D. & Berthold Crysmann. 2021. Unbounded dependencies. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 537–594. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599842.

Borsley, Robert D. & Stefan Müller. 2021. HPSG and Minimalism. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 1253–1329. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599874.

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.

Briscoe, Ted J. & Ann Copestake. 1999. Lexical rules in constraint-based grammar. *Computational Linguistics* 25(4). 487–526. http://www.aclweb.org/anthology/J99-4002 (10 February, 2021).

Chaves, Rui P. 2009. Construction-based cumulation and adjunct extraction. In Stefan Müller (ed.), *Proceedings of the 16th International Conference on Head-Driven Phrase Structure Grammar, University of Göttingen, Germany*, 47–67. Stanford, CA: CSLI Publications. http://csli-publications.stanford.edu/HPSG/2009/chaves.pdf (10 February, 2021).

Chaves, Rui P. 2014. On the disunity of right-node raising phenomena: Extraposition, ellipsis, and deletion. *Language* 90(4). 834–886. DOI: 10.1353/lan.2014.0081.

Chomsky, Noam. 1957. *Syntactic structures* (Janua Linguarum / Series Minor 4). Berlin: Mouton de Gruyter. DOI: 10.1515/9783112316009.

Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

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

Copestake, Ann, Dan Flickinger, Carl 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. 2021. Morphology. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 947–999. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599860.

Davis, Anthony R. & Jean-Pierre Koenig. 2021. The nature and role of the lexicon in HPSG. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 125–176. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599824.

Davis, Anthony R., Jean-Pierre Koenig & Stephen Wechsler. 2021. Argument structure and linking. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 315–367. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599834.

De Kuthy, Kordula. 2021. Information structure. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 1043–1078. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599864.

Donohue, Cathryn & Ivan A. Sag. 1999. Domains in Warlpiri. In *Sixth International Conference on HPSG–Abstracts. 04–06 August 1999*, 101–106. Edinburgh: University of Edinburgh.

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

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, Daniel, Carl Pollard & Thomas Wasow. 1985. Structure-sharing in lexical representation. In William C. Mann (ed.), *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, 262–267. Chicago, IL: Association for Computational Linguistics. DOI: 10.3115/981210.981242. https://www.aclweb.org/anthology/P85-1000 (17 February, 2021).

Flickinger, Dan, Carl Pollard & Thomas Wasow. 2021. The evolution of HPSG. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 47–87. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599820.

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. 2021. Complex predicates. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 419–488. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599838.

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

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

Hassamal, Shrita & Anne Abeillé. 2014. Degree adverbs in Mauritian. In Stefan Müller (ed.), *Proceedings of the 21st International Conference on Head-Driven Phrase Structure Grammar, University at Buffalo*, 259–279. Stanford, CA: CSLI Publications. http://cslipublications.stanford.edu/HPSG/2014/hassamal-abeille.pdf (7 August, 2020).

Höhle, Tilman N. 1999. An architecture for phonology. In Robert D. Borsley & Adam Przepiórkowski (eds.), *Slavic in Head-Driven Phrase Structure Grammar* (Studies in Constraint-Based Lexicalism 2), 61–90. Stanford, CA: CSLI Publications. Reprint as: An architecture for phonology. In Stefan Müller, Marga Reis & Frank Richter (eds.), *Beiträge zur deutschen Grammatik: Gesammelte Schriften von Tilman N. Höhle*, 2nd edn. (Classics in Linguistics 5), 571–607. Berlin: Language Science Press, 2019. DOI: 10.5281/zenodo.2588383.

Jackendoff, Ray S. 1977. *$\overline{X}$ syntax: A study of phrase structure* (Linguistic Inquiry Monographs 2). Cambridge, MA: MIT Press.

Kathol, Andreas. 2000. *Linear syntax* (Oxford Linguistics). Oxford: Oxford University Press.

Kathol, Andreas & Carl Pollard. 1995. Extraposition via complex domain formation. In Hans Uszkoreit (ed.), *33rd Annual Meeting of the Association for Computational Linguistics. Proceedings of the conference*, 174–180. Cambridge, MA: Association for Computational Linguistics. DOI: 10.3115/981658.981682.

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* (CSLI Lecture Notes 224), 87–118. Stanford, CA: CSLI Publications.

Kim, Jong-Bok. 2021. Negation. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 811–845. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599852.

Kiss, Tibor. 2005. Semantic constraints on relative clause extraposition. *Natural Language & Linguistic Theory* 23(2). 281–334. DOI: 10.1007/s11049-003-1838-7.

Koenig, Jean-Pierre & Frank Richter. 2021. Semantics. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 1001–1042. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599862.

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

Levine, Robert D. 2003. Adjunct valents, cumulative scopings and impossible descriptions. In Jong-Bok Kim & Stephen Wechsler (eds.), *The proceedings of the 9th International Conference on Head-Driven Phrase Structure Grammar, Kyung Hee University*, 209–232. Stanford, CA: CSLI Publications. http://csli-publications.stanford.edu/HPSG/3/ (10 February, 2021).

Levine, Robert D. & Thomas E. Hukari. 2006. *The unity of unbounded dependency constructions* (CSLI Lecture Notes 166). Stanford, CA: CSLI Publications.

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 D. & Ivan A. 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* (Studies in Constraint-Based Lexicalism 1), 63–78. Stanford, CA: CSLI Publications.

Meurers, W. Detmar. 2001. On expressing lexical generalizations in HPSG. *Nordic Journal of Linguistics* 24(2). 161–217. DOI: 10.1080/033258601753358605.

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. DOI: 10.1023/A:1005815413834.

Monachesi, Paola. 2005. *The verbal complex in Romance: A case study in grammatical interfaces* (Oxford Studies in Theoretical Linguistics 9). Oxford: Oxford University Press. DOI: 10.1093/acprof:oso/9780199274758.001.0001.

Montague, Richard. 1974. *Formal philosophy: Selected papers of Richard Montague*. Richmond H. Thomason (ed.). New Haven, CT: Yale University Press.

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

Müller, Stefan. 2004. Elliptical constructions, multiple frontings, and surface-based syntax. In Gerhard Jäger, Paola Monachesi, Gerald Penn & Shuly Wintner (eds.), *Proceedings of Formal Grammar 2004, Nancy*, 91–109. Stanford, CA: CSLI Publications. http://cslipublications.stanford.edu/FG/2004/ (3 February, 2021).

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*. 1st edn. (Textbooks in Language Sciences 1). Berlin: Language Science Press. DOI: 10.17169/langsci.b25.167.

Müller, Stefan. 2021a. Anaphoric binding. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 889–944. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599858.

Müller, Stefan. 2021b. Constituent order. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 369–417. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599836.

Müller, Stefan. 2021c. HPSG and Construction Grammar. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 1497–1553. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599882.

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. DOI: 10.1007/978-3-662-04230-4_17.

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, Dan 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. DOI: 10.1007/s11168-004-7430-4.

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 4). Chicago, IL: The University of Chicago Press.

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

Prince, Alan & Paul Smolensky. 2004. *Optimality Theory: Constraint interaction in Generative Grammar*. Malden, MA: Blackwell Publishing. DOI: 10.1002/9780470759400.

Przepiórkowski, Adam. 1999. On case assignment and "adjuncts as complements". In Gert Webelhuth, Jean-Pierre Koenig & Andreas Kathol (eds.), *Lexical and constructional aspects of linguistic explanation* (Studies in Constraint-Based Lexicalism 1), 231–245. Stanford, CA: CSLI Publications.

Pullum, Geoffrey K. & Barbara C. Scholz. 2001. On the distinction between generative-enumerative and model-theoretic syntactic frameworks. In Philippe de Groote, Glyn Morrill & Christian Retoré (eds.), *Logical Aspects of Compu-*

tational Linguistics: 4th International Conference (Lecture Notes in Computer Science 2099), 17–43. Berlin: Springer Verlag. DOI: 10.1007/3-540-48199-0_2.

Reape, Mike. 1994. Domain union and word order variation in German. In John Nerbonne, Klaus Netter & Carl Pollard (eds.), *German in Head-Driven Phrase Structure Grammar* (CSLI Lecture Notes 46), 151–198. Stanford, CA: CSLI Publications.

Richter, Frank. 2021. Formal background. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 89–124. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599822.

Richter, Frank & Manfred Sailer. 2004. Basic concepts of Lexical Resource Semantics. In Arnold Beckmann & Norbert Preining (eds.), *ESSLLI 2003 – Course material I* (Collegium Logicum 5), 87–143. Wien: Kurt Gödel Society.

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/richter-sailer.pdf (10 February, 2021).

Sag, Ivan A. 1997. English relative clause constructions. *Journal of Linguistics* 33(2). 431–483. 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, Stanford Department of Linguistics and CSLI's LinGO Lab*, 394–414. Stanford, CA: CSLI Publications. http://csli-publications.stanford.edu/HPSG/2007/ (10 February, 2021).

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 & Thomas 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. 1995. Extraction without traces. In Raul Aranovich, William Byrne, Susanne Preuss & Martha Senturia (eds.), *WCCFL 13: The 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*. 1st edn. (CSLI Lecture Notes 92). 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: Wiley-Blackwell. DOI: 9781444395037.ch10.

Sag, Ivan A. & Thomas Wasow. 2015. Flexible processing and the design of grammar. *Journal of Psycholinguistic Research* 44(1). 47–63. DOI: 10.1007/s10936-014-9332-4.

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. 2021. Idioms. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 777–809. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599850.

Steedman, Mark. 2000. *The syntactic process* (Language, Speech, and Communication 24). Cambridge, MA: MIT Press.

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

Van Eynde, Frank. 1998. The immediate dominance schemata of HPSG. In Peter-Arno Coppen, Hans van Halteren & Lisanne Teunissen (eds.), *Computational Linguistics in the Netherlands 1997: Selected papers from the Eighth CLIN Meeting* (Language and Computers: Studies in Practical Linguistics 25), 119–133. Amsterdam: Rodopi.

Van Eynde, Frank. 2021. Nominal structures. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 275–313. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599832.

Walther, Markus. 1999. *Deklarative prosodische Morphologie: Constraint-basierte Analysen und Computermodelle zum Finnischen und Tigrinya* (Linguistische Arbeiten 399). Tübingen: Max Niemeyer Verlag. DOI: 10.1515/9783110911374.

Wasow, Thomas. 2021. Processing. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 1081–1104. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599866.

Wechsler, Stephen & Ash Asudeh. 2021. HPSG and Lexical Functional Grammar. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook* (Empirically Oriented Theoretical Morphology and Syntax), 1395–1446. Berlin: Language Science Press. DOI: 10.5281/zenodo.5599878.