Introduction to Head-driven Phrase Structure Grammar*

Steve Harlow
sjh1@york.ac.uk

May 5, 2009

Contents

*Draft only. Please do not quote or cite

List of Tables

List of Exercises

1 Signs[1]

The basic linguistic unit in HPSG is called a **sign**. A sign is a collection of information about a linguistic entity's pronunciation, its syntax and its semantics. Each sign is defined as a feature structure of sort *sign* for which the attributes PHONOLOGY (PHON),[2] SYNSEM and some others (that we will come to later) are defined. This information is shown in the following AVM, in which sorts are (following standard convention) written in lower case italics and attributes are written in small capitals.

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

The value of PHON is a list of feature structures of sort *phonstring*. In this book we will have nothing more to say about this aspect of language and will simply represent PHON values as lists of words in their conventional orthographic form. The attribute SYNSEM takes feature structures of sort *synsem* as values and we will look next at what this means.

Syntactic and semantic information is partitioned into two kinds, LOCAL and NON-LOCAL information. Non-local information is that which is involved in unbounded dependency constructions in which a constituent is displaced from its normal position (e.g. *"ice cream"* in *"ice cream, I like"*) and will be discussed later in this chapter; we will take up LOCAL information here. This organisation of *synsem* is shown in the following AVM.

$$\begin{bmatrix} \textit{synsem} & \\ \textsc{loc} & \textit{local} \\ \textsc{nonloc} & \textit{non-local} \end{bmatrix}$$

LOCAL information comprises information about the purely syntactic properties of the sign, encoded under the attribute CAT(EGORY), and about its semantics, CONTENT, as shown in the following AVM.

$$\begin{bmatrix} \textit{local} & \\ \textsc{cat} & \textit{cat} \\ \textsc{content} & \textit{content} \end{bmatrix}$$

CATEGORY information comprises information about head properties (HEAD), whether the sign is a lexical category or not (LEX) and what additional syntactic arguments the sign combines with in order to form larger signs (SUBJECT, COMPLEMENTS and ARGUMENT-STRUCTURE) - what we will call **valence** information (also commonly known as **subcategorisation**).

$$\begin{bmatrix} \textit{cat} & \\ \textsc{head} & \textit{head} \\ \textsc{lex} & \textit{boolean} \\ \textsc{subj} & \textit{list(synsem)} \\ \textsc{comps} & \textit{list(synsem)} \\ \textsc{arg-str} & \textit{list(synsem)} \end{bmatrix}$$

[1] This chapter is part of a larger book. It therefore sometimes contains references to other chapters not included here. Don't worry about them.

[2] We will use abbreviated forms for both attributes and sorts when their interpretation is obvious.

The value of HEAD consists of part of speech information and ultimately specifies whether the sign is nominal, verbal, prepositional etc. The sort *head* is the top of a sortal hierarchy which is partitioned into two subsorts, *substantive* and *functional*. The former has the atomic subsorts *noun*, *verb*, *preposition* and *adjective*; the latter has various subsorts, of which *determiner* is the only example we will give here. This is shown in the sortal hierarchy in figure 1.[3]

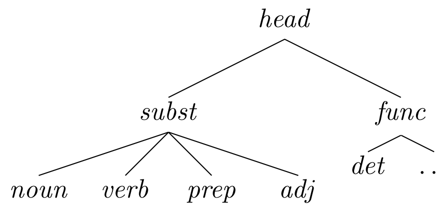


Figure 1: The *head* sortal hierarchy.

If we put all of this together, omitting for the moment NONLOCAL, CONTENT and valence information, the sign for a verb such as *"drinks"* will look like figure 2.

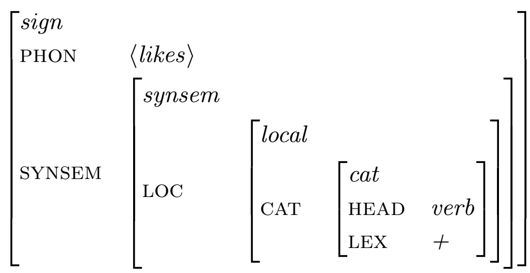
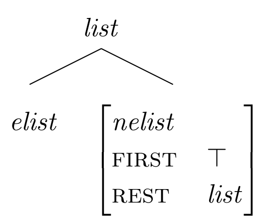


Figure 2: Partial description of the verb *"likes"*

2 Valence

Now that we have provided a brief outline of how linguistic information is represented in HPSG, we will proceed to fill in more of the details and in this section we will look in more detail at valence. *"Likes"* is not merely a verb, it is a transitive verb – one that, to form a sentence, needs to combine with two noun phrases, a subject and a direct object. This information, as mentioned above, is encoded in the values for SUBJECT, COMPLEMENTS and ARGUMENT-STRUCTURE.

Each of these attributes takes a list as its value. *list* is thus a sort, and has subsorts *empty-list* (*elist*) and *non-empty-list* (*nelist*). *nelist* has the attributes FIRST and REST, which in turn take values of sorts $\top$ and *list* respectively.[4] This is shown in the following diagram.



[3]Each of the maximal subsorts has further attributes defined for it. We will omit these for the moment and return to them later when they become directly relevant to the presentation.

[4]$\top$ is the (unique) most general sort.

The SUBJ and COMPS values consist of lists of linguistic entities with which a sign can combine. *"Likes"* combines with a single subject NP and a single complement NP. Using NP to abbreviate the actual feature structures involved for the moment, this means that the SUBJ and COMPS values for *"likes"* look like this.

$$\begin{bmatrix} \textit{cat} & \\ \textsc{subj} & \begin{bmatrix} \textit{nelist} & \\ \textsc{first} & \text{NP} \\ \textsc{rest} & \textit{elist} \end{bmatrix} \\ \textsc{comps} & \begin{bmatrix} \textit{nelist} & \\ \textsc{first} & \text{NP} \\ \textsc{rest} & \textit{elist} \end{bmatrix} \end{bmatrix}$$

For a verb such as *"gives"*, which takes two NP complements (in examples like *"Kim gives her friends a lot of attention"*), the COMPS value would look like this.

$$\begin{bmatrix} \textit{cat} & \\ \textsc{comps} & \begin{bmatrix} \textit{nelist} & \\ \textsc{first} & \text{NP} \\ \textsc{rest} & \begin{bmatrix} \textit{nelist} & \\ \textsc{first} & \text{NP} \\ \textsc{rest} & \textit{elist} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

These representations of lists can become difficult to decipher, particularly when embedded in complex feature structures, so we will follow the common notational convention of using angle brackets to enclose the elements of a list, thus modifying the above representation of the transitive verb *"likes"* to look like this:

$$\begin{bmatrix} \textit{cat} & \\ \textsc{subj} & \langle \text{NP} \rangle \\ \textsc{comps} & \langle \text{NP} \rangle \end{bmatrix}$$

and the di-transitive verb *"gives"* to look like this

$$\begin{bmatrix} \textit{cat} & \\ \textsc{subj} & \langle \text{NP} \rangle \\ \textsc{comps} & \langle \text{NP, NP} \rangle \end{bmatrix}$$

The value of ARGUMENT-STRUCTURE (ARG-STR) is simply the concatenation of the values of SUBJ and COMPS. We will represent concatenation with the an infix operator, $\oplus$, and represent the complete valence values for transitive verbs as follows (in which the value for ARG-STR will be ⟨NP, NP⟩). The tags $\boxed{1}$ and $\boxed{2}$ indicate that the co-tagged feature structures are identical.

$$\begin{bmatrix} \textit{cat} & \\ \textsc{subj} & \boxed{1}\langle \text{NP} \rangle \\ \textsc{comps} & \boxed{2}\langle \text{NP} \rangle \\ \textsc{arg-str} & \boxed{1} \oplus \boxed{2} \end{bmatrix}$$

If we combine this information with that supplied in figure 2, we get the more fully specified AVM in figure 3.

Exercise 1 *Draw AVMs for an intransitive verb such as "smiles" and a ditransitive verb such as "gives".*

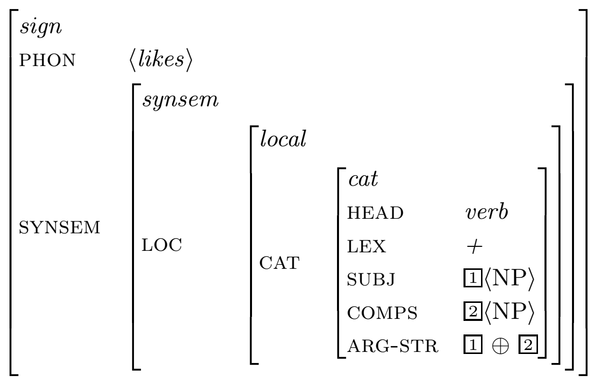


Figure 3: Partial description the verb *"likes"* with valence values added.

3 Words and phrases

In the previous section we outlined the properties of the feature structures associated with lexical items. Lexical items combine with other categories of specific kinds (noun phrases, adjectival phrases, sentences etc.) to form more complex phrases. So far we have only been dealing with one kind of sign. If we are to recognise a distinction between lexical and phrasal signs, we need to partition the sort *sign* into two subsorts – *word* for the lexical signs and *phrase* for the phrasal signs. Subsorts inherit feature specifications from their supersorts, so *word* and *phrase* will automatically have the attributes PHONOLOGY and SYNSEM, with appropriate values. Phrasal signs (of subsort *phrase*) have additional attributes containing information about the daughter categories that make up the phrase. They also take a negative value for the attribute LEX. Signs of subsort *word*, on the other hand, are positively specified for the LEX feature and are not defined for daughter attributes. In addition, only words are defined for the attribute ARG-STR introduced above. This information is summarised in figure 4.

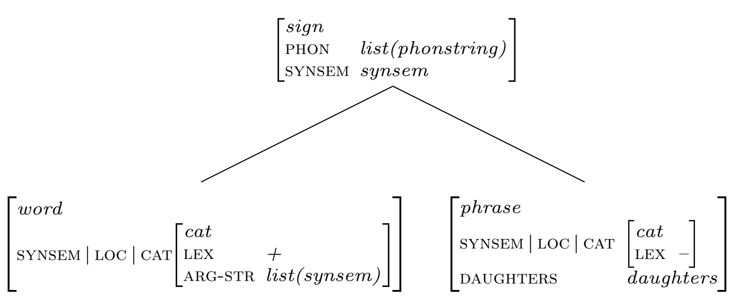


Figure 4: The sortal hierarchy for *sign*

The sort *phrase* is partitioned into a number of subsorts as shown in figure 5, and the major subsorts are exemplified briefly in table 1. We will discuss the properties of these phrases in some detail in the following pages, introducing further constraints which specify which attributes and values are appropriate for the various subsorts.

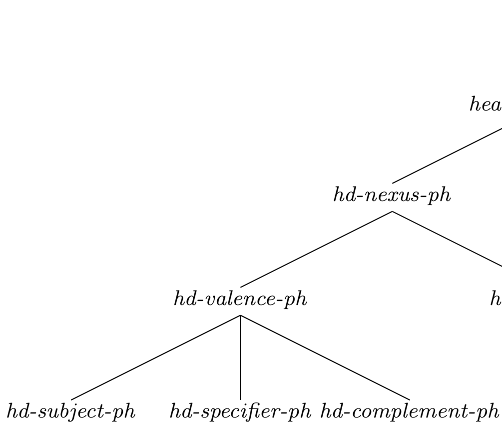


Figure 5: The sortal hierarchy for *phrase*

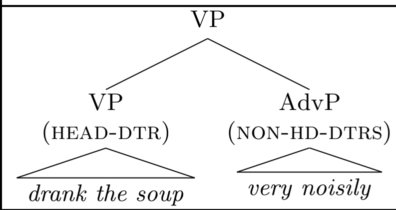
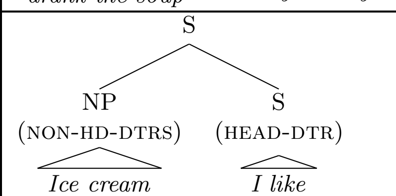
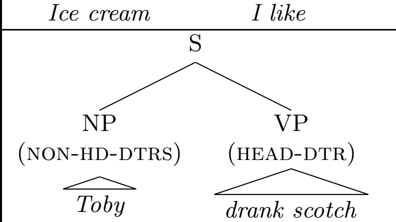
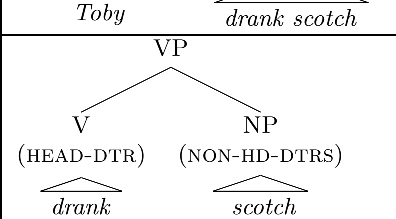
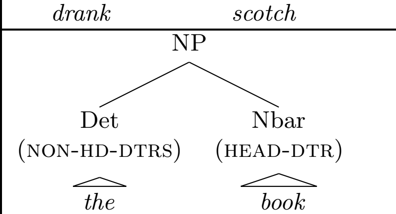
SORT	EXAMPLE
hd-adjunct-ph	
hd-filler-ph	
hd-subject-ph	
hd-complement-ph	
hd-specifier-ph	

Table 1: Examples of phrases

3.1 Complements

We will start by looking at the sort *head-complement-phrase*. This is a subsort of *headed-phrase*; a sort which is appropriate for the features HEAD-DAUGHTER and NON-HEAD-DAUGHTERS as shown in the following AVM.

$$\textit{headed-phrase} \implies \begin{bmatrix} \text{DTRS} & \begin{bmatrix} \textit{dtrs} & \\ \text{HEAD-DTR} & \textit{sign} \\ \text{NON-HD-DTRS} & \textit{list(sign)} \end{bmatrix} \end{bmatrix}$$

Only subsorts of *phrase* are defined for the various DAUGHTER attributes; words do not have daughters. The sort *head-complement-phrase* inherits these constraints and in addition must satisfy the following constraint:

Head-Complement Phrase[5]

$$\textit{hd-comp-ph} \implies \begin{bmatrix} \text{SYNSEM}|\ldots \begin{bmatrix} \text{SUBJ} & \boxed{1} \\ \text{SPR} & \boxed{2} \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{DTRS} \begin{bmatrix} \text{HD-DTR} & \begin{bmatrix} \textit{word} \\ \text{SYNSEM}|\ldots \begin{bmatrix} \text{SUBJ} & \boxed{1} \\ \text{SPR} & \boxed{2} \\ \text{COMPS} & \langle \boxed{3}, \ldots, \boxed{n} \rangle \end{bmatrix} \end{bmatrix} \\ \text{NON-HD-DTRS} & \langle [\text{SYNSEM}\ \boxed{3}], \ldots, [\text{SYNSEM}\ \boxed{n}] \rangle \end{bmatrix} \end{bmatrix}$$

This states that a well-formed *head-complement-phrase*

- must have a *lexical* head daughter (of sort *word*),
- must have an empty COMPS value
- the values of its head daughter's COMPS attribute must be token identical to the SYNSEM values of its non-head daughters, and
- its SUBJ and SPR values are identical to those of its head daughter

The combined effect of the second and third of these requirements is that the number and syntactic category of a lexical head's sisters must be identical to the items listed on the head's COMPS list and the whole phrase will be 'complement saturated', that is, will not combine with any further complements.

The partial lexical entry for *"likes"* (which is of sort *word*) given in figure 3 has the COMPS value NP. So, if we unify the HEAD-DAUGHTER value of *head-complement-phrase* with the lexical entry for *"likes"*, the resulting value for the attribute NON-HD-DTRS is constrained to be a list consisting of a sign containing [SYNSEM NP]. A phrase headed by *"likes"* will have the feature structure shown in figure 6.

Restricting the elements of the COMPS list to feature structures of sort *synsem* imposes the constraint that the information a head selects for is purely local and cannot reach into the category selected and require, say, that the complement have some particular configuration of daughters of

[5] SYNSEM|... abbreviates the path SYNSEM|LOCAL|CAT.

$$\begin{bmatrix} \textit{hd-comp-ph} & \\ \text{SYNSEM}\,|\,\text{LOC}\,|\,\text{CAT} & \begin{bmatrix}\text{SUBJ} & \boxed{1}\\ \text{COMPS} & \langle\,\rangle\end{bmatrix} \\ \text{DTRS} & \begin{bmatrix} \text{HD-DTR} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{likes}\rangle \\ \text{SYNSEM} & \begin{bmatrix} \textit{synsem} & \\ \text{LOC} & \begin{bmatrix}\textit{local} & \\ \text{CAT} & \begin{bmatrix} \textit{cat} & \\ \text{HEAD} & \textit{verb}\\ \text{LEX} & + \\ \text{SUBJ} & \boxed{1}\langle \text{NP}\rangle \\ \text{COMPS} & \boxed{2}\langle \boxed{3}\text{NP}\rangle \\ \text{ARG-STR} & \boxed{1}\oplus\boxed{2}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{bmatrix} \\ \text{NON-HD-DTRS} & \left\langle \begin{bmatrix}\text{SYNSEM} & \boxed{3}\end{bmatrix}\right\rangle \end{bmatrix}\end{bmatrix}$$

Figure 6: A head complement phrase.

its own or that it have a particular pronunciation. There are good reasons to believe that valence selections are indeed blind in this way to the internal structure of the categories selected.

The description that we have given of *head-complement-phrase* says nothing about the part of speech to which the phrase belongs. It is equally applicable to nouns, verb, prepositions and adjectives. It is, however, necessary to state that noun phrases have nouns as their head daughters, verb phrases have verbs, and so on. (Recall the discussion in section**??** of chapter**??**.) This is achieved by a general constraint on the sort *headed-phrase* (a super-sort of *head-complement-phrase*, which therefore inherits the constraint from it).[6]

The Head Feature Principle

$$\textit{headed-ph} \implies \begin{bmatrix}\text{HEAD} & \boxed{1} \\ \text{HD-DTR} & \begin{bmatrix}\text{HEAD} & \boxed{1}\end{bmatrix}\end{bmatrix}$$

The Head Feature Principle requires that the sign's HEAD value be token identical to that of its head daughter. If we add this requirement to the AVM in figure 6, we get figure 7.

[6] We will often adopt the convention of suppressing path information when this is recoverable from the context, so that, instead of writing, for example,

$$\begin{bmatrix}\textit{headed-ph} & \\ \text{SYNSEM}\,|\,\text{LOC}\,|\,\text{CAT}\,|\,\text{HEAD} & \textit{verb}\end{bmatrix}$$

we will simply write

$$\begin{bmatrix}\textit{headed-ph} & \\ \text{HEAD} & \textit{verb}\end{bmatrix}$$

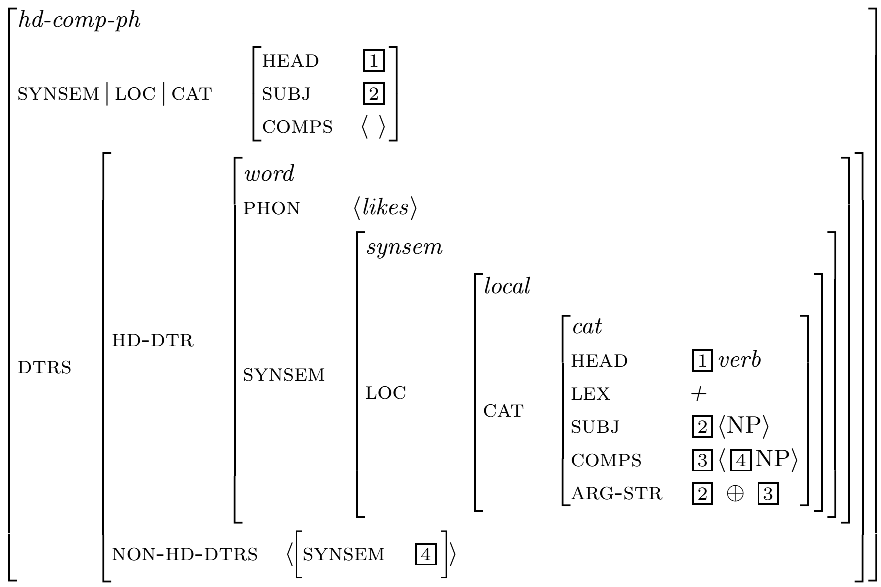


Figure 7: A verb phrase.

Although tree structures have no formal status in HPSG, it may help to follow the rather complex interrelationships in the feature structures we are discussing if we use annotated trees as an informal notation. The branching of the tree represents information about its daughters, so this allows us to omit the various daughters attributes. If a daughter is not present in a sub-tree, we simply omit it. The PHONOLOGY value is represented by the terminal nodes of the tree, and we will omit path information. Using these conventions, we can represent the information contained in the AVM in figure 7 by the tree in figure 8.

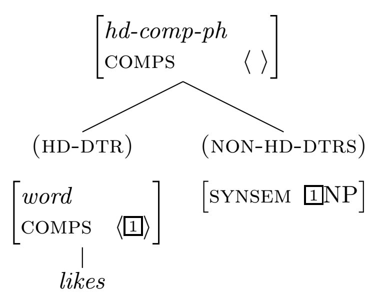


Figure 8: Tree illustrating a head complement phrase.

If we put all this information together and supply a PHONOLOGY value for the complement NP, the feature structure corresponding to the verb phrase *"drinks scotch"* is shown in figure 9.

"Likes" is a mono-transitive verb (i.e. it takes a single NP complement). For ditransitives such as *"gives"*, which can take two NP complements, the only modification that we need to make is the addition of a lexical entry for *"gives"*. This requires a COMPS value which is a list of two NPs:

$$\left[\text{COMPS} \quad \langle \text{NP}, \text{NP} \rangle\right]$$

Nothing further is required to allow ditransitive verbs to form head complement phrases. The definition of the sort *head-complement-phrase*, together with the Head Feature Principle, accounts for all head-complement constructions, in contrast to the range of rules which would be needed in a simple Phrase Structure grammar.

$$
\begin{bmatrix}
\textit{hd-comp-ph} & \\
\text{PHON} & \boxed{1} \oplus \boxed{2} \\
\text{HEAD} & \boxed{3} \\
\text{SUBJ} & \boxed{4} \\
\text{COMPS} & \langle\ \rangle \\
\text{HD-DTR} & \begin{bmatrix}
\textit{word} & \\
\text{PHON} & \boxed{1}\langle \textit{likes} \rangle \\
\text{SYNSEM} & \begin{bmatrix}
\textit{synsem} & \\
\text{LOC} & \begin{bmatrix}
\textit{local} & \\
\text{CAT} & \begin{bmatrix}
\textit{cat} & \\
\text{HEAD} & \boxed{3}\,\textit{verb} \\
\text{LEX} & + \\
\text{SUBJ} & \boxed{4}\langle \text{NP} \rangle \\
\text{COMPS} & \boxed{5}\langle \boxed{6}\text{NP} \rangle \\
\text{ARG-STR} & \boxed{4} \oplus \boxed{5}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix} \\
\text{NON-HD-DTRS} & \left\langle \begin{bmatrix}
\textit{sign} & \\
\text{PHON} & \boxed{2}\langle \textit{scotch} \rangle \\
\text{SYNSEM} & \boxed{6}
\end{bmatrix} \right\rangle
\end{bmatrix}
$$

Figure 9: Partial description of the VP *"likes scotch"*.

Exercise 2

Expand the abbreviated paths in the AVM in figure 9 .

Exercise 3

Give a lexical entry for di-transitive "gives" and show how the constraints associated with the sort head-complement-phrase and the Head Feature Principle interact to define a sign for "gives Andrew scotch".

Exercise 4

Draw AVMs showing the valence values of the underlined verbs in the following examples (using NP and PP as aliases for feature structure descriptions).

1. *"Toby gave scotch to Andrew"*
2. *"Andrew talked about Toby to Portia"*

Draw partial AVMs for the verb phrases

1. *"gave scotch to Andrew"*
2. *"talked about Toby to Portia"*

3.2 Subjects

Now that we have shown how complements are handled, let us turn our attention to subjects. For these we need to introduce a definition of the sort *head-subject-phrase*. (Cf. table 1.)

Head-Subject Phrase

$$
\textit{hd-subj-ph} \quad \Longrightarrow \quad
\begin{bmatrix}
\text{PHON} & \boxed{1} \oplus \boxed{2} \\
\text{SUBJ} & \langle\ \rangle \\
\text{SPR} & \boxed{3} \\
\text{COMPS} & \boxed{4} \\
\text{HEAD-DTR} & \begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \boxed{2} \\ \text{SUBJ} & \langle \boxed{5} \rangle \\ \text{SPR} & \boxed{3}\langle\ \rangle \\ \text{COMPS} & \boxed{4}\langle\ \rangle \end{bmatrix} \\
\text{NON-HD-DTRS} & \left\langle \begin{bmatrix} \textit{sign} & \\ \text{PHON} & \boxed{1} \\ \text{SYNSEM} & \boxed{5} \end{bmatrix} \right\rangle
\end{bmatrix}
$$

Note that, in contrast to head-complement phrases, the head daughter in a head-subject phrase must be a phrase and that the NON-HEAD-DAUGHTERS value is a singleton list.

This definition, in conjunction with the Head Feature Principle licenses feature structures like that in figure 10 in which the abbreviated AVM for the head daughter subsumes the more detailed description in figure 9.

$$
\begin{bmatrix}
\textit{hd-subj-ph} & \\
\text{PHON} & \boxed{1} \oplus \boxed{2} \\
\text{HEAD} & \boxed{3} \\
\text{SUBJ} & \langle\ \rangle \\
\text{COMPS} & \langle\ \rangle \\
\text{LEX} & - \\
\text{HD-DTR} & \begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \boxed{2}\langle \textit{likes, scotch} \rangle \\ \text{HEAD} & \boxed{3}\,\textit{verb} \\ \text{SUBJ} & \langle \boxed{4} \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{LEX} & - \end{bmatrix} \\
\text{NON-HD-DTRS} & \left\langle \begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \boxed{1}\langle \textit{Toby} \rangle \\ \text{SYNSEM} & \boxed{4}\ \text{NP} \end{bmatrix} \right\rangle
\end{bmatrix}
$$

Figure 10: Partial AVM for the sentence *"Toby likes scotch"*.

Note that one consequence of this analysis is that sentences are analysed as being projections of verbs (i.e. they are headed by categories containing the feature structure [HEAD *verb*]). The relationships between the different verbal projections that we have encountered are summarised in table 2. Categories like sentences, which have empty SUBJ and COMPS lists are referred to as **saturated** categories; verb phrases, by contrast, which have a non-empty SUBJ value, are examples of a **partially saturated** categories.

Verb	*word* HEAD *verb* SUBJ [1]⟨*synsem*⟩ COMPS [2]*list(synsem)* ARG-STR [1] ⊕ [2] LEX +
Verb Phrase	*hd-comp-ph* HEAD *verb* SUBJ ⟨*synsem*⟩ COMPS ⟨ ⟩ LEX – HD-DTR *word* NON-HD-DTRS *list(sign)*
Sentence	*hd-subj-ph* HEAD *verb* SUBJ ⟨ ⟩ COMPS ⟨ ⟩ LEX – HD-DTR *phrase* NON-HD-DTRS ⟨*sign*⟩

Table 2: Verbal projections.

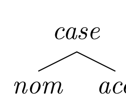
3.3 Nouns and Noun Phrases

In this section, we will go into the details of the feature structure lying behind the alias NP. One of the most striking consequences of the kind of HPSG analysis that we have outlined above is that almost all the apparatus that we need for nouns and noun phrases has already been provided. The part of the feature structure that identified the examples discussed above as verbs or projections of verbs is the value of the feature HEAD. To provide a parallel analysis of nominal projections, all we need to do is to change this value to one appropriate for nouns.

Firstly, recall from page 5 that the sort *head* has *noun* as one of its subsorts. The sort *noun* is in its turn defined for the feature CASE, with value *case*.

$$\begin{bmatrix} \textit{noun} & \\ \text{CASE} & \textit{case} \end{bmatrix}$$

The sort *case* has the two subsorts *nominative* and *accusative* (and possibly more).[7]



[7]Cf. the discussion of English pronouns in chapter **??**, section **??**. Subject pronouns, such as *"I"*, will contain the specification [HEAD [CASE *nom*]], while non-subject pronouns, such as *"me"* will contain [HEAD [CASE *acc*]]. Common nouns, which in English do not exhibit any overt case marking will be specified only for [HEAD [CASE *case*]].

This feature structure forms the value of the HEAD of every nominal category, both lexical and phrasal. It is the locus of the distinction between nominal categories and categories belonging to other parts of speech.

A proper noun such as *"Toby"* will a have feature structure very similar to that for the verb *"likes"* in figure 3. The only differences will be

- PHONOLOGY, which has the value ⟨*Toby*⟩, rather than ⟨*likes*⟩,
- HEAD, which will have the value $\begin{bmatrix} \textit{noun} & \\ \text{CASE} & \textit{case} \end{bmatrix}$, rather than *verb*, and
- the valence values SUBJ and COMPS, which will both have the empty list as value, since proper nouns take neither complements nor subjects

The result is shown in the AVM in figure 11.

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{Toby} \rangle \\ \text{SYNSEM} & \begin{bmatrix} \textit{synsem} & \\ \text{LOC} & \begin{bmatrix} \textit{local} & \\ \text{CAT} & \begin{bmatrix} \textit{cat} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{CASE} & \textit{case} \end{bmatrix} \\ \text{LEX} & + \\ \text{SUBJ} & \boxed{1} \langle\ \rangle \\ \text{COMPS} & \boxed{2} \langle\ \rangle \\ \text{ARG-STR} & \boxed{1} \oplus \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 11: Partial AVM for the proper noun *"Toby"*.

This lexical entry satisfies the definition of the head daughter in a head-complement phrase, which, together with the Head Feature, licenses the head-complement phrase shown in figure 12. This is the nominal counterpart of the verb phrase shown in figure 9.

$$\begin{bmatrix} \textit{hd-comp-ph} & \\ \text{PHON} & \boxed{1} \\ \text{HEAD} & \boxed{3} \\ \text{SUBJ} & \boxed{5} \\ \text{COMPS} & \langle\ \rangle \\ \text{LEX} & - \\ \text{HD-DTR} & \begin{bmatrix} \textit{word} & \\ \text{PHON} & \boxed{1} \langle \textit{Toby} \rangle \\ \text{HEAD} & \boxed{3} \begin{bmatrix} \textit{noun} & \\ \text{CASE} & \textit{case} \end{bmatrix} \\ \text{SUBJ} & \boxed{5} \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{LEX} & + \end{bmatrix} \\ \text{NON-HD-DTRS} & \langle\ \rangle \end{bmatrix}$$

Figure 12: Partial AVM for the Nbar *"Toby"*.

If we turn our attention to common nouns, such as *"dagger"*, we need to take into account the presence of determiners in phrases such as *"the dagger"*, *"every dagger"*, etc. It is tempting to think that determiners are the nominal counterpart to subjects – where verbs require subjects, nouns may require determiners. However, things turn out to be more complex, since common nouns can have both a subject and a determiner, as does the common noun *"clown"* in the following examples.

(1) *"Toby is a clown."*

(2) *"Everybody considered Toby a clown."*

Note that the semantics of both of these sentences require the sub-translation *clown1(t)* as a component, exactly parallel to *drink1(t)* for the semantics of *"Toby drinks"*, in which the argument t is indisputably the translation of the subject NP. The HPSG solution to this is to posit an additional valence feature called SPECIFIER (abbreviated to SPR) which specifies what determiner if any the head requires. This requires the following modification of the features appropriate to the sort *category*.

$$\begin{bmatrix} \textit{cat} & \\ \text{HEAD} & \textit{head} \\ \text{LEX} & \textit{boolean} \\ \text{SUBJ} & \boxed{6}\ \textit{list(synsem)} \\ \text{SPR} & \boxed{7}\ \textit{list(synsem)} \\ \text{COMPS} & \boxed{8}\ \textit{list(synsem)} \\ \text{ARG-STR} & \boxed{6} \oplus \boxed{7} \oplus \boxed{8} \end{bmatrix}$$

"Dagger" will therefore have the following *category* specification.

$$\begin{bmatrix} \textit{cat} & \\ \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{CASE} & \textit{case} \end{bmatrix} \\ \text{LEX} & + \\ \text{SUBJ} & \boxed{1}\ \langle\ \rangle \\ \text{SPR} & \boxed{2}\langle \text{DetP} \rangle \\ \text{COMPS} & \boxed{3}\ \langle\ \rangle \\ \text{ARG-STR} & \boxed{1} \oplus \boxed{2} \oplus \boxed{3} \end{bmatrix}$$

Specifiers comprise a broader class than determiners and include the underlined items in the following examples.

(3) *"Macbeth was very/too/six feet tall."*

(4) *"Glamis is just/right/four miles over the border."*

(5) *"Toby drank too/very/very much too fast."*

Heads select their specifiers, just as they select their subject and complements, so common nouns like *"dagger"* will contain the valence specification [SPR ⟨DetP⟩],[8] while proper nouns like *"Toby"* will have [SPR ⟨ ⟩]. However, in contrast to subjects and complements, specifiers also select the head with which they co-occur; determiners require a nominal sister, degree specifiers like *"very"* and *"too"* require an adjectival sister and so on.

Restricting attention to determiners, recall that the sort *det* is a subsort of *head* (cf. the sortal hierarchy on page 5). The sort *det* is appropriate for a new feature SPECIFIED (abbreviated SPEC) which determines the kind of category with which the determiner combines. So the head value for determiners looks like this

[8] The fact that determiners can themselves be complex, like *"six feet"*, *"four miles"* and *"very much too"* in (3)-(5) requires the value for SPR here should be a determiner phrase (DetP), rather than just a determiner.

$$\begin{bmatrix} \textit{cat} & \\ \text{HEAD} & \begin{bmatrix} \textit{det} & \\ \text{SPEC} & \text{Nbar} \end{bmatrix} \\ \text{LEX} & + \\ \text{SUBJ} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix}$$

in which Nbar abbreviates the following feature description.

$$\begin{bmatrix} \textit{synsem} & \\ \text{LOC} \mid \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SPR} & \langle \text{DetP} \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{LEX} & - \end{bmatrix} \end{bmatrix}$$

A determiner is therefore a category which selects, via the SPEC attribute, a category which itself selects, via its SPR attribute, a determiner phrase. If we replace the aliases DetP and NP in the above AVMs we come to figure 13 as the feature structure representation of the determiner *"every"* (in which LOC|...|SPEC abbreviates the path LOC|CAT|HEAD|SPEC), and to figure 14 as that of the common noun *"dagger"*. Note that both of these feature structures are cyclic.

$$\begin{bmatrix} \textit{word} & & \\ \text{PHON} & \langle \textit{the} \rangle & \\ \text{SYNSEM} & \boxed{1} & \begin{bmatrix} \textit{synsem} & \\ \text{LOC} \mid \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{det} & \\ \text{SPEC} & \begin{bmatrix} \textit{synsem} & \\ \text{LOC} \mid \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SPR} & \left\langle \begin{bmatrix} \textit{synsem} & \\ \text{LOC} \mid \ldots \mid \text{SPEC} & \boxed{1} \end{bmatrix} \right\rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{LEX} & - \end{bmatrix} \end{bmatrix} \end{bmatrix} \\ \text{SUBJ} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{LEX} & + \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 13: Partial feature structure of the determiner *"the"*.

Now that we have a lexical category of determiners, we need to define the kind of constituent structures in which they can occur. We do this by specifying the attributes appropriate to the sort *head-specifier-phrase*. (Cf. table 1.)

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{dagger} \rangle \\ \text{SYNSEM} & \boxed{1} \begin{bmatrix} \textit{synsem} & \\ \text{LOC}\,|\,\text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SUBJ} & \langle\ \rangle \\ \text{SPR} & \left\langle \begin{bmatrix} \textit{synsem} & \\ \text{LOC}\,|\,\text{CAT}\,|\,\text{HEAD} & \begin{bmatrix} \textit{det} & \\ \text{SPEC} & \boxed{1} \end{bmatrix} \\ \text{LEX} & - \end{bmatrix} \right\rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{LEX} & + \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 14: Partial feature structure of the common noun *"dagger"*.

Head-Specifier Phrase

$$\textit{hd-spr-ph} \Longrightarrow \begin{bmatrix} \text{PHON} & \boxed{1} \oplus \boxed{2} \\ \text{SPR} & \langle\ \rangle \\ \text{HD-DTR} & \begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \boxed{2} \\ \text{SPR} & \langle \boxed{3} \rangle \end{bmatrix} \\ \text{NON-HD-DTRS} & \left\langle \begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \boxed{1} \\ \text{SYNSEM} & \langle \boxed{3} \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

To illustrate we will show the gross structure of the NP *"the dagger"* in figure 15.[9]

Complex AVMs like figure 15 are rather difficult to digest. As we have seen above, annotated trees can provide a more readable informal notation and we will specify a number of additional notational conventions to be used in writing such trees that will serve to suppress clutter.

- We will use the symbol X to stand for categories of sort *word*, i.e. lexical categories which contain the feature specification [LEX +])
- We will use the symbol XP to stand for an object of sort *phrase*
- We will use X'[10] to stand for an object of sort $\begin{bmatrix} \textit{phrase} & \\ \text{SPR} & \langle [\ ] \rangle \end{bmatrix}$, i.e. a phrase with a non-empty SPR value
- We will use X''[11] to stand for an object of sort $\begin{bmatrix} \textit{phrase} & \\ \text{SPR} & \langle\ \rangle \end{bmatrix}$
- We will use the symbol VP as an alias for XP $\begin{bmatrix} \textit{verb} & \\ \text{SUBJ} & \langle [\ ] \rangle \end{bmatrix}$

[9] Since *head-specifier-phrase* is a subsort of *headed-phrase*, the Head Feature Principle also applies.
[10] Pronounced "X-bar".
[11] Pronounced "X-double-bar".

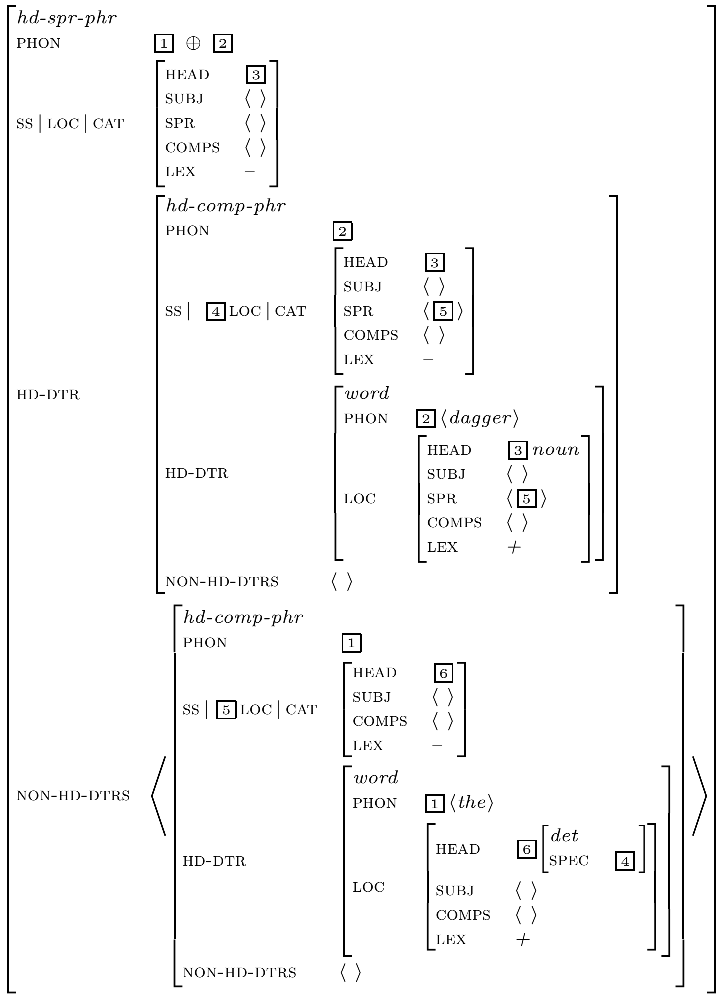


Figure 15: Partial description of the noun phrase *"the dagger"*.

- We will use the symbol S as an alias for XP $\begin{bmatrix} \textit{verb} \\ \text{SUBJ} \ \langle\ \rangle \end{bmatrix}$

- We will also replace X in the above by N, A, P, Det and so on, to abbreviate X $\begin{bmatrix}\text{HEAD} \ \text{NOUN}\end{bmatrix}$etc.

Additional features will be written as AVMs under the category node. We will also use NP, PP, DetP for the **maximal** (i.e. fully saturated) projection of the relevant lexical categories.[12] With these conventions, the NP *"the dagger"* can be written as in figure 16.[13]

[12]Note that VP is the exception to this convention. The maximal projection of a verb is written S.

[13]Note that representations like figure 16 involve an overloading of category aliases such as N′ and DetP. When these occur as the value of SPEC, SPR or other LOCAL features, they denote only the SYNSEM value of the category concerned. When they occur as a node in a tree (corresponding to the value of a DTRS feature), they denote a sign.

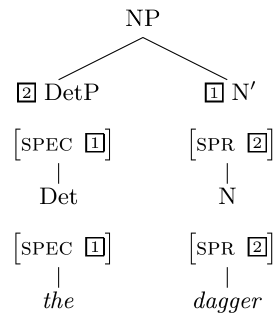


Figure 16: The NP *"the dagger"* in tree notation.

In the example that we have been discussing, the head noun *"dagger"* did not take any complements. Many nouns, however, do, as illustrated in the following examples (in which the complement is underlined).

(6) *"Your disapproval <u>of our plans</u>."*

(7) *"A book <u>about linguistics</u>."*

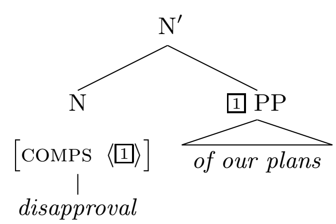
(8) *"Her delight <u>at winning</u>."*

(9) *"That photograph <u>of your brother</u>."*

All that is necessary to accommodate such cases is the inclusion in the grammar of the relevant lexical entries along the following lines (details omitted).

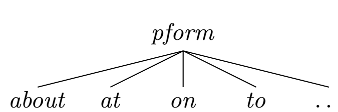
PHON	*disapproval*
SUBJ	⟨ ⟩
COMPS	⟨PP⟩

The definition of Head-Complement phrases and the other principles introduced in this section guarantee that the grammar will admit well-formed phrases of the following kind.



3.4 Prepositions and PPs

The analysis of prepositions proceeds in much the same way as that of the other categories we have discussed. They are distinguished from other categories by the value of their HEAD feature, which is *prep*. One feature of examples (6)-(9) immediately above is that there is a restriction on the preposition possible in the complement of any given head noun: *"disapproval"* and *"photograph"* require *"of"*, whereas *"delight"* requires *"at"*, and *"book"* requires *"about"* or *"on"*. In order to capture this relationship, we will specify that the sort *prep* is appropriate for a feature PFORM with value *pform* which forms the following sortal hierarchy.



Prepositions have lexical entries of the following kind, in which the the PHON value is paired with an appropriate PFORM value. [14]

$$\begin{bmatrix} \text{PHON} & \langle \textit{about} \rangle \\ \text{HEAD} & \begin{bmatrix} \textit{prep} & \\ \text{PFORM} & \textit{about} \end{bmatrix} \\ \text{SUBJ} & \langle\ \rangle \\ \text{COMPS} & \langle \text{NP}[\textit{acc}] \rangle \end{bmatrix}$$

Since PFORM is a HEAD feature, it and its value will be shared by its mother. To ensure the correct matching between noun and preposition in examples like (6)-(9), we only need to modify the lexical entry of the selecting noun so that it specifies the PFORM value of its PP complement, e.g.

$$\begin{bmatrix} \text{PHON} & \langle \textit{disapproval} \rangle \\ \text{SUBJ} & \langle\ \rangle \\ \text{COMPS} & \langle \text{PP}[\textit{of}] \rangle \end{bmatrix}$$

These additions give us phrases of the kind shown in figure 17, in which the tags $\boxed{1}$, $\boxed{6}$and $\boxed{7}$ indicate structure sharing of HEAD values between mother and daughter. The remaining tags indicate structure sharing between valence or SPEC values and DAUGHTERS values.

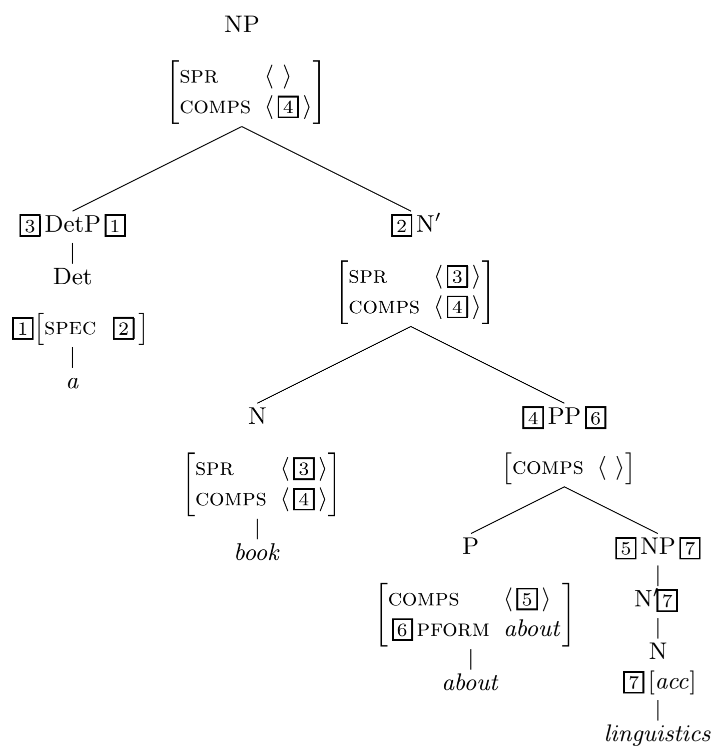


Figure 17: Tree representation for description of *"a book about linguistics"*.

Exercise 5 *Re-draw figure 17 in AVM notation.*

[14]Note, however, that the value of PHON and that of PFORM are not the same things. The value of PHON is an object of sort *phonstring*, for which '*about*' is only an conventional abbreviation.

3.5 Verbs and auxiliaries

We have already touched on some of the major characteristics of the HPSG analysis of verbs in our discussion of heads and valence. There is, however, quite a good deal more to be said about them. Our starting point will be the following set of examples which show verbs selecting VPs as their complements.

(10) *"Toby may drink scotch."*

(11) *"Toby is drinking scotch."*

(12) *"Toby has drunk scotch."*

We assume without discussion that sentences like these have the structure shown in figure 18.

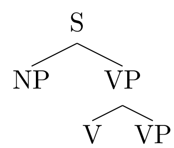


Figure 18: Constituent structure for auxiliary verbs.

There are a number of interesting things to note about examples like these.

- Firstly, the verb DRINK appears in a range of different forms (*"drink"*, *"drinking"*, *"drunk"*),
- secondly, the appropriate form of DRINK in each example is determined by the preceding verb. For example, *"is"* (or any other form of the lexeme BE) requires *"drinking"*, any other choice is ungrammatical (* *"Toby is drink scotch"*, * *"Toby is drunk scotch"*), and,
- thirdly, *"Toby"* is the subject of the second verb as well as the first one. (Who is doing the drinking?)

English verbs can appear in a range of different forms, as illustrated in table 3 with the lexeme DRINK, each with a distinctive distribution. To distinguish between these different forms of the verb, we will introduce the attribute VFORM, with the value *vform*. The sort *vform* has the subsorts shown in figure 19, whose names are drawn from the labels in table 3.

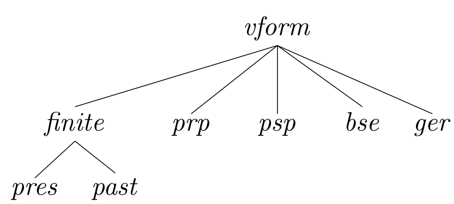


Figure 19: Sortal hierarchy for *vform*.

Finite	present tense	*"Toby drinks"*
	past tense	*"Toby drank"*
Present participle		*"Toby is drinking"*
Past participle		*"Toby has drunk"*
Base		*"Toby can drink"*
Gerund		*"Toby likes drinking"*

Table 3: Table of English verb forms

The attribute VFORM is appropriate for the *head* subsort *verb*, so every verbal category will contain the following feature structure.

$$\begin{bmatrix} \textit{cat} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{vform} \end{bmatrix} \end{bmatrix}$$

Verbs may possess the complete range of verbal forms (as DRINK does), or may be restricted to some subset of them. Among the first verbs in the sequences in (10)-(12), BE has all the forms shown for DRINK, whereas MAY lacks all except the finite forms. (There are no forms * *"maying"* or * *"mayed"*). However, BE, MAY and HAVE also possess grammatical characteristics that are not shared by other verbs such as DRINK. These are neatly summed up by the acronym NICE.

Negation: *"Toby isn't tall."*

Inversion: *"Is Toby tall?"*

Contraction: *"Toby's tall."*

Ellipsis: *"People say Toby is tall, and he is."*

Negation means that the verb has a distinct negative form, usually represented orthographically with *n't*. **Inversion** refers to the capacity of the verb to precede the subject in interrogatives and some other sentence-types. **Contraction** means the verb has an alternative pronunciation which is shorter than its citation form – sometimes this is given orthographic recognition as with *'s* for *"is"* and *"has"*, sometimes not, as with the contracted version of *"can"* (rather like *"c'n"*). Finally, **ellipsis** means that a constituent which normally follows the verb may be omitted and its interpretation recovered from the context. Verbs that exhibit some or all of these properties are known as **auxiliary verbs**, or simply **auxiliaries**.

As illustrated by the inversion example, in contrast to main verbs, auxiliaries may precede or follow the subject - a phenomenon known as **subject-auxiliary inversion** (SAI). One further detail of SAI is that the form of the verb which occurs in pre-subject position may be different to the one that occurs in post-subject position.

(13) *"I am not happy."* / *"I'm not happy."*[15] /* *"I aren't happy."*

(14) *"Aren't I happy."*

"Aren't" can only co-occur with a first person singular subject if it precedes it, not if it follows it. These distinctions motivate the postulation of two additional HEAD features for verbs: AUX and INV. AUX is *boolean* valued and partitions the class of verbs into auxiliaries ([AUX +]) and non-auxiliaries ([AUX –]). [INV +] identifies those forms of auxiliaries that precede the subject, [INV –] those that follow it.

If we put these developments together, the value of HEAD for verbal projections looks like this.

$$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{vform} \\ \text{AUX} & \textit{bool} \\ \text{INV} & \textit{bool} \end{bmatrix} \end{bmatrix}$$

We stated above that auxiliaries select VPs as their complements. To be precise, a VP here means a LOCAL value which has a HEAD value of sort *verb*, an empty COMPS list and a non-empty SUBJ value. This means that what appears on the COMPS list of an auxiliary is a description of the following kind.

[15] Or *"I amn't happy."* in some varieties of English.

$$\begin{bmatrix} \textsc{head} & \begin{bmatrix} \textit{verb} \\ \textsc{inv} & - \end{bmatrix} \\ \textsc{subj} & \langle [\ ] \rangle \\ \textsc{comps} & \langle\ \rangle \end{bmatrix}$$

The various realisations of the lexeme BE are followed by VPs headed by a present participle, so their lexical entry will contain the information shown in figure 20, in which this information is augmented by the addition of the feature specification [VFORM *prp*].

$$\begin{bmatrix} \textsc{head} & \begin{bmatrix} \textit{verb} \\ \textsc{aux} & + \end{bmatrix} \\ \textsc{subj} & \boxed{1} \\ \textsc{comps} & \left\langle \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} \\ \text{VFORM} & \textit{prp} \\ \text{INV} & - \end{bmatrix} \\ \text{SUBJ} & \boxed{1} \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 20: Partial lexical entry for the auxiliary verb *"be"*.

The sentence *"Toby is drinking scotch"* will have the structure shown in figure 21.

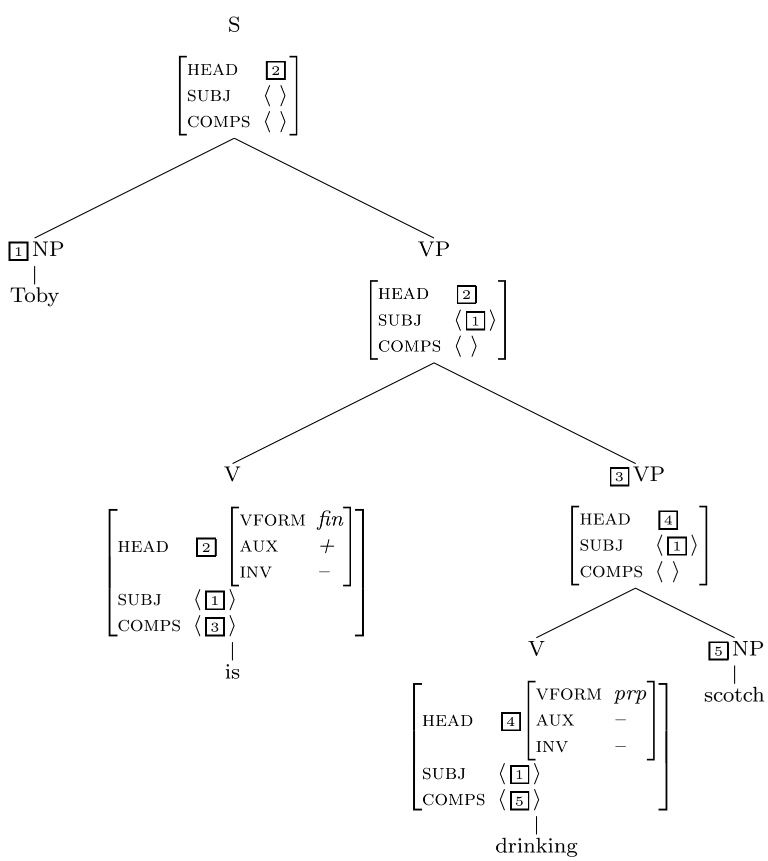


Figure 21: *"Toby is drinking scotch."*

MAY belongs to a large subclass of auxiliaries known as **modals**.[16] These require that their complement contains the *base* form of the verb and they exhibit the peculiarity (mentioned in respect of MAY above) that they are **defective** and possess only the *finite* form. (Which precludes them from following any other auxiliary, since no auxiliaries select a finite VP.) Their lexical entries therefore look like figure 22.

$$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{fin} \\ \text{AUX} & + \end{bmatrix} \\ \text{SUBJ} & \boxed{6} \\ \text{COMPS} & \left\langle \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{bse} \\ \text{INV} & - \end{bmatrix} \\ \text{SUBJ} & \boxed{6} \\ \text{COMPS} & \langle\,\rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 22: Partial lexical entry for a modal auxiliary.

HAVE, the other auxiliary introduced in (10)-(12), requires that its complement be in the past participle form, giving the lexical entry in figure 23.

$$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AUX} & + \end{bmatrix} \\ \text{SUBJ} & \boxed{6} \\ \text{COMPS} & \left\langle \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{psp} \\ \text{INV} & - \end{bmatrix} \\ \text{SUBJ} & \boxed{6} \\ \text{COMPS} & \langle\,\rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 23: Partial lexical entry for the auxiliary verb HAVE.

A potentially confusing property of HAVE is that it represents the pronunciation of more than one lexical category: both an auxiliary and a main verb. The version which subcategorises for a VP is an auxiliary and exhibits all the NICE properties.

- **Negation**: *"Toby hasn't drunk the scotch."*
- **Inversion**: *"Has Toby drunk the scotch?"*
- **Contraction**: *"Toby's drunk the scotch."*
- **Ellipsis**: *"People say Toby's drunk the scotch, and he has."*

The version that subcategorises for an NP, for many speakers, does not, and shares the distribution of main verbs like DRINK.[17]

[16] Other modals are *"can"*, *"could"*, *"will"*, *"would"*, *"shall"*, *"should"*, *"might"*, *"must"*.

[17] For some speakers (mainly British), this version of *"have"* can also behave as an auxiliary, giving

(i) *"Toby hasn't a book."*

(ii) *"Has Toby a book?"*

(15) (a) *"Toby has a book."*
(b) *"Toby drinks scotch."*

(16) (a) *"Toby doesn't have a book."*
(b) *"Toby doesn't drink scotch."*

(17) (a) *"Does Toby have a book."*
(b) *"Does Toby drink scotch."*

(18) (a) *"I don't have a book, but Toby does."*
(b) *"I don't drink scotch, but Toby does."*

With main verbs, in the NICE contexts, a 'dummy' auxiliary verb *"do"* is required. Auxiliary DO, like the modals, requires its complement to contain the *base* form of a verb. Like HAVE, DO leads a double life, as both auxiliary and main verb, giving rise to sentences in which both auxiliary and main verb DO co-occur, such as *"What did you do?"*

The auxiliaries HAVE and DO are both defective. Auxiliary HAVE lacks present and past participle forms, thus disallowing sentences like the following.

(19) * *"Toby is having drunk scotch."*

(20) * *"Toby has had drunk scotch."*

Auxiliary DO, like the modals, lacks all but the *finite* form and, in addition, requires VP complements which are [AUX –] (thus precluding sentences like * *"Toby doesn't have drunk the scotch"*).

The final auxiliary that we will discuss here may seem rather surprising. It is TO in sentences such as *"Toby wants to leave"*. This is a highly defective auxiliary (it lacks finite, contracted and negated forms), but it does allow ellipsis: *"Toby says he isn't leaving, but I'm sure he really wants to"*. TO requires us to posit an additional subsort of *vform*, *inf*, giving it the lexical entry shown in figure 24. The phrase *"wants to drink"* will have the analysis shown in figure 25, in which *"wants"* selects a VP[VFORM *inf*].

$$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{inf} \\ \text{AUX} & + \\ \text{INV} & - \end{bmatrix} \\ \text{SUBJ} & \boxed{6} \\ \text{COMPS} & \left\langle \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{bse} \\ \text{INV} & - \end{bmatrix} \\ \text{SUBJ} & \boxed{6} \\ \text{COMPS} & \langle\,\rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 24: Partial lexical entry for the auxiliary verb TO.

(iii) *"I haven't a book, but Toby has."*

For such speakers, *"have"* has three lexical entries:

1. an auxiliary selecting a VP complement
2. an auxiliary selecting a NP complement
3. a main verb selecting a NP complement.

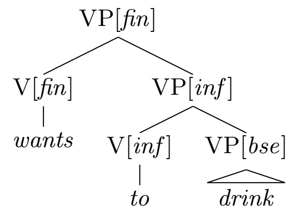


Figure 25: Tree for the phrase *"wants to drink"* showing VFORM values.

As a conclusion to this section, we observe that the lexical entries for auxiliaries that we have discussed here, together with the Head Feature Principle and the definition of *head-complement-phrase*, allow for sentences containing sequences of auxiliary verbs.

(21) *"Toby may be drinking scotch."*

(22) *"Toby may have been drinking scotch."*

(23) *"Toby has been drinking scotch."*

The flow of VFORM and valence information for (22) can be seen in the tree in figure 26.

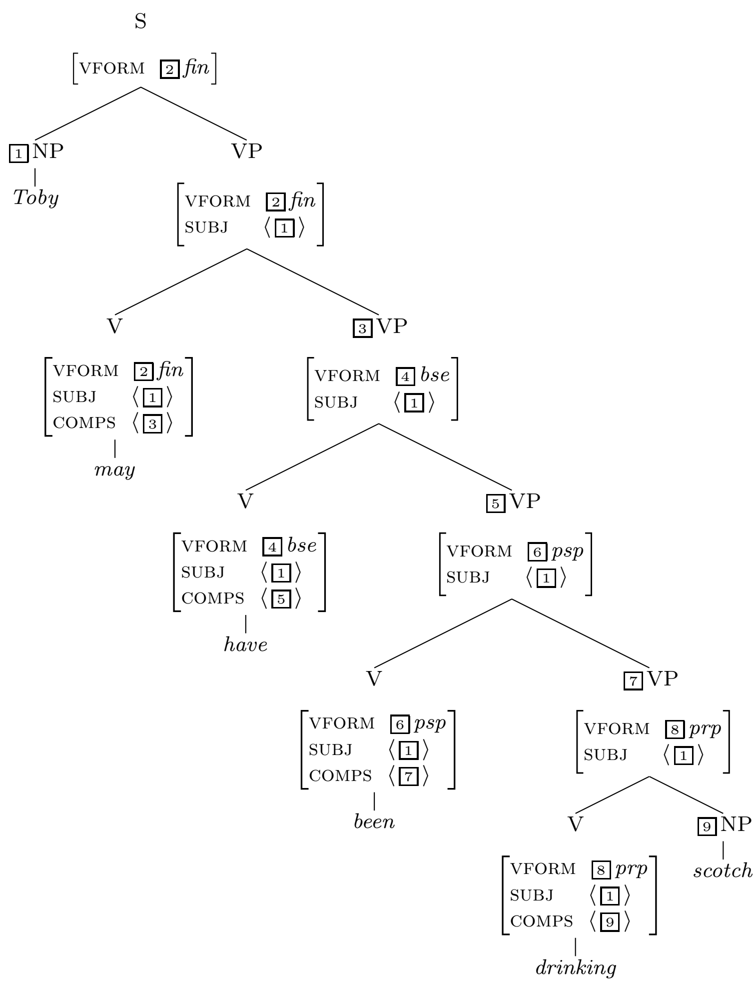


Figure 26: *"Toby may have been drinking scotch."*

Exercise 6 *Show how the analysis of English auxiliaries sketched out above predicts the ungrammaticality of the following sentences.*

1. **"Toby has done drink the scotch."*
2. **"Toby is having drunk the scotch."*
3. **"Toby did may drink the scotch."*

3.6 Clauses

It was stated in the previous section that clauses (i.e. sentences) are projections of verbs. Figure 26 shows how, as a consequence of this and the way the Head Feature Principle operates, the VFORM value of the clause will be shared with that of the the 'highest' verb in the clause. This means that it is straightforward to account for the following patterns of distribution.

(24) *"Andrew said Toby was drinking."*
(25) *"Andrew said that Toby was drinking."*
(26) *"That Toby was drinking surprised Andrew."*
(27) *"For Toby to be drinking is most unusual."*
(28) *"Andrew demanded that Toby stop drinking."*
(29) * *"Andrew said Toby be drinking."*
(30) * *"Andrew said (for) Toby to be drinking."*

(24)-(30) are all examples of sentences which contain a subordinate clause. In (24) the subordinate clause *"Toby was drinking"* is headed by a finite verb (*"was"*). Examples (29) and (30) show that the verb *"say"* cannot be followed by a subordinate clause headed by a *bse* (BE) or an *inf* verb (TO). Bearing in mind that the clause itself shares the value of VFORM with its head verb, this restriction is easily accounted for. All that is required is that the COMPS value of SAY is specified as being ⟨S[*fin*]⟩.[18]

Example (25) is very similar, but the subordinate clause is introduced by the **complementiser** *"that"*.[19] Complementisers are lexical items which select a clausal complement, forming a constituent called a **complementiser phrase** (CP). (See figure 27.) Many verbs which subcategorise for a clausal complement are, like *"say"*, indifferent as to whether it is S or CP. One way of accommodating this fact is to complicate the sortal hierarchy of heads slightly, by defining a new sort for complementisers, *comp*, and introducing a new sort *verbal* of which *comp* and *verb* are subsorts. The features VFORM, AUX and INV are appropriate for the new supersort and consequently are inherited by both subsorts,[20] as shown here.

[18] I.e.
$$\left\langle \begin{bmatrix} \textit{synsem} & \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{fin} \end{bmatrix} \\ \text{SUBJ} & \langle\,\rangle \\ \text{COMPS} & \langle\,\rangle \\ \text{LEX} & - \end{bmatrix} \right\rangle$$

[19] The term 'complementiser' has been in common use in linguistics for several decades. A more traditional term is **subordinating conjunction**.

[20] This analysis follows Sag, 1997, English relative clause constructions, Journal of Linguistics, 33:2, p.456f and not that of HPSG94, Chap. 3.

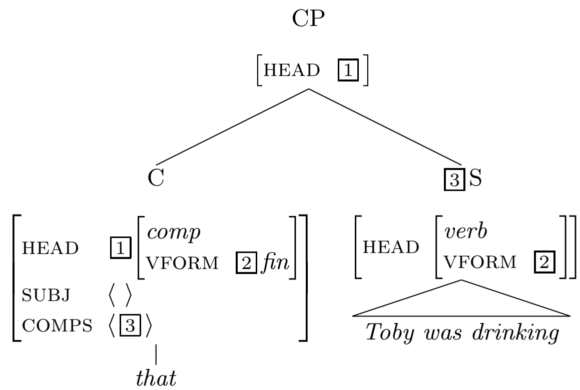
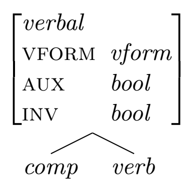


Figure 27: Tree for *"that Toby was drinking"*



Since *"that"* introduces finite clauses, it is defined in feature structure terms as follows.

$$\begin{bmatrix} \textit{word} & \\ \textsc{phon} & \langle \textit{that} \rangle \\ \textsc{head} & \begin{bmatrix} \textit{comp} & \\ \textsc{vform} & \boxed{10}\,\textit{fin} \end{bmatrix} \\ \textsc{subj} & \langle\ \rangle \\ \textsc{spr} & \langle\ \rangle \\ \textsc{comps} & \langle \mathrm{S}[\textsc{vform}\ \boxed{10}] \rangle \end{bmatrix}$$

With this lexical entry, the phrase *"that Toby was drinking"* is assigned the analysis in figure 27 as an instance of a head-complement structure. Verbs like *"say"* are defined as taking a complement specified as

$$\begin{bmatrix} \textsc{head} & \begin{bmatrix} \textit{verbal} & \\ \textsc{vform} & \textit{fin} \end{bmatrix} \\ \textsc{subj} & \langle\ \rangle \\ \textsc{comps} & \langle\ \rangle \end{bmatrix}$$

which subsumes both S and CP.

It requires only a slight modification to the lexical entry for the complementiser *"that"* to accommodate example (28) in which DEMAND selects a CP headed by a *bse* verb – changing the VFORM value from *fin* to *fin*∨*bse*, as shown below.

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{that} \rangle \\ \text{HEAD} & \begin{bmatrix} \textit{comp} & \\ \text{VFORM} & \boxed{2}\,\textit{fin} \vee \textit{bse} \end{bmatrix} \\ \text{SUBJ} & \langle\ \rangle \\ \text{COMPS} & \langle \text{S}[\text{VFORM}\ \boxed{2}] \rangle \end{bmatrix}$$

The contrast between the subcategorisation requirements of verbs like SAY and verbs like DEMAND is that the latter require a COMPS value which is ⟨CP[VFORM *bse*]⟩.

SURPRISE in sentence (26) differs from the previous examples in taking a finite CP as its subject. One of its subcategorisation requirements is therefore the following.

$$\begin{bmatrix} \text{SUBJ} & \langle \text{CP}[\textit{fin}] \rangle \\ \text{COMPS} & \langle \text{NP} \rangle \end{bmatrix}$$

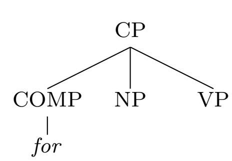
Example (27) is one in which the subject of *"is surprising"* is an infinitival clause (i.e. one whose head contains [VFORM *inf*]). Infinitival clauses take a different complementiser, *"for"*, whose syntax is defined as follows.[21]

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{for} \rangle \\ \text{HEAD} & \begin{bmatrix} \textit{comp} & \\ \text{VFORM} & \boxed{2}\,\textit{inf} \end{bmatrix} \\ \text{SUBJ} & \langle\ \rangle \\ \text{COMPS} & \left\langle \boxed{4}\,\text{NP},\ \text{VP}\begin{bmatrix} \text{VFORM} & \boxed{2} \\ \text{SUBJ} & \langle \boxed{4} \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

The SUBJ value of expressions like *"is most unusual"* is simply ⟨CP[*inf*]⟩.

Before we conclude this section, there is a further comment to be made about the relationship between VFORM values and the case of NPs. As noted in chapter**??** section**??**, pronouns in English exhibit differences in case marking, depending upon the syntactic position in which they occur. These differences are related to the kind of clause in which they occur: the subjects of finite clauses are *nominative*, other NPs are *accusative*. This can be handled by specifying that when a finite verb takes an NP subject, it specifies its SUBJ value as ⟨NP[*nom*]⟩. Any NP on a COMPS list, on the other hand, is specified as NP[*acc*]. The lexical entry for the finite transitive verb *"drinks"* will therefore contain the following.

[21] Note that the valence values of this complementiser are different to those of the complementiser *"that"*. *"That"* takes a single (sentential) complement, whereas *"for"* take a sequence of two complements, an NP and a VP. In tree terms, the constituent structure of non-finite clauses defined by this lexical entry for *"for"* is the following.



$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{drinks} \rangle \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{fin} \\ \text{AUX} & - \\ \text{INV} & - \end{bmatrix} \\ \text{SUBJ} & \langle \text{NP}[\textit{nom}] \rangle \\ \text{COMPS} & \langle \text{NP}[\textit{acc}] \rangle \end{bmatrix}$$

Exercise 7 *What value of* CASE *should be specified for the* SUBJ *values of auxiliary verbs? You should consider the following examples in making your decision and draw trees for the grammatical sentences, showing all the* SUBJ *values.*

1. *"I may have been wrong."*
2. **"Me may have been wrong."*
3. *"For me to be have been wrong would be unusual."*
4. **"For I to be have been wrong would be unusual."*

Exercise 8 *Draw AVMs showing the* SUBJ *and* COMPS *values for the underlined verbs in the following examples.*

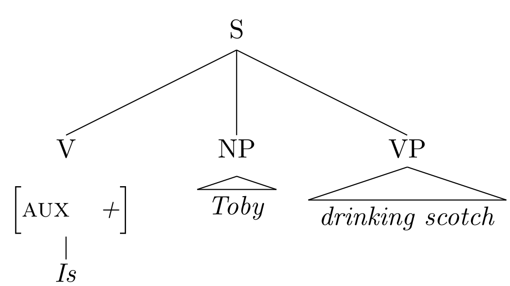
1. *"Toby <u>likes</u> scotch"*
2. *"Toby <u>gives</u> scotch to Andrew"*
3. *"A book <u>about</u> linguistics"*
4. *"Ophelia is <u>singing</u>"*

3.7 Subject-auxiliary Inversion (SAI)

The clauses that we have looked at so far have all taken the form NP VP. There are also English clauses in which an auxiliary verb precedes the subject:

(31) *"Is Toby drinking scotch."*
(32) *"What is Toby drinking."*
(33) *"Seldom did Toby drink scotch."*

We take the view that clauses like these simply have a flat structure in which the auxiliary, subject and post-subject constituent are sisters:



We need a way of providing for these constructions and do so by introducing a new type of clausal construction, *sai-ph*, a subsort of *hd-nexus-ph*, with the following constraint:

sai-phrase

$$
\textit{sai-ph} \Longrightarrow \begin{bmatrix} \text{SUBJ} & \langle\ \rangle \\ \text{HD-DTR} & \begin{bmatrix} \textit{word} & \\ \text{INV} & + \\ \text{AUX} & + \\ \text{SUBJ} & \langle \boxed{1} \rangle \\ \text{COMPS} & \boxed{2} \end{bmatrix} \\ \text{NON-HD-DTRS} & \langle \boxed{1} \rangle \oplus \boxed{2} \end{bmatrix}
$$

This constraint ensures that only verbs marked as [AUX +, INV +] will appear in such clauses and that the sister constituents will obey the constraints imposed by such verbs on their subject and complements.

Much more needs to be said about the semantics of these constructions and their syntactic distribution, but space considerations preclude a more detailed discussion.

4 The Lexicon

We have seen in the preceding sections of this chapter how HPSG represents linguistic information in terms of feature structure descriptions. Since HPSG is strongly lexical, the bulk of the information required by the grammar is encoded in lexical entries, as shown in a typical lexical entry (for the transitive verb LIKE) in figure 28. Now, this is the lexical entry for just a single English word.[22] Since every English other mono-transitive verb will require an almost identical lexical entry, it looks as if the lexicon will contain massive amounts of repetition. However, looked at from a different perspective, the fact that much of this information is shared with other verbs provides the opportunity to avoid unnecessary redundancy by organising lexical entries as an inheritance hierarchy. LIKE shares all the information in figure 28 except its pronunciation with every other mono-transitive verb; the value of HEAD is shared by every other verb, irrespective of valence; its value for SUBJ is shared by many other finite verbs, and so on. These observations lead to the conclusion that the lexicon can be structured into an inheritance hierarchy in which it is necessary to specify in the lexicon only the most idiosyncratic information for any given word.

$$
\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{like} \rangle \\ \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{fin} \\ \text{AUX} & - \\ \text{INV} & - \end{bmatrix} \\ \text{LEX} & + \\ \text{SUBJ} & \boxed{3}\langle \text{NP} \rangle \\ \text{SPR} & \boxed{4}\langle\ \rangle \\ \text{COMPS} & \boxed{5}\langle \text{NP}[\textit{acc}] \rangle \\ \text{ARG-STR} & \boxed{3} \oplus \boxed{4} \oplus \boxed{5} \end{bmatrix} \end{bmatrix}
$$

Figure 28: Lexical entry for LIKE.

Let us pursue this idea by first of all looking at parts of speech (i.e. noun, verb, etc.), and set up a sortal hierarchy of lexical types, i.e. partitions of the sort *word*. This is shown in table 4. Each

[22] It is also incomplete. We have yet to discuss the NON-LOCAL and CONTENT attributes.

sort is an immediate subsort of the entry given in its ISA column. The CONSTRAINTS column specifies those features and values appropriate for the sort in the left-hand column. Each subsort inherits the constraints of its supersorts. This hierarchy is also shown in graphical form in figure 29, with the constraint information omitted.

SORT	CONSTRAINT	ISA
sign	$\begin{bmatrix} \text{PHON} & \textit{list(phonstring)} \\ \text{SYNSEM} & \begin{bmatrix} \textit{synsem} & \\ \text{LOCAL} & \begin{bmatrix} \textit{local} & \\ \text{CAT} & \begin{bmatrix} \textit{cat} & \\ \text{SUBJ} & \textit{list(synsem)} \\ \text{SPR} & \textit{list(synsem)} \\ \text{COMPS} & \textit{list(synsem)} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$	$\top$
word	$\begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{LEX} & + \\ \text{SUBJ} & \boxed{1} \\ \text{SPR} & \boxed{2} \\ \text{COMPS} & \boxed{3} \\ \text{ARG-STR} & \boxed{1} \oplus \boxed{2} \oplus \boxed{3} \end{bmatrix} \end{bmatrix}$	*sign*
verb-wd	$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{vform} \\ \text{AUX} & \textit{bool} \\ \text{INV} & \textit{bool} \end{bmatrix} \\ \text{SUBJ} & \langle \textit{synsem} \rangle \\ \text{SPR} & \langle\ \rangle \end{bmatrix}$	*word*
main-verb-wd	$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AUX} & - \\ \text{INV} & - \end{bmatrix} \end{bmatrix}$	*verb-wd*
aux-verb-wd	$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{AUX} & + \end{bmatrix} \\ \text{ARG-STR} & \langle \textit{synsem}, \text{VP} \rangle \end{bmatrix}$	*verb-wd*
noun-wd	$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{noun} & \\ \text{CASE} & \textit{case} \end{bmatrix} \end{bmatrix}$	*word*
prep-wd	$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{prep} & \\ \text{PFORM} & \textit{pform} \end{bmatrix} \end{bmatrix}$	*word*

Table 4: Hierarchy of lexical parts of speech

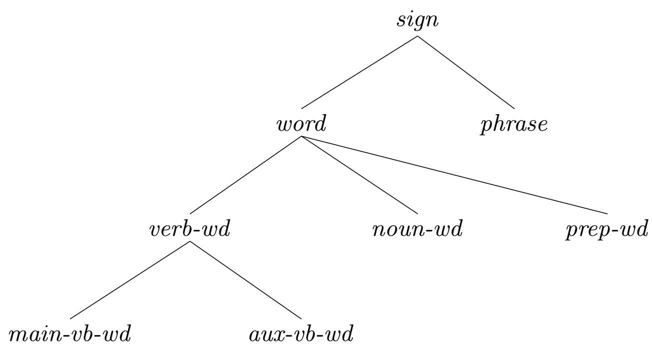


Figure 29: Part of speech hierarchy.

With the definitions in this sortal hierarchy, we can simplify the lexical entry for *"like"* to that below. All the other information specified in figure 28 is inherited from the sort *main-verb-wd*.[23]

$$\begin{bmatrix} \textit{main-vb-wd} & \\ \text{PHON} & \langle \textit{like} \rangle \\ \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{vform} \\ \text{AUX} & - \\ \text{INV} & - \end{bmatrix} \\ \text{ARG-STR} & \langle \text{NP, NP} \rangle \end{bmatrix} \end{bmatrix}$$

Let us now turn our attention to valence and show that a similar inheritance hierarchy is possible. We will consider separately the set of possibilities in subject position and the set of possible complements. We start by drawing a distinction between words which require a subject (such as verbs) and those which do not (such as prepositions), assigning the former to the sort *predicator-wd* and the latter to *non-predicator-wd*.

The sort *predicator-wd* can be subdivided into those words which require an NP subject (*np-predicator-wd*) and those that require a CP subject (*cp-predicator-wd*). The sort *cp-predicator-wd* can be further partitioned into words requiring a finite CP(*finite-cp-predicator-wd*) or an infinitival CP (*infinitival-cp-predicator-wd*).[24] This hierarchy and the constraints associated with the various subsorts are listed in table 5 and shown in diagrammatic form in figure 30.

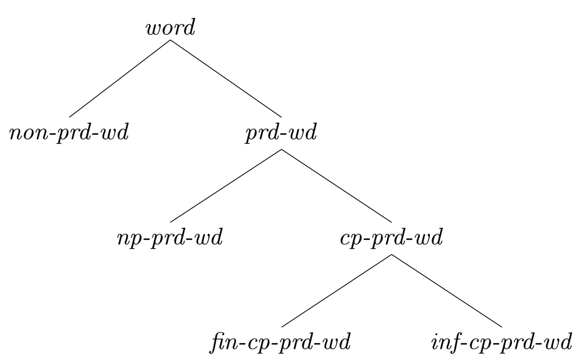


Figure 30: Subject valence hierarchy diagram.

[23]We will suffix all subsorts of *word* with '*-wd*', to prevent confusion. Thus *verb-wd* is a subsort of *word*, while *verb* is a subsort of *head*.

[24]There are further possibilities: infinitival VP subjects, gerundive subjects etc. We will not embark on an exhaustive treatment.

SORT	CONSTRAINT	ISA
non-predicator-wd	$\left[\text{SUBJ} \quad \langle\ \rangle\right]$	*word*
predicator-wd	$\left[\text{SUBJ} \quad \langle \textit{synsem} \rangle\right]$	*word*
np-predicator-wd	$\left[\text{SUBJ} \quad \langle \text{NP} \rangle\right]$	*prd-wd*
cp-predicator-wd	$\left[\text{SUBJ} \quad \langle \text{CP} \rangle\right]$	*prd-wd*
finite-cp-prd-wd	$\left[\text{SUBJ} \quad \langle \text{CP}[\textit{fin}] \rangle\right]$	*cp-prd-wd*
infinitival-cp-prd-wd	$\left[\text{SUBJ} \quad \langle \text{CP}[\textit{inf}] \rangle\right]$	*cp-prd-wd*

Table 5: Subject valence hierarchy.

Turning next to complements, the primary distinction is between those words which take a direct object NP and those which don't. The former are partioned under the subsort *transitive-wd*, which can be further partitioned into those words which permit only a single complement (the sort *mono-transitive-wd*) and those that require two complements *poly-transitive-wd*. The latter can be further partitioned into those whose second complement is an NP (*ditransitive-wd*), those whose second complement is a PP (*to-transitive-wd*) and those whose second complement is an CP or S. Those words which do not take a direct object NP are grouped under the sort *intransitive-wd*), which is further subdivided into those permitting no additional complements at all (*strict-intransitive-wd* and those taking an unsaturated complement (*intransitive-xcomp-wd*). The only instance of the latter that we have encountered so far are the auxiliaries which take a VP complement and also share their SUBJ value with it. This is represented by the subsort *ssr-wd* (for 'subject-to-subject-raising' – a term by which this kind of construction is referred to by linguists). This classification is summarised in table 6 and in figure 31.

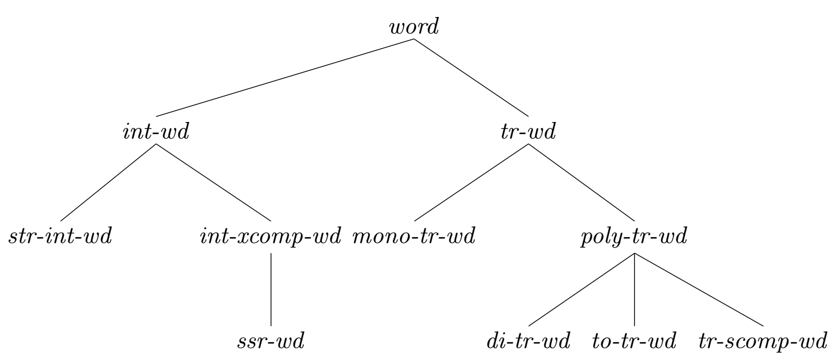


Figure 31: Complement valence hierarchy diagram.

SORT	CONSTRAINT	ISA
intran-wd		*word*
strict-intran-wd	$\left[\text{ARG-STR}\ \left[\text{FIRST}\ \ \textit{synsem}\right]\right]$	*intran-wd*
intran-xcomp-wd	$\left[\text{ARG-STR}\ \left[\text{REST}\ \left[\text{FIRST}\ \ \text{XP}[\text{SUBJ}\ \langle \textit{synsem}\rangle]\right]\right]\right]$	*intran-wd*
ssr-wd	$\left[\text{ARG-STR}\ \ \langle \boxed{1}, \begin{bmatrix}\text{SUBJ} & \boxed{1}\\ \text{COMPS} & \langle\ \rangle\end{bmatrix}\rangle\right]$	*intran-xcomp-wd*
trans-wd	$\left[\text{ARG-STR}\ \ \langle \textit{synsem},\ \text{NP}[\textit{acc}]\rangle\right]$	*word*
trans-wd	$\left[\text{ARG-STR}\ \left[\text{REST}\ \left[\text{FIRST}\ \text{NP}[\textit{acc}]\right]\right]\right]$	*word*
mono-trans-wd	$\left[\text{ARG-STR}\ \left[\text{REST}\ \left[\text{REST}\ \langle\ \rangle\right]\right]\right]$	*trans-wd*
poly-trans-wd	$\left[\text{ARG-STR}\ \left[\text{REST}\ \left[\text{REST}\ \ \textit{nelist(synsem)}\right]\right]\right]$	*trans-wd*
di-trans-wd	$\left[\text{ARG-STR}\ \left[\text{REST}\ \left[\text{REST}\ \ \langle\ \text{NP}[\textit{acc}]\rangle\right]\right]\right]$	*poly-trans-wd*
to-trans-wd	$\left[\text{ARG-STR}\ \left[\text{REST}\ \left[\text{REST}\ \ \langle \text{PP}[\textit{to}]\rangle\right]\right]\right]$	*poly-trans-wd*
trans-scomp-wd	$\left[\text{ARG-STR}\ \left[\text{REST}\ \left[\text{REST}\ \ \langle \text{XP}[\textit{verbal}]\rangle\right]\right]\right]$	*poly-trans-wd*

Table 6: Complement valence hierarchy

The part of speech hierarchy and the valence hierarchies classify words along three different **dimensions** and a given token of a word can possess properties from more than one of them, e.g. it can be both a verb and intransitive or take both an NP subject and an NP direct object. Figure 32 shows the place of a mono-transitive verb such as *"like"* in terms of the different dimensions of the *word* sort. The labels for the different dimensions are placed in boxes to indicate that they are not themselves sorts and are not disjoint partitions of *word*. The lowest subsort *np-predicator-mono-transitive-wd* inherits from all three dimensions, POS (Part of Speech), SUBJ-VAL (Subject Valence) and COMP-VAL (Complement Valence), a phenomenon known as **multiple inheritance**.[25] This allows us to further simplify the lexical entry for *"likes"*, to that shown below. The sort *np-predicator-mono-transitive-wd* is the meet of three sorts (*main-vb-wd*∧*np-predicator-wd*∧ *mono-transitive-wd*).

[25] We are here making an open world assumption – any conjunction of sorts is well-formed unless explicitly excluded.

$$\begin{bmatrix} \textit{np-prd-mono-tr-wd} & \\ \text{PHON} & \langle \textit{likes} \rangle \\ \text{HEAD} & \begin{bmatrix} \text{VFORM} & \textit{fin} \end{bmatrix} \end{bmatrix}$$

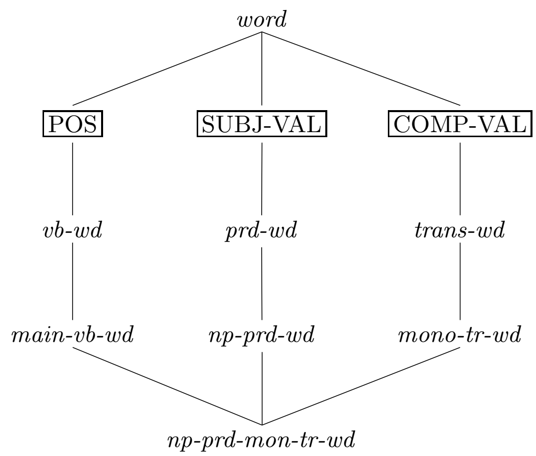


Figure 32: The place of *"like"* in the lexical hierarchy.

Ditransitive verbs such as *"give"* which take two NP complements, are assigned to the sort *np-predicator-di-transitive-wd*, defined as inheriting from *main-verb-wd*, *np-predicator-wd* and *di-transitive-wd*. The lexical entry for *"gives"* is simply the following.

$$\begin{bmatrix} \textit{np-prd-di-tr-wd} & \\ \text{PHON} & \langle \textit{gives} \rangle \\ \text{HEAD} & \begin{bmatrix} \text{VFORM} & \textit{fin} \end{bmatrix} \end{bmatrix}$$

SORT	CONSTRAINT	ISA
inf-wd	$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{VFORM} & \textit{inf} \end{bmatrix} \end{bmatrix}$	*verb-wd*
fin-wd	$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{VFORM} & \textit{fin} \end{bmatrix} \end{bmatrix}$	*verb-wd*
bse-wd	$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{VFORM} & \textit{bse} \end{bmatrix} \end{bmatrix}$	*verb-wd*
prp-wd	$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{VFORM} & \textit{prp} \end{bmatrix} \end{bmatrix}$	*verb-wd*
psp-wd	$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{VFORM} & \textit{psp} \end{bmatrix} \end{bmatrix}$	*verb-wd*

Table 7: Lexical hierarch of VFORM values.

We have still to draw a distinction between the various values of VFORM that a verb can contain. We can do this by recognising a dimension of *verb-wd* which classifies the range of possibilities. This is shown in table 7. Recall that the distinction between main and auxiliary verbs is also a partition of the sort *verb-wd*. We now have two different dimension in terms of which *verb-wd* is subclassified, VFORM and AUX/MAIN. Multiple inheritance from these two dimensions allows us to define sorts such as, *finite-main-verb-wd* as the meet of the sorts *main-verb-wd* and *finite-wd*. The unification of this sort with the valence dimensions allows the definition of sorts like *finite-np-predicator-mono-transitive-verb-wd* in figure 33.

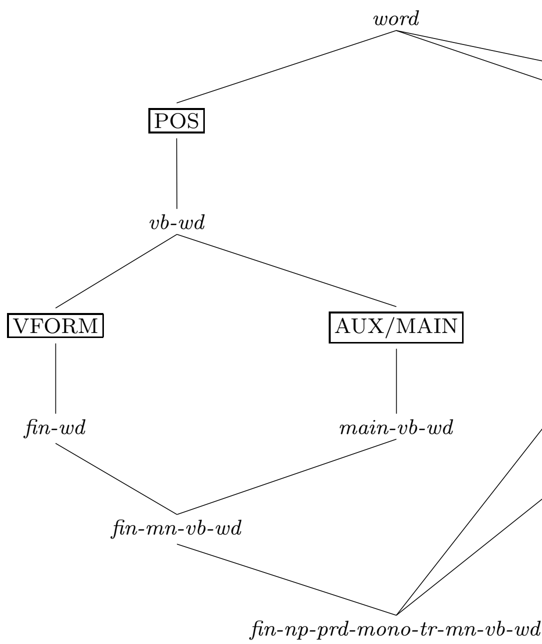


Figure 33: The place of finite mono-transitive verbs in the inheritance hierarchy

This allows us to provide the following lexical entry for a verb such as *"likes"*.

$$\begin{bmatrix} \textit{fin-np–mono-tr-vb-wd} \\ \textsc{phon} \quad \langle \textit{likes} \rangle \end{bmatrix}$$

To conclude this section, we will present a classification of the auxiliaries, i.e. subsorts of *aux-verb-wd*. This is shown in table 8.

SORT	CONSTRAINT	ISA	INSTANCE
perf-cmp-aux-vb-wd	$[\textsc{comps} \quad \langle \text{VP}[\textit{psp}] \rangle]$	*aux-vb-wd*	*"have"*
prog-cmp-aux-vb-wd	$[\textsc{comps} \quad \langle \text{VP}[\textit{prp}] \rangle]$	*aux-vb-wd*	*"be"*
bse-cmp-aux-vb-wd	$[\textsc{comps} \quad \langle \text{VP}[\textit{bse}] \rangle]$	*aux-vb-wd*	*"may"*
do-aux-vb-wd	$\left[\textsc{comps} \ \langle \text{VP}\begin{bmatrix} \textsc{vform} & \textit{bse} \\ \textsc{aux} & - \end{bmatrix} \rangle\right]$	*bse-cmp-aux-vb-wd*	*"do"*

Table 8: Auxiliary verb hierarchy

The constraints specify the VFORM values of the VP complements which these verbs select (discussed in section 3.5). We can define the lexical entry of, for example, the auxiliary *"has"* as a sort *has-wd* which inherits from the sorts *perf-comp-aux-vb-wd*, *fin-wd*, *prd-wd* and *ssr-wd*, giving simply the following.

$$\begin{bmatrix} \textit{has-wd} & \\ \text{PHON} & \langle \textit{has} \rangle \\ \text{VFORM} & \textit{pres} \end{bmatrix}$$

This requires the feature structure corresponding to *"has"* to satisfy the following cascade of constraints. To be a *has-wd* it must be a *perf-comp-aux-vb-wd.* Therefore,

- It must be a *word* (table 4):

$$\begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{LEX} & + \\ \text{SUBJ} & \boxed{1}\,\langle \textit{synsem} \rangle \\ \text{SPR} & \boxed{2}\,\textit{list(synsem)} \\ \text{COMPS} & \boxed{3}\,\textit{list(synsem)} \\ \text{ARG-STR} & \boxed{1} \oplus \boxed{2} \oplus \boxed{3} \end{bmatrix} \end{bmatrix}$$

- It must be a *verb-wd* (table 4):

$$\begin{bmatrix} \textit{vb-wd} & \\ \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{vform} \\ \text{AUX} & \textit{bool} \\ \text{INV} & \textit{bool} \end{bmatrix} \end{bmatrix} \\ \text{SPR} & \langle\ \rangle \end{bmatrix}$$

- It must be an *aux-verb-wd* (table 4):

$$\begin{bmatrix} \textit{aux-verb-wd} & \\ \text{CAT} & \begin{bmatrix} \text{AUX} & + \end{bmatrix} \end{bmatrix}$$

- It must be a *perf-aux-verb-wd* (table 4):

$$\begin{bmatrix} \text{COMPS} & \langle \text{VP}[\textit{psp}] \rangle \end{bmatrix}$$

In addition, it must unify with *fin-wd* which entails that it satisfy the following pair of constraints.

- It must be a *verb-wd* (table 7).
- It must be a *fin-wd* (table 7):

$$\begin{bmatrix} \textit{fin-wd} & \\ \text{HEAD} & \begin{bmatrix} \text{VFORM} & \textit{fin} \end{bmatrix} \end{bmatrix}$$

It must also unify with *prd-wd* which requires that it satisfy the following constraint (table 5).

- $\begin{bmatrix} \textit{prd-wd} & \\ \text{SUBJ} & \langle \textit{synsem} \rangle \end{bmatrix}$

It must unify with *ssr-wd* which entails that it satisfy in addition the following cascade of constraints.

- It must be an *intran-xcomp-wd* (table 6).

$$\begin{bmatrix} \textit{intran-xcomp-wd} & \\ \textsc{comps} & \langle \mathrm{XP}[\textsc{subj}\ \langle \textit{synsem} \rangle] \rangle \end{bmatrix}$$

- It must be an *ssr-wd* (table 6).

$$\begin{bmatrix} \textit{ssr-wd} & \\ \textsc{subj} & \boxed{1} \\ \textsc{comps} & \begin{bmatrix} \textsc{subj} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

And, finally,

- it must satisfy the constraints stipulated in its lexical entry:

$$\begin{bmatrix} \textsc{phon} & \langle \textit{has} \rangle \\ \textsc{vform} & \textit{pres} \end{bmatrix}$$

$$\begin{bmatrix} \textit{word} & \\ \textsc{phon} & \langle \textit{has} \rangle \\ \textsc{cat} & \begin{bmatrix} \textsc{head} & \begin{bmatrix} \textit{verb} & \\ \textsc{vform} & \textit{fin} \\ \textsc{aux} & + \\ \textsc{inv} & \textit{bool} \end{bmatrix} \\ \textsc{lex} & + \\ \textsc{subj} & \boxed{1} \\ \textsc{spr} & \boxed{2}\langle\ \rangle \\ \textsc{comps} & \boxed{3}\langle \mathrm{VP}\begin{bmatrix} \textsc{subj} & \boxed{1}\langle \textit{synsem} \rangle \end{bmatrix} \rangle \\ \textsc{arg-str} & \boxed{1} \oplus \boxed{2} \oplus \boxed{3} \end{bmatrix} \end{bmatrix}$$

Figure 34: AVM for auxiliary *"has"*.

The unification of all of these constraints results in the description in figure 34.

For a account of the lexicon of English, further elaboration of the system of lexical sorts is necessary, but we hope to have shown in this section how much of the redundancy in the specification of lexical entries can be eliminated. This concludes our discussion of LOCAL values. In the next section we turn to the NON-LOCAL attribute.

Exercise 9

Go through the stages in inheritance of the properties of the feature structure for auxiliary "has", enumerated above, and draw an AVM for each step showing how the initial supersort is added to.

Exercise 10

Provide lexical entries for the following,

1. *the modal auxiliary "may", and*
2. *the infinitival auxiliary "to".*

Exercise 11

Show how to avoid the necessity of specifying in lexical entries that the subject of a finite verb must be nominative.

Exercise 12

Simplify the following lexical entries by defining appropriate sorts based on those given in the text. Explain how the required information is inherited from the lexical entry's supersorts by drawing diagrams like figure 32. Make use of your answers to the preceding exercise.

1. $$\begin{bmatrix} \text{PHON} & \langle \textit{laughs} \rangle \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{fin} \\ \text{AUX} & - \\ \text{INV} & - \end{bmatrix} \\ \text{SUBJ} & \langle \text{NP}[\textit{nom}] \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{LEX} & + \end{bmatrix}$$

2. $$\begin{bmatrix} \text{PHON} & \langle \textit{gave} \rangle \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{past} \\ \text{AUX} & - \\ \text{INV} & - \end{bmatrix} \\ \text{SUBJ} & \langle \text{NP}[\textit{nom}] \rangle \\ \text{COMPS} & \langle \text{NP}[\textit{acc}], \text{PP}[\textit{to}] \rangle \\ \text{LEX} & + \end{bmatrix}$$

3. $$\begin{bmatrix} \text{PHON} & \langle \textit{aren't} \rangle \\ \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{VFORM} & \textit{pres} \\ \text{AUX} & + \\ \text{INV} & + \\ \text{NEG} & + \end{bmatrix} \\ \text{SUBJ} & \boxed{1} \\ \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \left\langle \text{VP}\begin{bmatrix} \text{SUBJ} & \boxed{1} \\ \text{VFORM} & \textit{prp} \end{bmatrix} \right\rangle \\ \text{LEX} & + \end{bmatrix}$$

Exercise 13

Provide full lexical entries for the underlined words in the following phrases. Show how to simplify your entries by using multiple inheritance, positing whatever additional sorts you require.

1. *"bought a book from Helena."*
2. *"said that she was happy."*

3. *"told me that Macbeth is dead."*
4. *"a book about drama."*

5 Lexical Relations

Consider the following pair of sentences.

(34) *"Andrew gave Toby scotch"*

(35) *"Toby was given scotch by Andrew"*

There are obvious systematic syntactic and semantic relationships between such pairs.

- They have the same truth conditions
- The subject of (34) appears as the object of the preposition *"by"* in (35)
- The direct object of (34) appears as the subject in (35)
- The verb of (34) appears in a related form in (35) and is preceded by a form of the auxiliary verb *"be"*

Sentences like (34) are called **active** sentences and those like (35) are called **passive** sentences, and any descriptively adequate account of English grammmar should explicitly recognise the existence of such systematic correspondences. A long-standing way of doing this is to take one of the sentences as being more basic (typically the active one) and to map it onto the other. More recently, in highly lexicalised frameworks, such as HPSG, it is argued that the relationship can be captured in terms of a relationship between the active and passive forms of verbs (e.g. *"give"* and *"given"*).

If we look at AVMs for the active and passive forms of the verb *"give"*, shown in figure 35, it is easy to see the general nature of the relationship. From a comparison of the two AVMs, it is clear that, although they differ in a number of respects, quite a lot of information is also shared between the two forms.

There are a number of different ways in HPSG of treating this kind of relationship. The most traditional approach is is to map one form onto the other, preserving the similarities and that is what we will sketch out here. This kind of mapping is called a **lexical rule**. Lexical rules are statements of the form "if a word of form A (the 'input' to the rule) exists in the lexicon, then a word of form B (the 'output' of the rule) is also in the lexicon", where B is the result of applying some function to A.

For example, a lexical rule for passive verbs might look like figure 36.

$$\begin{bmatrix} \multicolumn{2}{l}{\textit{trans-mn-vb-wd}} \\ \text{PHON} & \langle \boxed{1} \rangle \\ \text{HEAD} & \begin{bmatrix} \text{VFORM} & \textit{bse} \end{bmatrix} \\ \text{ARG-STR} & \langle \boxed{2}_{\boxed{i}}, \boxed{3} \rangle \oplus \boxed{4} \\ \text{CONT} & \boxed{5} \end{bmatrix} \Longrightarrow \begin{bmatrix} \multicolumn{2}{l}{\textit{pass-mn-vb-wd}} \\ \text{PHON} & \langle f_{pass}(\boxed{1}) \rangle \\ \text{HEAD} & \begin{bmatrix} \text{VFORM} & \textit{pas} \end{bmatrix} \\ \text{ARG-STR} & \langle \boxed{3} \rangle \oplus \boxed{4} \oplus (\langle \text{PP}[\textit{by}]_{\boxed{i}} \rangle) \\ \text{CONT} & \boxed{5} \end{bmatrix}$$

Figure 36: Passive lexical rule.

The essence of this mapping lies in the difference between the two ARG-STR values. The initial item of the 'input' is dropped and its index transferred to a PP appended to the end of the original

active	$\begin{bmatrix} \text{PHON} & \langle \textit{give} \rangle \\ \text{HEAD} & \begin{bmatrix}\text{VFORM} & \textit{bse}\end{bmatrix} \\ \text{ARG-STR} & \langle \boxed{1}\text{NP}_{\boxed{i}}, \boxed{2}\text{NP}_{\boxed{j}}, \boxed{3}\text{PP}[\textit{to}]_{\boxed{k}} \rangle \\ \text{SUBJ} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle \boxed{2}, \boxed{3} \rangle \\ \text{CONT} & \begin{bmatrix} \text{QUANTS} & \langle\ \rangle \\ \text{NUCL} & \begin{bmatrix} \textit{give-rel} & \\ \text{GIVER} & \boxed{i} \\ \text{GIFT} & \boxed{j} \\ \text{RECIPIENT} & \boxed{k} \end{bmatrix} \end{bmatrix} \end{bmatrix}$
passive	$\begin{bmatrix} \text{PHON} & \langle \textit{given} \rangle \\ \text{HEAD} & \begin{bmatrix}\text{VFORM} & \textit{pas}\end{bmatrix} \\ \text{ARG-STR} & \langle \boxed{1}\text{NP}_{\boxed{j}}, \boxed{2}\text{PP}[\textit{to}]_{\boxed{k}}, \boxed{3}\text{PP}[\textit{by}]_{\boxed{i}} \rangle \\ \text{SUBJ} & \langle \boxed{1} \rangle \\ \text{COMPS} & \langle \boxed{2}, \boxed{3} \rangle \\ \text{CONT} & \begin{bmatrix} \text{QUANTS} & \langle\ \rangle \\ \text{NUCL} & \begin{bmatrix} \textit{give-rel} & \\ \text{GIVER} & \boxed{i} \\ \text{GIFT} & \boxed{j} \\ \text{RECIPIENT} & \boxed{k} \end{bmatrix} \end{bmatrix} \end{bmatrix}$

Figure 35: Lexical entries for the active and passive forms of the verb *"give"*.

argument list.[26] The second NP on the orginal argument list now becomes the first item and, from the definition of the sort *word* given in table 4 on page 33, the first item on the argument list is structure-shared with the SUBJ value, so the direct object in the input is mapped onto the subject in the output. f_{pass} represents a morphological operation which maps the phonological representation of the *base* form of the verb onto the corresponding passive form (i.e. f_{pass}*(give)=given*). The stipulation that the 'output' is of sort *pass-wd* entails that its VFORM value is *pas(ive)*. It is assumed that any other values not explicitly changed by the lexical rule are transferred unchanged from input to output.[27]

The derived lexical entry for passive verbs, together with other constraints of the grammar of English, will give us passive verb phrases containing passive verbs and there complements, but we also need some way of allowing passive VPs to form sentences. Passive VPs can appear in a range of constructions.

(1) *"Duncan was killed by Macbeth"*

(2) *"Painted by Leonardo, the Mona Lisa is one of the most famous paintings in the world"*

(3) *"I'm looking for a book written by a linguist"*

[26] Parentheses round the PP are intended to indicate that the constituent is optional, allowing 'agentless passives' like *"Duncan was killed"*.

[27] So, strictly speaking, the inclusion of CONTENT in figure 36 is redundant.

Here will limit ourselves to sentential passives like (1). These are introduced by the auxiliary BE. We already have a lexical entry for BE (figure 20 on 24), but that entry specifies that its complement must be a verb phrase bearing the specification [VFORM *prp*]. It would, of course, be possibly to simply add another lexical entry, identical to figure 20, except for the specification [VFORM *prp*], but a further look at BE suggests that this would not be a good move.

BE is not restricted to taking the two sorts of complements (VP[VFORM *prp*] and VP[VFORM *pas*]) that we have encountered so far.

(4) *"Macbeth is in Glamis."* (PP)

(5) *"Duncan was (the) king of Scotland."* (NP)

(6) *"Toby is fond of scotch."* (AP)

To accommodate these additional examples, we would need a proliferation of BEs. Items that can occur as complements of BE are traditionally known as 'predicative complements', which suggests that we could capture their distribution in terms of a boolean-valued head feature PRED. All the kinds of constituents that can appears as complements to BE are [PRED +], those that cannot are [PRED -]. Passive and progressive verbs are [PRED +], other forms of verbs (finite, past participle) are [PRED -].

The lexical entry for BE is modified that that in figure 37

$$\begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{AUX} & + \end{bmatrix} \\ \text{SUBJ} & \boxed{6} \\ \text{COMPS} & \left\langle \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{verb} & \\ \text{PRED} & + \\ & \end{bmatrix} \\ \text{SUBJ} & \boxed{6} \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \right\rangle \end{bmatrix}$$

Figure 37: Revised lexical entry for the auxiliary verb *"be"*.

6 Unbounded dependency constructions

We mentioned in section 1 the existence of constructions exemplified by sentences like (36), in which a constituent (in this case the NP *"ice cream"*) is displaced from its normal position (after the verb *"likes"*).

(36) *"Ice cream, I like."*

Such constructions are known as **unbounded dependency constructions** (UDCs), and have the following characteristics.[28]

- there is a dependency between the displaced constituent, the 'filler,' and its 'original' position, the 'gap' – if one of them is filled, the other must be empty

28 They are also known as ***wh*-constructions**, because a significant subset of them involve the displacement of phrases containing a ***wh*-word** (i.e. one whose spelling begins with the letters *"wh"*), as in examples (37)-(40) in the text below. The phenomenon of displacement is also called ***wh*-movement**. It is also known as $\bar{A}$-movement, since it involves 'movement' to a non-argument position.

- the distance which can intervene between the filler and the gap is potentially unbounded, subject to performance considerations – *"Ice cream, he said that she said that he said that she said … I like."*

UDCs cover a range of more specific constructions, including **topicalisation** (36), ***wh*-questions** (37) and **relative clauses** (38), **cleft sentences** (39) and **pseudo-clefts** (40), among others.

(37) *"Which ice cream do you like?"*

(38) *"The ice cream which I like is very expensive."*

(39) *"It is Portia who I like."*

(40) *"What I like is ice-cream."*

UDCs are handled in HPSG by the SYNSEM|NONLOCAL attribute. This takes as its value a feature structure of sort *nonlocal* which is appropriate for the features SLASH, REL and QUE, which all take sets of various sorts as values, as shown in figure 38.

$$\left[\text{SYNSEM}\ \left[\begin{array}{ll} \textit{synsem} & \\ \text{NONLOCAL} & \left[\begin{array}{ll} \textit{nonlocal} & \\ \text{SLASH} & \textit{set(local)} \\ \text{REL} & \textit{set(ref-index)} \\ \text{QUE} & \textit{set(quantifiers)} \end{array}\right] \end{array}\right]\right]$$

Figure 38: NONLOCAL values.

The attributes REL and QUE are used to handle some other classes of long-distance phenomena associated with *wh*-questions and relative clauses, respectively, and we will defer discussion of them until section 8.3. The attribute SLASH is the one which deals with UDCs. Note that it takes as its value a *set* of feature structures of sort *local*. Its name comes from a notational device used to represent UDCs in trees. The notation XP/YP denotes informally a constituent of type XP which has somewhere within it a gap of type YP. In the case of (36), XP is S and YP is NP. The structure of (36) is shown in figure 39 using this notation. As can be seen from this example, information about the dependency is passed from the filler through the tree, from mother to daughter, until it reaches the gap (or vice versa, since directionality is not at issue).

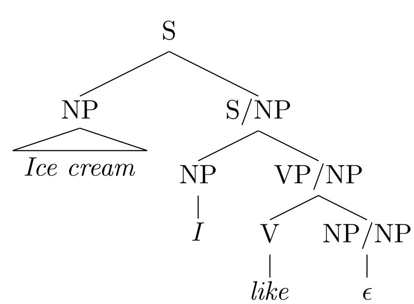


Figure 39: *"Ice cream, I like."*

UDCs can be decomposed into three components:

the top, where the unbounded dependency is introduced,

the bottom, where it is resolved, and

the middle, where the dependency passes through the intervening structure.

In Phrase Structure grammar terms, for the top we require a rule of roughly the following kind.

$$\mathsf{S} \quad \rightarrow \quad \mathsf{XP} \quad \mathsf{S/XP}$$

The intended interpretation of this rule is that a sentence may consist of some phrase XP, followed by a sentence which contains a gap of the same type (XP).

For the bottom, we require a lexical entry of the following kind,

$$\mathsf{XP/XP} \quad \rightarrow \quad \epsilon$$

which states that the 'empty category' XP/XP has no phonetic realisation.

For the middle, we require some general principle which determines how the slash value is shared between mother and daughters.

In translating these informal ideas into HPSG, we will start with the top of the construction, which is defined as a phrase of sort *head-filler-ph* (cf. figure 5 on page 8.), with XP being the filler and S/XP being the head. The sort *head-filler-ph* has the following constraints.

Head Filler Phrase

$$hd\text{-}filler\text{-}ph \quad \Longrightarrow \quad \begin{bmatrix} \text{NONLOC} & \begin{bmatrix} \text{SLASH} & \{\ \} \end{bmatrix} \\ \text{HD-DTR} & \begin{bmatrix} \text{HEAD} & verb \\ \text{SUBJ} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{LEX} & - \\ \text{SLASH} & \{\boxed{1}\} \end{bmatrix} \\ \text{NON-HD-DTR} & \left\langle \begin{bmatrix} \text{LOC} & \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix}$$

These constraints require that the head daughter at the top of a UDC must be an S which has a single item in its SLASH value set and that this item is token identical to the LOCAL value of the non-head daughter. This ensures that the category of the filler and the category of the slash (and hence that of the gap) match. Furthermore, the phrase's own value for SLASH must be the empty set – the dependency cannot be 'passed up' any further. The values for example (36) are shown in figure 40.

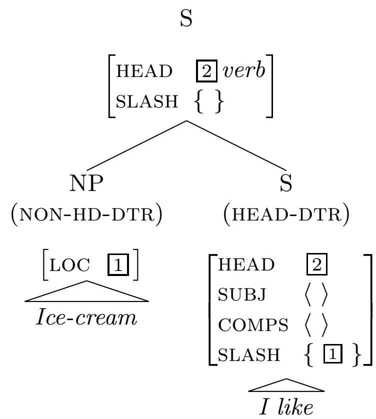


Figure 40: The top of an unbounded NP dependency.

For the bottom of a UDC we have the following lexical entry for an ‘empty category’ – also called a **trace**.

The Lexical entry for trace

$$\begin{bmatrix} \textit{sign} & \\ \text{PHONOLOGY} & \langle\ \rangle \\ \text{SYNSEM} & \begin{bmatrix} \text{LOCAL} & \boxed{1} \\ \text{NONLOCAL} & \begin{bmatrix}\text{SLASH} & \{\boxed{1}\}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

Here the value of the attribute PHONOLOGY is the empty list and the sign’s LOCAL[29] value is shared with the only member of the NONLOCAL|SLASH value. (The counterpart of XP/XP).

Finally, we provide a general constraint on *head-valence-ph* that handles the middle of a UDC.

Slash Inheritance Principle

> In a object of sort *head-valence-ph*, the value of SLASH is the set union of the SLASH values of the daughters.[30]

In addition, we need now to specify that all lexical entries with the exception of the empty category above contain [NONLOCAL|SLASH {}].

---

[29] The restriction to *local* entities means that UDCs are prevented from referring to *nonlocal* information – for example, specifying that an extracted constituent must itself contain a gap.

[30] Recall that *hd-valence-ph* subsumes the sorts *hd-subject-ph*, *hd-complement-ph* and *hd-specifier-ph*. Cf. figure 5 on page 8.

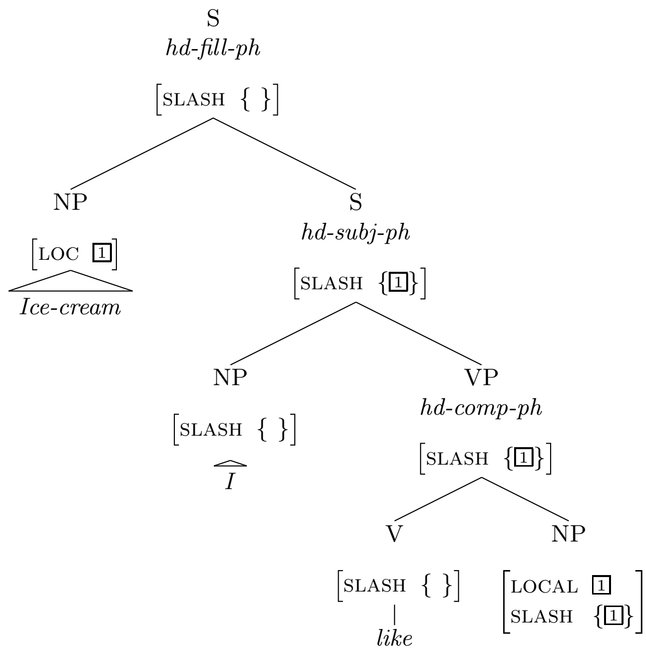


Figure 41: HPSG analysis of *"Ice cream, I like"*.

We are now in a position to translate figure 39 into HPSG terms, as shown in figure 41. The slash values of VP and the lower S are the union of the slash values of their daughters, in conformance with the Slash Inheritance Principle. Since *head-filler-ph* is not a subsort of *head-valence-phrase*, the topmost S of the construction is not constrained by the Slash Inheritance Principle; instead, the constraint associated with head-filler phrases requires that the SLASH value of the mother be empty, terminating the unbounded dependency. The coindexing of the LOCAL values of the filler and the gap ensures that they have the same syntactic category, in this case NP.[31]

The analysis of UDCs that we have just outlined assumes the existence of an empty category. Although we have not yet discussed the question of parsing using feature structures, we will show in the next chapter that the techniques discussed in earlier chapters are also applicable to these kinds of representations. The existence of empty categories therefore leads to the postulation of ϵ-productions, which, as was pointed out in chapter 3, cause problems of non-termination for bottom-up parsers. One of the possibilities we mentioned in that chapter for avoiding these problems is to use a grammar that does not use ϵ-productions. It is of interest, therefore, to note that a more recent alternative treatment of unbounded dependencies in HPSG in fact proposes the abandonment of the empty category approach to unbounded dependencies in favour of a lexical one.

This alternative exploits the strongly lexicalist nature of HPSG. Every lexical sign already encodes, via its valence attributes, the syntactic arguments with which it combines to form phrases. It is not necessary, therefore, to actually build phrases in order to specify that one or more arguments may be missing. This can be accomplished by modifying the information contained in the lexical

[31] Note that CASE is also a part of LOCAL, so that the case specification of the filler and gap will be identical, guaranteeing that only (i) is defined as well-formed.

(i) *"Me, he likes."*

(ii) * *"I, he likes."*

Taking the union of the daughters' SLASH values may seem to be vacuous, since in this example only one daughter has a non-empty value. There are, however, languages which permit more than one gap in questions and topicalisations and even in English there is one construction that permits multiple gaps, in examples like *"[Which violins]$_i$ are [those sonatas]$_j$ easy to play gap$_j$ on gap$_i$"*, for which set union correctly increases the number of elements in the mother's SLASH value (i.e. the phrase *"play gap$_j$ on gap$_i$"*.).

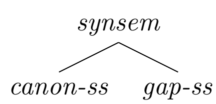
entry itself. The lexical version of UDCs requires a number of modifications to our earlier account. The first is that we modify the constraints associated with the sort *word* to include the following.

The Slash Amalgamation Constraint

$$word \Longrightarrow \begin{bmatrix} \text{LOC} & \begin{bmatrix} \text{ARG-STR} & \left\langle \begin{bmatrix}\text{SLASH } \boxed{1}\end{bmatrix}, \ldots, \begin{bmatrix}\text{SLASH } \boxed{n}\end{bmatrix} \right\rangle \\ \text{BIND} & \boxed{2} \end{bmatrix} \\ \text{NONLOC} & \begin{bmatrix} \text{SLASH} & (\boxed{1} \uplus \ldots \uplus^{32} \boxed{n}) - \boxed{2} \end{bmatrix} \end{bmatrix}$$

This ensures that the SLASH value of each of a lexical head's arguments is shared by the head. If these are all the empty set, then the head's SLASH value will be the empty set (and there will be no UDC). If, on the other hand, at least one of the arguments has a non-empty SLASH value, the SLASH value of the head itself will share that non-null value. The feature BIND, which takes a set of *synsem* objects as value, is introduced to deal with cases (not discussed so far) where a lexical head may bind off an unbounded dependency.[33]

The second modification is to partition the sort *synsem* into two subsorts, called *canonical-synsem* and *gap-synsem*.



The sort *canonical-synsem* is just the set of SYNSEM values that we have been using so far. The sort *gap-synsem* is defined as follows.

$$\begin{bmatrix} \textit{gap-synsem} & \\ \text{LOCAL} & \boxed{1} \\ \text{NONLOCAL} & \begin{bmatrix} \text{SLASH} & \{\boxed{1}\} \end{bmatrix} \end{bmatrix}$$

In contrast to all previous examples, no actual lexical item in English (or any other language) contains a non-canonical SYNSEM value.

The third modification is to change the constraint on the sort *word* which relates the values of the SUBJ, SPR, COMPS and ARG-STR attributes. Instead of simply relating them via append, we have the more complex Argument Realisation Constraint.[34]

[32] $\uplus$ denotes the **disjoint union** of two sets: $A \uplus B \equiv A \cup B \wedge (A \cap B = \emptyset)$. If, for example, $A \uplus B = \{a, b, c\}$, then A and B can have the following values:

$\{\}$	$\{a, b, c\}$
$\{a\}$	$\{b, c\}$
$\{a, b\}$	$\{c\}$
$\{a, b, c\}$	$\{\}$

In contrast to normal set union, what is not possible is $A = \{a, b\} \wedge B = \{b, c\}$.

[33] An example of such a head is the adjective *"easy"*, and many other semantically related adjectives, as shown in (i), where there is an NP gap following *"deceive"*.

(i) *"Malvolio is easy to deceive."*

[34] The Argument Realisation Constraint replaces the constraint on valence values given in table 4 on page 33.

The Argument Realisation Constraint

$$
\textit{word} \implies \begin{bmatrix} \textsc{subj} & \boxed{1} \\ \textsc{spr} & \boxed{2} \\ \textsc{comps} & \boxed{3}\,\textit{list(canon-ss)} \\ \textsc{arg-str} & \boxed{1} \oplus \boxed{2} \oplus (\boxed{3} \bigcirc \textit{list(gap-ss)}) \end{bmatrix}
$$

This states that the ARG-STR list may contain objects of sort *gap-synsem*, but the COMPS list may not – complements must be phonetically realised. The symbol $\bigcirc$ represents the **sequence union** or **shuffle** operator. The shuffle relation holds of three sequences A, B and C if C is a sequence that contains all and only the elements of A and B, and the relative order of the elements in A and the relative order of the elements in B are both preserved in C. Suppose that $C = \langle a, b, c \rangle$ then $A \bigcirc B$ is true of each of the following pairs of values of A and B.

A	B
$\langle a, b, c \rangle$	$\langle\ \rangle$
$\langle a, b \rangle$	$\langle c \rangle$
$\langle a, c \rangle$	$\langle b \rangle$
$\langle a \rangle$	$\langle b, c \rangle$
$\langle b, c \rangle$	$\langle a \rangle$
$\langle b \rangle$	$\langle a, c \rangle$
$\langle c \rangle$	$\langle a, b \rangle$
$\langle\ \rangle$	$\langle a, b, c \rangle$

Suppose that the ARG-STR value of some head is $\langle \text{NP}_1, \text{NP}_2, \text{PP} \rangle$ and that there is no specifier (i.e. the value of the tag $\boxed{2}$ is the empty list). The bracketing of the ARG-STR list in the definition of the Argument Realisation Constraint indicates that the shuffle relation is defined only over arguments to the right of the specifier, so the tag $\boxed{1}$ always has the value $\langle \text{NP}_1 \rangle$. If both the remaining arguments are of sort *canonical-synsem* then the other values in the Argument Realisation Constraint have the following values, $\boxed{3} = \langle \text{NP}_2, \text{PP} \rangle$ and *list(gap-ss)* $= \langle\ \rangle$. If, say, PP is of sort *gap-synsem*, then the values are: $\boxed{3} = \langle \text{NP}_2 \rangle$ and *list(gap-ss)* $= \langle \text{PP} \rangle$. If both the second and third arguments are of sort *gap-synsem*, then the values are: $\boxed{3} = \langle\ \rangle$ and *list(gap-ss)* $= \langle \text{NP}_2, \text{PP} \rangle$.

Exercise 14

There is one further possibility. What is it?

A more subtle effect of the Argument Realisation Constraint is that it says nothing about SUBJ values; it does not constrain them to be canonical, but, at the same time, it does not allow them to be 'transferred' to the SLASH value. In contrast to complements, whatever appears in first position on the ARG-STR list also appears on the SUBJ list. We will return to the issue of subjects and gaps in section 6.1.

The final modification needed is to the Slash Inheritance Principle (page 47). The mother now simply inherits the SLASH value of its head daughter.

Slash Inheritance Principle - Revised version

$$
\textit{hd-valence-ph} \implies \begin{bmatrix} \textsc{slash} & \boxed{1} \\ \textsc{hd-dtr} & \begin{bmatrix} \textsc{slash} & \boxed{1} \end{bmatrix} \end{bmatrix}
$$

Let us now put all these changes together and show their effect on a mono-transitive verb such as *"likes"*. The lexical entry for *"likes"* is described by the following AVM.

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{likes} \rangle \\ \text{ARG-STR} & \left\langle \begin{bmatrix}\text{LOC} & \text{NP}\\ \text{SLASH} & \boxed{1}\end{bmatrix}, \begin{bmatrix}\text{LOC} & \text{NP}[\textit{acc}]\\ \text{SLASH} & \boxed{2}\end{bmatrix} \right\rangle \\ \text{SLASH} & \boxed{1} \uplus \boxed{2}\end{bmatrix}$$

The COMPS value of this lexical entry can satisfy the Argument Realisation Constraint in either of the two ways shown in figure 42.

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{likes} \rangle \\ \text{SUBJ} & \left\langle \boxed{1} \begin{bmatrix} \textit{ss} & \\ \text{LOC} & \text{NP} \\ \text{SLASH} & \boxed{2} \end{bmatrix} \right\rangle \\ \text{COMPS} & \left\langle \boxed{3} \begin{bmatrix} \textit{canon-ss} & \\ \text{LOC} & \text{NP}[\textit{acc}] \\ \text{SLASH} & \boxed{4} \end{bmatrix} \right\rangle \\ \text{ARG-STR} & \langle \boxed{1}, \boxed{3} \rangle \\ \text{SLASH} & \boxed{2} \uplus \boxed{4} \end{bmatrix}$$

(a)

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{likes} \rangle \\ \text{SUBJ} & \left\langle \boxed{1} \begin{bmatrix} \textit{ss} & \\ \text{LOC} & \text{NP} \\ \text{SLASH} & \boxed{2} \end{bmatrix} \right\rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{ARG-STR} & \left\langle \boxed{1}, \begin{bmatrix} \textit{gap-ss} & \\ \text{LOC} & \boxed{4} \\ \text{SLASH} & \boxed{4} \end{bmatrix} \right\rangle \\ \text{SLASH} & \boxed{2} \uplus \left\{ \boxed{4}\text{NP}[\textit{acc}] \right\} \end{bmatrix}$$

(b)

Figure 42: Argument realisation for a mono-transitive verb.

In figure 42(a), the second argument is a *canonical-ss* and it therefore realised as the value of COMPS.[35] This is the kind of feature structure which occurs in sentences in which there is no UDC (e.g. (41)), where each of the SLASH values is the empty set, and also in sentences in which the SLASH value of the complement is non-empty, in which case the gap appears *within* the complement, as in (42), in which it occurs inside the NP *"every kind of"*. In figure 42(b), the second argument is a *gap-ss* and appears in the SLASH set of *"likes"* and is therefore realised as a gap, as in (43)

(41) *"Toby likes every kind of scotch."*

(42) *"Scotch, Toby like every kind of."*

(43) *"Scotch, Toby likes"*

The tree in figure 43 shows the middle of the construction for both (41) and (42). The tag $\boxed{5}$ is instantiated to the value of the NP complement's SLASH value. This appears as part of the SLASH value of the verb *"likes"* because of the Slash Amalgamation Constraint and on the VP and S because of the Slash Inheritance Principle. If this value the empty set, then the NP contains no gap; if it is non-empty, then there must be a missing constituent of the appropriate kind somewhere within the NP.

A continuation of the tree in figure 43 in which the NP contains an NP gap is shown in figure 44, which also illustrates the lexical termination of a UDC. The argument of the preposition *"of"* is of sort *gap-synsem* and, therefore, as a result of the Argument Realisation Constraint, cannot appear on the preposition's COMPS list, which only permits objects of sort *canonical-synsem*. From the Slash Amalgamation Constraint it follows that presence of an object of sort *gap-synsem* on the head's ARG-STR results in the preposition itself having a non-empty SLASH value.

[35] As discussed above, the SUBJ value can be either canonical or a gap, hence the specification *synsem*

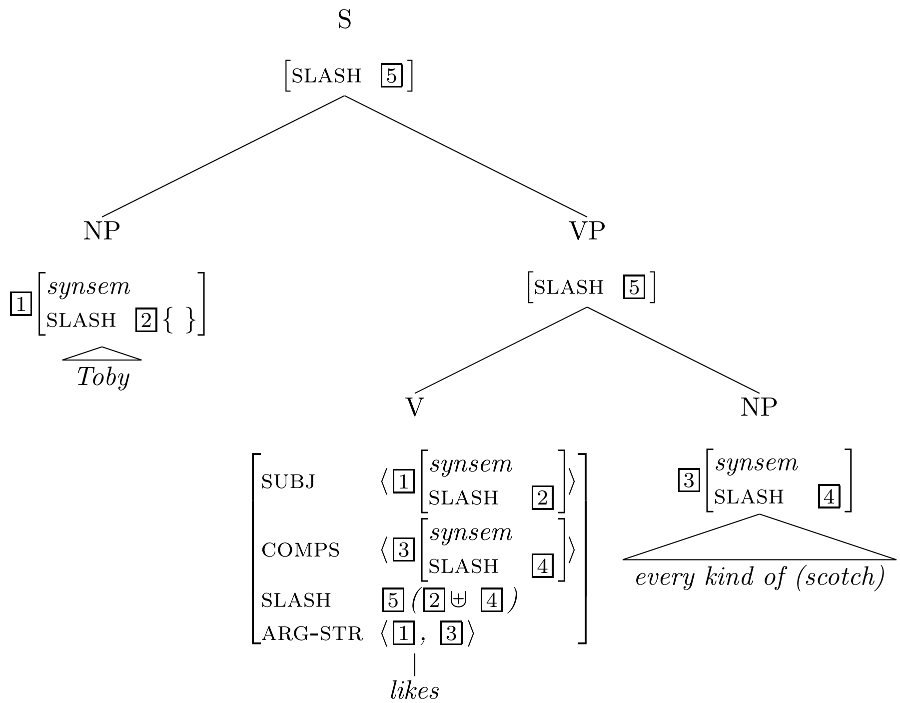


Figure 43: The Slash Inheritance Principle.

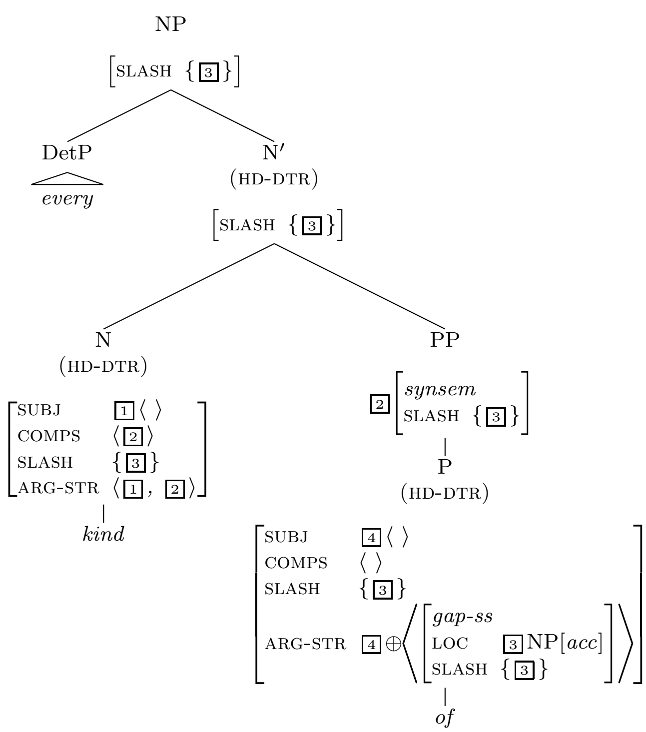


Figure 44: An NP containing an NP gap.

6.1 Subject extraction

All of the discussion of UDCs above has been restricted to those that terminate in what would have been a complement position. In this section, we extend the analysis to subject gaps. There is a well-known restriction on the distribution of subject gaps in English (and many other languages), namely that they cannot occur immediately after a complementiser, as shown by the contrast between (44) and (45).

(44) * *"Who did Andrew say that ___ liked scotch?"*

(45) *"Who did Andrew say ___ liked scotch?"*

While the lexical analysis of UDCs described in the preceding section requires that elements of the COMPS list must be of sort *canonical-synsem*, it places no sortal restriction on the SYNSEM value of subjects. Consequently, the theory permits signs such as the following.

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{liked} \rangle \\ \text{SUBJ} & \left\langle \boxed{1} \begin{bmatrix} \textit{gap-ss} & \\ \text{LOC} & \boxed{2} \\ \text{SLASH} & \{\boxed{2}\} \end{bmatrix} \right\rangle \\ \text{COMPS} & \langle \boxed{3}\,\text{NP}[\textit{acc}] \rangle \\ \text{ARG-STR} & \langle \boxed{1}, \boxed{3} \rangle \\ \text{SLASH} & \{\boxed{2}\} \end{bmatrix}$$

This word, together with the other principles of HPSG, licenses phrases like that shown in figure 45.

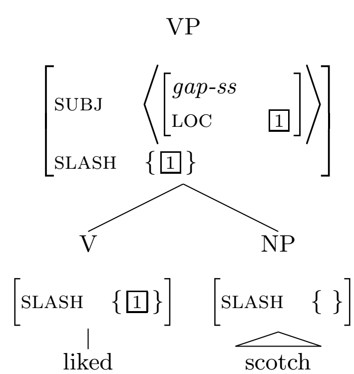


Figure 45: VP with a *gap-synsem* SUBJ value.

Examples like (45) can be accounted for if verbs such as *"say"* are specified having the particular lexical property of taking, not a [SUBJ ⟨ ⟩] complement, as proposed on page 29, but rather a complement specified as [SUBJ *list(gap-ss)*]. This means that such a verb can combine with the SYNSEM component of signs like that in figure 45, as shown in figure 46. If, on the other hand, the complementiser *"that"* is specified as requiring a [SUBJ ⟨ ⟩] complement, (44) is automatically disallowed. Note that [SUBJ ⟨ ⟩] is one of the possibilities subsumed by [SUBJ *list(gap-ss)*] (since *elist* is a subsort of *list*), which means that verbs such as *"say"* may also take the saturated clausal complements S and CP.

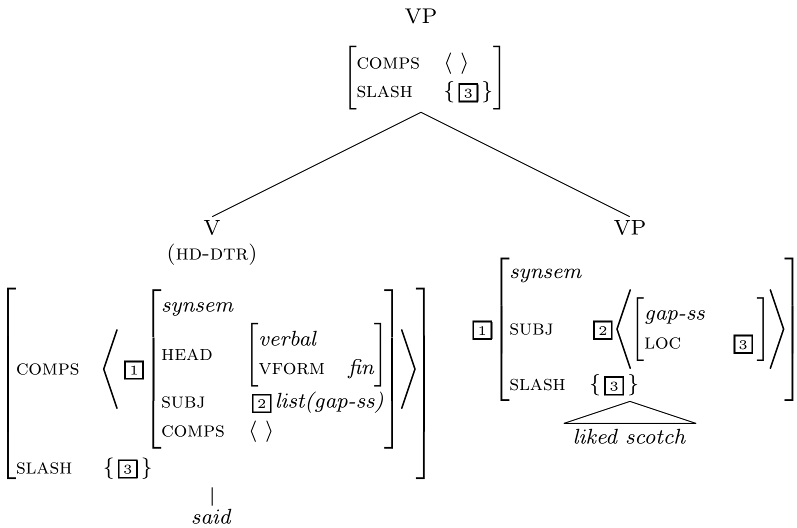


Figure 46: A subject ‘gap’ in an unbounded dependency.

These remarks are only directed at the subjects of subordinate clauses. In main clauses there is never a subject gap. The *“who”* in *“Who drinks scotch?”* is simply in the normal subject position.

Exercise 15

*Show, by drawing the relevant subparts of trees, that the requirement that verbs such as “say” select phrasal verbal complements specified as [*SUBJ *list(gap-ss)] correctly determines the following patterns of grammaticality.*

(a) “Kim said that Sandy had left”

(b) “Kim said Sandy had left”

(c) “I wonder who Kim said had left”

*(d) *“I wonder who Kim said that had left”*

(e) “I wonder who Kim said Sandy had met”

Exercise 16

Provide and analysis of the sentence “Who drinks scotch?” by drawing a tree or AVM containing the relevant features and values.

7 Semantics

So far, all of our discussion of HPSG has been restricted to syntax. In this section we turn our attention briefly to semantics. Space does not permit an extended discussion of the HPSG treatment of semantics and we will restrict ourselves to outlining some of its central properties.[36]

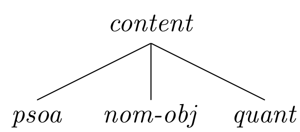
The first point to make is that semantic information, like syntactic information, is represented by feature structures. One of the advantages of using feature structures is that they provide a uniform

[36]Since the inception of HPSG, its treatment of semantics has been inspired by Situation Semantics. We will follow this tradition here, pointing out correspondences to standard logical treatments as we go. It is important to realise, however, that HPSG is perfectly compatible with other approaches to semantics. Richter and Sailer (1999), for example, show how to encode standard predicate logic in typed feature structures.

representation language for a wide range of different phenomena. A second point is that semantic information is also organised into sortal systems. The attribute that bears semantic information is called CONTENT and has values of sort *content*.

$$\begin{bmatrix} \textit{sign} & \\ \text{SYNSEM} & \begin{bmatrix} \textit{synsem} & \\ \text{LOCAL} & \begin{bmatrix} \textit{local} & \\ \text{CAT} & \textit{category} \\ \text{CONTENT} & \textit{content} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

content forms the top of sortal hierarchy with subsorts *parameterised-state-of-affairs*, *nominal-object* and *quantifier*, as shown below.



These three sorts are used to define the semantics for different classes of syntactic objects. Broadly speaking, verbs and verbal projections have *psoa* CONTENT values, nouns and nominal projections *nom-obj* values and determiners *quantifier* values. The sort *psoa* corresponds very roughly in FOL terms to a predicate whose argument positions are occupied by variables (the 'parameters' of the name). It has the following constraint.

$$\begin{bmatrix} \textit{psoa} & \\ \text{QUANTS} & \textit{list(quantifier)} \\ \text{NUCLEUS} & \textit{relation} \end{bmatrix}$$

The value of NUCLEUS is the sort *relation*. The idea is that in *psoa*s the quantificational information appears as the value of QUANTS and is segregated from the quantifier-free component of the semantics in NUCLEUS. Let us move rapidly to a concrete example by giving the CONTENT value for *"like"*.

$$\begin{bmatrix} \textit{psoa} & \\ \text{QUANTS} & \langle\,\rangle \\ \text{NUCLEUS} & \begin{bmatrix} \textit{like-rel} & \\ \text{LIKER} & \textit{index} \\ \text{LIKED} & \textit{index} \end{bmatrix} \end{bmatrix}$$

This corresponds to FOL representations like *like1(x,y)*, discussed in chapter 5, but provides more fine-grained information about the semantic rôles of the arguments. Relations form a sortal hierarchy, which permits a systematic structuring of lexical relations. We will not discuss this aspect of the framework any further here. QUANTS has the empty list as value because this is an unquantified expression.

To proceed any further, we need to say something about the CONTENT of NPs. This is of sort *nominal-object* and has the following attributes.

$$\begin{bmatrix} \textit{nom-obj} & \\ \text{INDEX} & \textit{index} \\ \text{RESTRICTION} & \textit{set(psoa)} \end{bmatrix}$$

The sort *index* is further partitioned into subsorts *referential*, *there* and *it*. The *referential* indices are used for contentful nouns and PPs in argument positions; *there* and *it* for the non-referential 'dummy' NPs *"there"* and *"it"* in sentences such as *"There appears to be a unicorn in the garden"* or *"It is easy to see you don't like ice cream"*. We will only discuss referential indices here. To be interpreted, an index needs to model-theoretically anchored to some appropriate real-world entity. The sort *index* has the following constraint.

$$\begin{bmatrix} \textit{index} & \\ \text{PERSON} & \textit{person} \\ \text{NUMBER} & \textit{number} \\ \text{GENDER} & \textit{gender} \end{bmatrix}$$

The sign for a proper noun such as *"Toby"* will have the following CONTENT value, in which the value of the RESTRICTION attribute is empty,

$$\begin{bmatrix} \text{PHON} & \langle \textit{Toby} \rangle \\ \text{CONTENT} & \begin{bmatrix} \textit{nom-obj} & \\ \text{INDEX} & \begin{bmatrix} \textit{ref} & \\ \text{PERSON} & \textit{3} \\ \text{NUMBER} & \textit{sing} \\ \text{GENDER} & \textit{masc} \end{bmatrix} \\ \text{RESTR} & \{\ \} \end{bmatrix} \end{bmatrix}$$

whereas a common noun, such as *"book"*, will look like the following.

$$\begin{bmatrix} \text{PHON} & \langle \textit{book} \rangle \\ \text{CONTENT} & \begin{bmatrix} \textit{nom-obj} & \\ \text{INDEX} & \boxed{1} \begin{bmatrix} \textit{ref} & \\ \text{PERSON} & \textit{3} \\ \text{NUMBER} & \textit{sing} \\ \text{GENDER} & \textit{neut} \end{bmatrix} \\ \text{RESTR} & \left\{ \begin{bmatrix} \textit{psoa} & \\ \text{QUANTS} & \langle\ \rangle \\ \text{NUCLEUS} & \begin{bmatrix} \textit{book-rel} & \\ \text{INSTANCE} & \boxed{1} \end{bmatrix} \end{bmatrix} \right\} \end{bmatrix} \end{bmatrix}$$

The RESTRICTION value for *"book"* corresponds to the FOL *book1(x)*.

7.1 The semantics of verbs.

We are now in a position to discuss the connection between the syntactic and semantic components of signs, starting with verbs. We will use the abbreviation in figure 47, in which a subscripted tag indicates the INDEX value of an NP's CONTENT.

Abbreviation	Simplified AVM
$NP_{\boxed{i}}$	$\left[\text{LOC} \left[\begin{array}{ll}\text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \textit{noun} \\ \text{SUBJ} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{LEX} & - \end{array}\right] \\ \text{CONTENT} \mid \text{INDEX} & \boxed{i}\end{array}\right]\right]$

Figure 47: Abbreviation for the NP SYNSEM value.

This allows us to express the feature structure for *"likes"* as figure 48.

$$\left[\begin{array}{ll} \textit{word} & \\ \text{PHON} & \langle \textit{likes} \rangle \\ \text{LOCAL} & \left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \textit{verb} \\ \text{SUBJ} & \langle \boxed{1}\text{NP}[\textit{nom}]_{\boxed{i}[3rd, sing]} \rangle \\ \text{COMPS} & \langle \boxed{2}\text{NP}[\textit{acc}]_{\boxed{j}} \rangle \\ \text{ARG-STR} & \langle \boxed{1}, \boxed{2} \rangle \end{array}\right] \\ \text{CONTENT} & \left[\begin{array}{ll} \textit{psoa} & \\ \text{QUANTS} & \langle\ \rangle \\ \text{NUCL} & \left[\begin{array}{ll} \textit{like-rel} & \\ \text{LIKER} & \boxed{i} \\ \text{LIKED} & \boxed{j} \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right]$$

Figure 48: The syntax and semantics of *"likes"*.

The INDEX values of the subject and complement NP's are structure-shared with the values of the argument rôles of the verb's CONTENT, expressing the fact that the LIKER rôle is filled by the referential index of the subject and the LIKED rôle is filled by the referential index of the complement. In this way, lexical signs link up the syntactic and semantic contributions made by their arguments. Since *"likes"* is a finite verb, the index of the subject is assigned the values [PERSON *3*] and [NUMBER *singular*] (abbreviated as [*3rd*, *sing*]), which ensures the correct subject-verb agreement.[37]

The lexical entries for intransitive and ditransitive verbs are analogous. The subject of an intransitive will contribute its index to the only rôle of the relation. With some simplification through the omission of path information, the sign for the intransitive verb *"laughs"* is shown in figure 49(a) and the *to-transitive* verb *"gives"*is shown in figure 49(b).

[37] Note that this way of analysing subject-verb agreement treats it as a semantic, rather than a syntactic, phenomenon.

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \text{LAUGHS} \rangle \\ \text{SUBJ} & \langle \boxed{1}\,\text{NP}[\textit{nom}]_{\boxed{i}[3rd,sing]} \rangle \\ \text{ARG-STR} & \langle \boxed{1} \rangle \\ \text{CONT} & \begin{bmatrix} \textit{psoa} & \\ \text{QUANTS} & \langle\ \rangle \\ \text{NUCL} & \begin{bmatrix} \textit{laugh-rel} & \\ \text{LAUGHER} & \boxed{i} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(a)

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \text{GIVES} \rangle \\ \text{SUBJ} & \langle \boxed{1}\,\text{NP}[\textit{nom}]_{\boxed{i}[3rd,sing]} \rangle \\ \text{COMPS} & \langle \boxed{2}\,\text{NP}[\textit{acc}]_{\boxed{j}},\ \boxed{3}\,\text{PP}[\textit{to}]_{\boxed{k}} \rangle \\ \text{ARG-STR} & \langle \boxed{1},\ \boxed{2},\ \boxed{3} \rangle \\ \text{CONT} & \begin{bmatrix} \textit{psoa} & \\ \text{QUANTS} & \langle\ \rangle \\ \text{NUCL} & \begin{bmatrix} \textit{give-rel} & \\ \text{GIVER} & \boxed{i} \\ \text{GIVEN} & \boxed{k} \\ \text{GIFT} & \boxed{j} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(b)

Figure 49: Simplified AVMs for *"laughs"* and *"gives"*.

To project from the semantics of lexical entries such as these to the semantics of phrases containing them is very simple. We simply need to specify a relationship between the CONTENT value of the phrase and the CONTENT value of the head daughter.

The Content Principle

$$\textit{hd-nexus-ph} \implies \begin{bmatrix} \text{CONTENT} & \boxed{1} \\ \text{HD-DTR} & \begin{bmatrix} \text{CONTENT} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

This is a constraint on *head-nexus-phrase*, which subsumes all subsorts of *headed-phrase* except *head-adjunct-phrase*. It will therefore be applicable to all the structures that we have discussed so far in this chapter. We will briefly illustrate its effect with respect to two examples: (46) with normal constituent order and (47) its counterpart with a UDC in which the complement NP has been displaced to the front of the clause.

(46) *"Toby likes Andrew."*

(47) *"Andrew, Toby likes."*

A tree representation of the structure of example (46) is shown in figure 50.

The CONTENT value of the whole sentence is

$$\begin{bmatrix} \text{QUANTS} & \langle\ \rangle \\ \text{NUCLEUS} & \begin{bmatrix} \textit{like-rel} & \\ \text{LIKER} & \boxed{i} \\ \text{LIKED} & \boxed{j} \end{bmatrix} \end{bmatrix}$$

where the tags $\boxed{i}$ and $\boxed{j}$ are keyed to the individuals denoted by the subject and object NPs *"Toby"* and *"Andrew"* respectively. This NUCLEUS value is identical to that of the head verb *"likes"* and is shared by each of the projections of the head daughter as a consequence of the Content Principle.

The second example (47), whose tree is given in figure 51, is particularly interesting from the point of view of its semantics, because, despite its radically different syntax, absolutely nothing more needs to be said about CONTENT values. The values of the roles LIKER and LIKED are supplied by the INDEX values of the members of the ARG-STR list of the head verb *"likes"*, in exactly the same way as for (46). The structure sharing of LOCAL values in UDCs ensures that this relationship between a head and its arguments is preserved in the UDC.

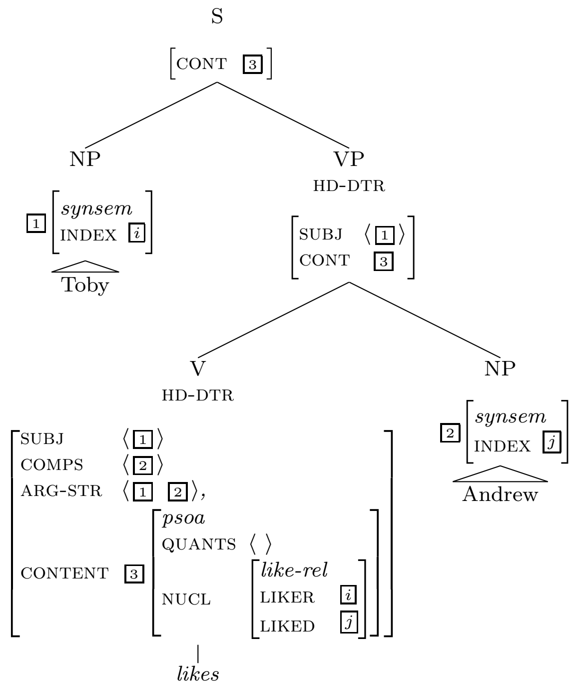


Figure 50: The CONTENT values of *"Toby likes Andrew"*.

7.1.1 Context

The observant reader will have noticed that in figure 51 we have lost information about the names of the individuals involved in the liking relationship. In FOL terms, we have a representation akin to $like1(x, y)$, with x and y anchored to two individuals in the Universe of Discourse.

The HPSG solution to this omission is to assume that meaning not only consists of CONTENT values, but also includes contextual information. With this information added, the structure of *sign*s looks like this:

$$\begin{bmatrix} \textit{sign} & \\ \text{SYNSEM} & \begin{bmatrix} \text{CAT} & \textit{cat} \\ \text{CONTENT} & \textit{content} \\ \text{CONTEXT} & \begin{bmatrix} \text{CONTEXTUAL-INDICES} & \begin{bmatrix} \text{SPEAKER} & \textit{index} \\ \text{HEARER} & \textit{index} \\ \ldots & \end{bmatrix} \\ \text{BACKGROUND} & \textit{set(relation)} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Adding BACKGROUND information to the lexical entry for a proper noun such as *"Toby"* gives the following:

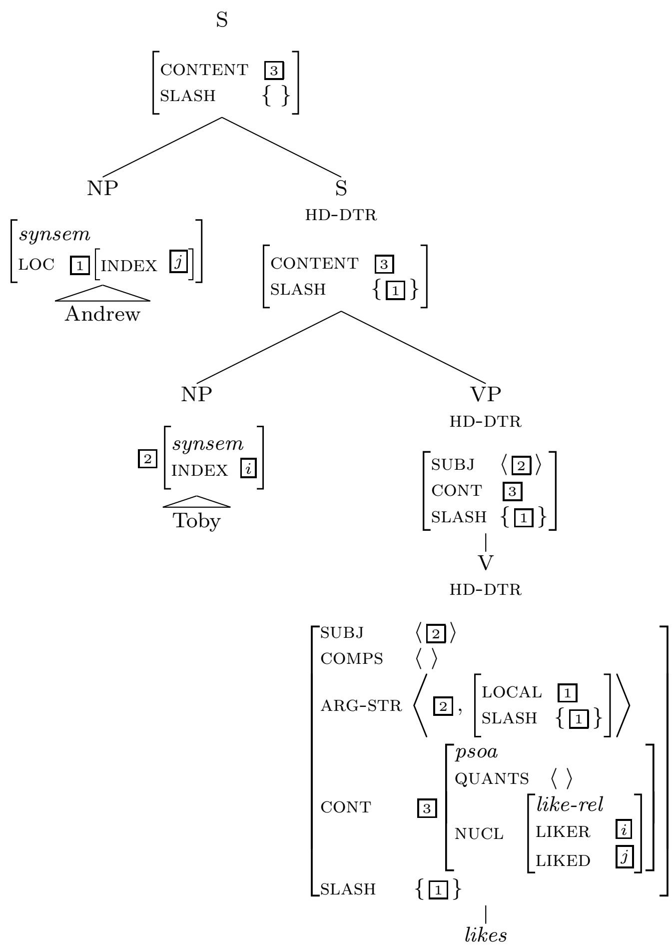


Figure 51: The CONTENT values of the UDC *"Andrew, Toby likes"*.

$$\begin{bmatrix} \text{PHON} & \langle \textit{Toby} \rangle \\ \text{SYNSEM} & \begin{bmatrix} \text{CONTENT} & \begin{bmatrix} \textit{nom-obj} & \\ \text{INDEX} & \boxed{1} \begin{bmatrix} \textit{ref} & \\ \text{PERSON} & \textit{3} \\ \text{NUMBER} & \textit{sing} \\ \text{GENDER} & \textit{masc} \end{bmatrix} \\ \text{RESTR} & \{\ \} \end{bmatrix} \\ \text{CONTEXT} & \begin{bmatrix} \text{BKGRND} & \left\{ \begin{bmatrix} \textit{name-rel} & \\ \text{BEARER} & \boxed{1} \\ \text{NAME} & \textit{Toby} \end{bmatrix} \right\} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The contribution of BACKGROUND values to the larger structures in which they occur is determined by the following constraint:

Principle of Contextual Consistency

> The CONTEXT|BACKGROUND value of a given phrase is the union of the CONTEXT|BACKGROUND values of the daughters.

This ensures that the meaning representation for the sentence *"Toby likes Andrew"* is

$$\begin{bmatrix} \text{SYNSEM} & \begin{bmatrix} \text{CONTENT} & \boxed{2} \begin{bmatrix} \textit{psoa} & \\ \text{QUANTS} & \langle\ \rangle \\ \text{NUCL} & \begin{bmatrix} \textit{like-rel} & \\ \text{LIKER} & \boxed{i} \\ \text{LIKED} & \boxed{j} \end{bmatrix} \\ \text{CONTEXT} & \begin{bmatrix} \text{BKGRND} & \left\{ \begin{bmatrix} \textit{name-rel} & \\ \text{BEARER} & \boxed{i} \\ \text{NAME} & \textit{Toby} \end{bmatrix}, \begin{bmatrix} \textit{name-rel} & \\ \text{BEARER} & \boxed{j} \\ \text{NAME} & \textit{Andrew} \end{bmatrix} \right\} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

There is considerably more to be said about the CONTEXT attribute, but we will not do so here.

7.2 Prepositional Phrases

Having shown with respect to verbs how syntax and semantics are interleaved in lexical entries, we will turn in the next section to the more complex situation represented by NPs but, before we do so, we look briefly at the semantics of prepositional phrases. In section 3.4 we discussed the syntax of those PPs that occur on the argument structure list of another head, as in examples like *"Toby gave a drink to Andrew"*. From a semantic point of view, apart from indicating that Andrew is the recipient of the drink, the preposition *"to"* does not have any significant semantic content.[38] (Note that the sentence is synonymous with *"Toby gave Andrew a drink"*, which contains no preposition.) This can be accounted for if we assign such prepositions the following kind of feature structure.

$$\begin{bmatrix} \text{PHON} & \langle \textit{to} \rangle \\ \text{COMPS} & \langle \text{NP}[\text{CONTENT } \boxed{1}] \rangle \\ \text{CONTENT} & \boxed{1} \end{bmatrix}$$

[38]This not true of all prepositions. Locative prepositions, for example, clearly make a significant semantic contribution.

According to this, the CONTENT of the preposition is simply the CONTENT of its NP complement. Because the value of this CONTENT is of sort *nominal-object*, the Content Principle requires that any PP of which this preposition is the head daughter has an identical CONTENT value, and so the PP inherits an NP denotation.

Exercise 17

Show how the CONTENT *value of the following sentence follows from the lexical entries of the component words and the constraints imposed by the Content Principle.*

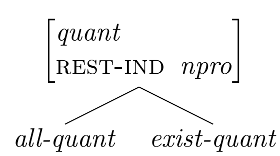
1. *"Toby gave scotch to Andrew."*

7.3 Determiners and quantifiers

The preceding section introduced the basic ingredients of the HPSG treatment of semantics. In this section we extend that to cover quantification. Quantification in HPSG essentially utilises a version of quantifier storage to permit alternative scopings.[39] To incorporate this we modify the definition of the sort *local* to include the attribute QSTORE, which encodes information about stored quantificational meanings.

$$\begin{bmatrix} \textit{local} & \\ \textsc{cat} & \textit{category} \\ \textsc{content} & \textit{content} \\ \textsc{qstore} & \textit{set(quantifier)} \end{bmatrix}$$

Before showing what the feature structure of a determiner such as *"every"* looks like, we need to introduce the sort *quantifier*. This is a subsort of *content* and has subsorts *all-quantifier* and *exist-quantifier*, with the intended interpretations of universal and existential quantifiers respectively.



The sort *quantifier* is appropriate for a feature RESTRICTED-INDEX (abbreviated REST-IND), whose value is an object of sort *non-pronominal* (a subsort of *nominal-object*, and abbreviated to *npro*) with a non-empty RESTRICTION value.

$$\begin{bmatrix} \textit{npro} & \\ \textsc{index} & \textit{index} \\ \textsc{restriction} & \textit{neset} \end{bmatrix}$$

Putting these modifications together, we can now add CONTENT and QSTORE values to the syntax of determiners discussed in section 3.3. The resulting lexical entry for the determiner *"every"* is shown in figure 52.

[39] Cf. chapter 6, section 7.

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{every} \rangle \\ \text{LOCAL} & \begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{det} & \\ \text{SPEC} & \text{N}'[\text{CONT } \boxed{1}] \end{bmatrix} \end{bmatrix} \\ \text{CONT} & \boxed{2}\begin{bmatrix} \textit{all-quant} & \\ \text{REST-IND} & \boxed{1} \end{bmatrix} \\ \text{QSTORE} & \{\boxed{2}\} \end{bmatrix} \end{bmatrix}$$

Figure 52: Abbreviated lexical entry for the determiner *"every"*.

There are two things to note about this feature structure. Firstly, the CONTENT value of the N′ which the determiner selects through its SPEC attribute contributes the value of the determiner's REST-IND, and, secondly, the value of the determiner's own CONTENT is co-tagged with the only element of its QSTORE set – quantifier storage is built into the lexical entry.

The actual value of the tag $\boxed{1}$ is supplied by the N′ in construction with the determiner. A typical example (for the common noun *"book"*) is given in figure 53.

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{book} \rangle \\ \text{LOCAL} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SPR} & \langle \text{DetP} \rangle \\ \text{CONT} & \begin{bmatrix} \textit{npro} & \\ \text{INDEX} & \boxed{1}\,\textit{3ps} \\ \text{RESTR} & \left\{ \begin{bmatrix} \textit{book-rel} & \\ \text{INST} & \boxed{1} \end{bmatrix} \right\} \end{bmatrix} \\ \text{QSTORE} & \{\ \} \end{bmatrix} \end{bmatrix}$$

Figure 53: Abbreviated lexical entry for the noun *"book"* showing CONTENT values.

If we combine these two pieces of information in the phrase *"every book"*, the determiner's CONTENT value will be that shown in figure 54. The QSTORE value thus contains information roughly equivalent to the FOL expression $\forall x(book1(x))$.

$$\begin{bmatrix} \text{CONT} & \boxed{1}\begin{bmatrix} \textit{all-quant} & \\ \text{REST-IND} & \begin{bmatrix} \textit{npro} & \\ \text{INDEX} & \boxed{2}\begin{bmatrix} \textit{ref} & \\ \text{PERSON} & \textit{3} \\ \text{NUMBER} & \textit{sing} \\ \text{GENDER} & \textit{neut} \end{bmatrix} \\ \text{RESTR} & \left\{ \begin{bmatrix} \textit{psoa} & \\ \text{QUANTS} & \langle\ \rangle \\ \text{NUCL} & \begin{bmatrix} \textit{book-rel} & \\ \text{INST} & \boxed{2} \end{bmatrix} \end{bmatrix} \right\} \end{bmatrix} \end{bmatrix} \\ \text{QSTORE} & \{\boxed{1}\} \end{bmatrix}$$

Figure 54: CONTENT value for the determiner *"every"* in the phrase *"every book"*.

QSTORE values are inherited by phrases in a very similar way to SLASH values in UDCs. Firstly, the QSTORE value of a lexical head is the union of the QSTORE values of its arguments.

The Quantifier Amalgamation Principle – First version

$$\textit{word} \implies \begin{bmatrix} \text{ARG-STR} & \langle [\text{QSTORE } \boxed{1}], \ldots, [\text{QSTORE } \boxed{n}\,] \rangle \\ \text{QSTORE} & \boxed{1} \uplus \ldots \uplus \boxed{n} \end{bmatrix}$$

Secondly, the mother inherits the QSTORE value of its head daughter.

The Quantifier Inheritance Principle

$$\textit{head-nexus-ph} \implies \begin{bmatrix} \text{QUANTS} & \boxed{1} \\ \text{HD-DTR} & \begin{bmatrix} \text{QUANTS} & \boxed{1} \end{bmatrix} \end{bmatrix}$$

However, quantifiers must also be retrieved from storage at appropriate points in structure. We therefore modify the definition of the relationship between a word's QSTORE and those of its arguments given on page 64 above in the first version of the Quantifier Amalgamation Principle to take account of this possibility.

The Quantifier Amalgamation Principle – Final version

$$\textit{word} \implies \begin{bmatrix} \text{ARG-STR} & \langle [\text{QSTORE } \boxed{1}], \ldots, [\text{QSTORE } \boxed{n}\,] \rangle \\ \text{QSTORE} & (\boxed{1} \uplus \ldots \uplus \boxed{n}) \text{ - } \boxed{2} \\ \text{CONTENT} & \begin{bmatrix} \textit{psoa} & \\ \text{QUANTS} & \textit{order}(\boxed{2}) \end{bmatrix} \end{bmatrix}$$

The attribute QUANTS plays a rôle here for the first time. Recall (page 55) that QUANTS has as its value a list of objects of sort *quantifier*. The intention is that these represent the quantifiers that have been retrieved from storage and that their order in the list corresponds to their scope – quantifiers on the left scope over those to their right. The tag $\boxed{2}$ denotes a set, so *order*($\boxed{2}$) is a list in which the elements of $\boxed{2}$ appear in some order or other.

The QSTORE of a word, then, is the union of the QSTORES of its arguments, minus any quantifiers that occur in the QUANTS list.

Let us work through an example, using sentence (48).

(48) *"Some student read every book."*

The CONTENT value for the determiner *"every"* in the NP *"every book"* is shown in figure 54. The tree in figure 55 shows the CONTENT and QSTORE values for the phrase *"every book"*. Neither *quantifier* nor *nominal-object*, the CONTENT values for the determiner and the noun, are defined for the attribute QUANTS, so there is no possibility of quantifier retrieval.

The NP *"some student"* has almost exactly the same structure, except that the determiner's CONTENT value is of sort *exist-quantifier* and the noun's NUCLEUS value is of sort *student-rel*.

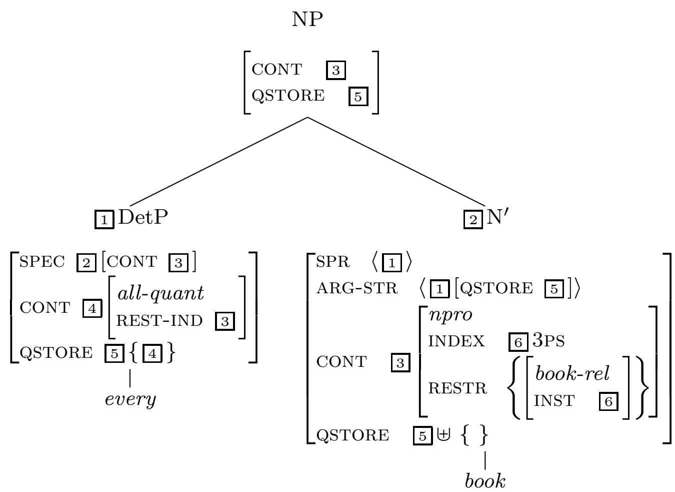


Figure 55: CONTENT and QSTORE values in the NP *"every book"*.

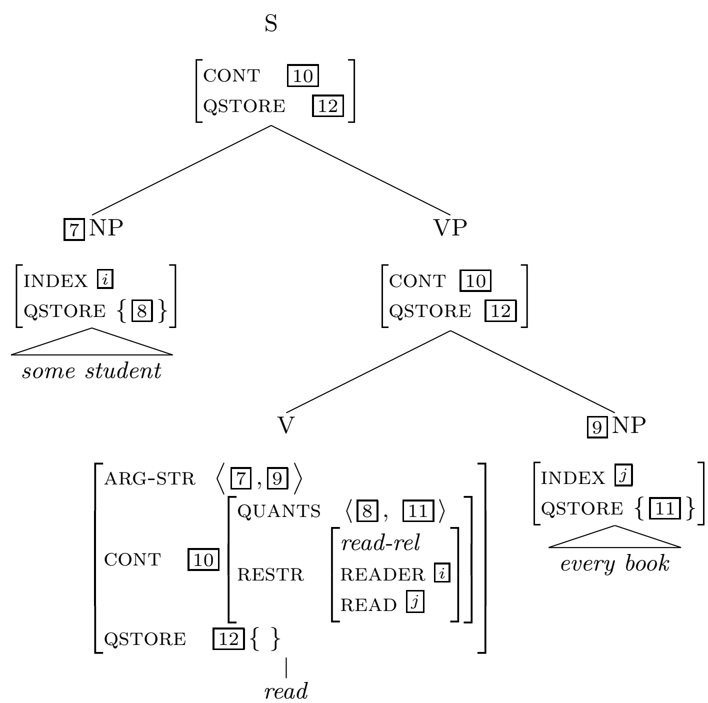


Figure 56: CONTENT, QSTORE and QUANTS values in one reading of *"Some student read every book"*.

The tree in figure 56 shows one possible distribution of values for the whole sentence. In the QUANTS value of the verb *"read"*, both quantifiers occur, in the order: subject's quantifier before complement's quantifier. (This will give a reading in which the existential quantifier has wide scope – a particular student has read every book.) This means that both of these quantifiers must be removed from the value of the verb's QSTORE. Since there are only two items on the verb's argument list, there are no more QSTORE values to union, so the verb's QSTORE value is the empty set. The verb's CONTENT value is inherited by the VP and ultimately by the S. The CONTENT value for the sentence is also shown in AVM form in figure 57. It corresponds to the quantifier scoping given by the FOL formula $\exists x(student1(x) \wedge \forall y(book1(y) \supset read1(x,y)))$

The ordering of the QUANTS list is not deterministic, so the reverse order of quantifiers is also legitimate and would represent the reading in which the universal quantifier has wide scope.

There is also no necessity for both quantifiers to be recovered at this point. In example (49),

(49) *"Every teacher thinks some student read every book."*

if only the quantifier corresponding to *"every book"* is retrieved by the verb *"read"*, its QSTORE set will contain the quantifier corresponding to *"some student"*. This will be inherited by the QSTORE values of the VP and ultimately by that of the CP *"that some student has read every book"*. This CP is on the argument list of the verb *"thinks"*, so its QSTORE contributes to the QSTORE of *"thinks"*, and can be retrieved into the QUANTS value of *"thinks"*. Ordering the QUANTS list provides two different scopings for *"every teacher"* and *"some student"*.

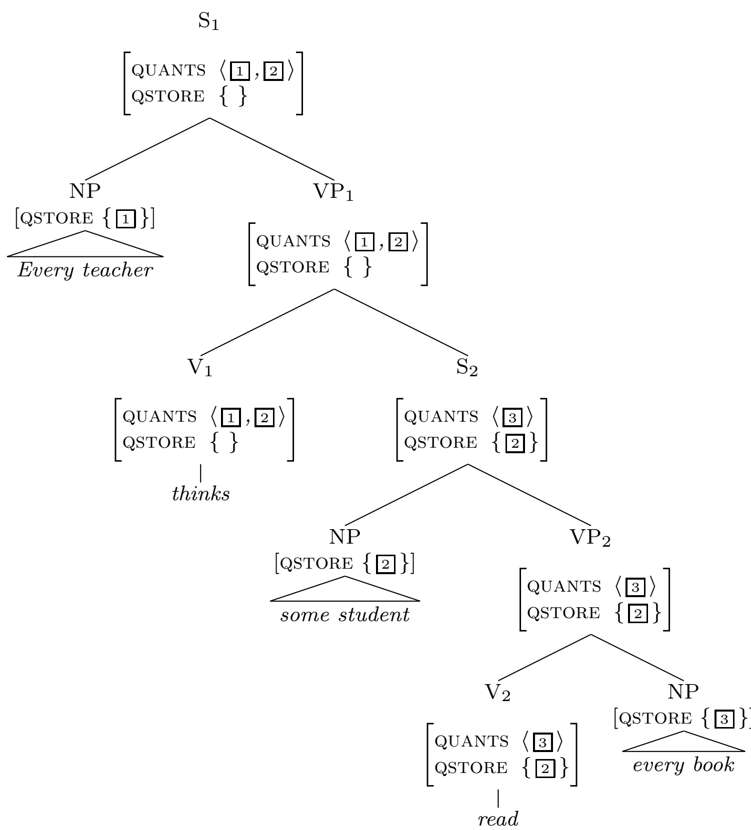


Figure 58 shows the CONTENT value of the reading of (49) in which *"some student"* has wide scope over the whole sentence.

$$\left[\text{CONT}\ \left[\begin{array}{ll} \textit{psoa} & \\ \text{QUANTS} & \left\langle \left[\begin{array}{ll}\textit{exist-quant} & \\ \text{REST-IND} & \left[\begin{array}{ll}\textit{npro} & \\ \text{IND} & \boxed{i} \\ \text{RESTR} & \left\{\left[\begin{array}{ll}\textit{student-rel} & \\ \text{INST} & \boxed{i}\end{array}\right]\right\}\end{array}\right]\end{array}\right], \left[\begin{array}{ll}\textit{all-quant} & \\ \text{REST-IND} & \left[\begin{array}{ll}\textit{npro} & \\ \text{IND} & \boxed{j} \\ \text{RESTR} & \left\{\left[\begin{array}{ll}\textit{book-rel} & \\ \text{INST} & \boxed{j}\end{array}\right]\right\}\end{array}\right]\end{array}\right]\right\rangle \\ \text{NUCLEUS} & \left[\begin{array}{ll}\textit{read-rel} & \\ \text{READER} & \boxed{i} \\ \text{READ} & \boxed{j}\end{array}\right]\end{array}\right]\right]$$

Figure 57: AVM of the CONTENT value for one reading of *"Some student read every book"*.

$$\begin{bmatrix} \text{CONT} & \begin{bmatrix} \textit{psoa} & \\ \text{QUANTS} & \langle \boxed{1}, \boxed{2} \rangle \\ \text{NUCL} & \begin{bmatrix} \textit{think-rel} & \\ \text{THINKER} & \boxed{3} \\ \text{SOA-ARG} & \begin{bmatrix} \textit{psoa} & \\ \text{QUANTS} & \left\langle \boxed{4} \right\rangle \\ \text{NUCL} & \begin{bmatrix} \textit{read-rel} & \\ \text{READER} & \boxed{5} \\ \text{READ} & \boxed{6} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 58: CONTENT value for one reading of *"Every teacher thinks some student read every book"*.

The values of the tags in the QUANTS values are shown in figure 59. The corresponding FOL formula is

$\exists x(student1(x) \wedge \forall y(teacher1(y) \supset think1(y, \forall z(book1(z) \supset read1(x, z)))))$

$$\boxed{1} \begin{bmatrix} \textit{exist-quant} & \\ \text{REST-IND} & \begin{bmatrix} \text{INDEX} & \boxed{5} \\ \text{RESTR} & \begin{bmatrix} \text{NUCL} & \begin{bmatrix} \textit{student-rel} & \\ \text{INST} & \boxed{5} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\boxed{2} \begin{bmatrix} \textit{all-quant} & \\ \text{REST-IND} & \begin{bmatrix} \text{INDEX} & \boxed{3} \\ \text{RESTR} & \begin{bmatrix} \text{NUCL} & \begin{bmatrix} \textit{teacher-rel} & \\ \text{INST} & \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\boxed{4} \begin{bmatrix} \textit{all-quant} & \\ \text{REST-IND} & \begin{bmatrix} \text{INDEX} & \boxed{6} \\ \text{RESTR} & \begin{bmatrix} \text{NUCL} & \begin{bmatrix} \textit{book-rel} & \\ \text{INST} & \boxed{6} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 59: QUANTS values for figure 58.

This concludes our discussion of quantification. In it we have shown how to provide a treatment of quantification and quantifier scoping within a feature structure framework. Note, however, that we have not addressed the problems raised in chapter 6 concerning soundness and completeness nor the vexed matter of determining scoping preferences.

8 Adjuncts

In this section we will look at the syntax and semantics of adjuncts. The term 'adjuncts' covers a range of syntactically heterogeneous constructions. They include modifiers of VPs (50 and 51), so-called 'sentential adverbs' (52), pre-nominal adjective modifiers (53), post-nominal PP modifiers (54), relative clauses (55), 'reduced' relative clauses (56), among others, shown underlined in the following examples.[40]

[40]Some examples of adjuncts were discussed in chapter 6, section 5 in the context of attachment ambiguities.

(50) *"Toby drank the scotch rapidly."*

(51) *"Toby drank in the garden."*

(52) *"Toby drinks scotch, apparently."*

(53) *"Portia is an intelligent person."*

(54) *"The man on the bus was a spy."*

(55) *"The man who we saw on the bus was a spy."*

(56) *"The man seen on the bus was a spy."*

There are many unresolved issues in the identification and analysis of adjuncts, and to some extent they might appear to be the residue left after we have identified subjects, complements and fillers. Nonetheless, some general characterisation of adjuncts is possible. Firstly, the combinatory possibilities allowed between an adjunct and the set of objects it can modify are much wider than those between, say, complements and a head. Typically, a relative clause can be the adjunct of any common noun,[41] whereas the number and form of complements is restricted by the selecting noun (e.g. *"disapproval"* requires its PP complement to contain the preposition *"of"*). (Cf. section 3.4.) Secondly, adjuncts are *optional* constituents; all the sentences in (50)-(56) are perfectly well-formed if the adjunct is omitted. Thirdly, while the number of complements is typically restricted by the head which selects them (e.g. *"read"* permits a maximum of one complement), this is not the case with adjuncts (e.g. *"Macbeth saw Banquo at a banquet in the great hall in the evening"*).

8.1 Adjectival adjuncts

Adjuncts are sensitive to the syntactic status of the constituent they modify – relative clauses modify common nouns, manner adverbs like *"quickly"* modify verb phrases, and not vice versa. The HPSG analysis of all adjuncts proposes, therefore, that adjuncts select the head they modify and uses a feature MODIFIED to effect this selection, in a manner analogous to the SPEC feature of determiners. We will restrict ourselves here to a discussion of modifiers of N′, such as adjectives.[42]

The syntactic component of the feature structure assigned to an adjunct adjective such as *"Scottish"* is the following.

$$\begin{bmatrix} \textit{local} & \\ \textsc{cat} & \begin{bmatrix} \textsc{head} & \begin{bmatrix} \textit{adj} & \\ \textsc{mod} & \mathsf{N}' \\ \textsc{pred} & - \end{bmatrix} \\ \textsc{subj} & \langle\,\rangle \\ \textsc{spr} & \langle\,\rangle \\ \textsc{comps} & \langle\,\rangle \end{bmatrix} \end{bmatrix}$$

This states that the adjective selects an N′ as the item it modifies, and that all its valence lists are empty. The new HEAD feature PRED(ICATIVE) (with value *boolean*) is posited to deal with an additional complication – many words that function as adjuncts lead a dual life. *"Scottish"*, for example, is not restricted to appearing as an adjunct (57), it can also occur as the complement to the copula auxiliary BE (58).

[41] Although not all of the possible combinations will be pragmatically plausible.

(i) *"The number that I met on Thursday."*

(ii) *"A man whose cardinality is greater than zero."*

[42] But we will extend the coverage to VP modifiers in our discussion of implementation in the next chapter.

$$\begin{bmatrix} \textit{synsem} & \\ \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{adj} & \\ \text{MOD} & \text{N}'\begin{bmatrix}\textit{npro} & \\ \text{INDEX} & \boxed{1} \\ \text{RESTR} & \boxed{2}\end{bmatrix} \\ \text{PRED} & - \end{bmatrix} \\ \text{SUBJ} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{CONTENT} & \begin{bmatrix} \text{INDEX} & \boxed{1} \\ \text{RESTR} & \left\{\begin{bmatrix}\textit{psoa} & \\ \text{QUANTS} & \langle\ \rangle \\ \text{NUCL} & \begin{bmatrix}\textit{Scottish-rel} & \\ \text{INST} & \boxed{1}\end{bmatrix}\end{bmatrix}\right\} \cup \boxed{2} \end{bmatrix} \end{bmatrix}$$

Figure 60: SYNSEM value for an attributive adjective.

(57) *"The Scottish king was killed by Macbeth."*

(58) *"Macbeth himself was Scottish."*

The former are called **attributive** uses of the adjective, the latter **predicative** uses. The feature PREDICATIVE is used to distinguish between attributive ([PRED –]) and predicative ([PRED +]) versions of categories.[43]

The semantics of attributive adjectives like *"Scottish"* is quite simple. If, for example, someone satisfies the description *"Duncan was a Scottish king"* then he also satisfies the description *"Duncan was a king"* and the description *"Duncan was Scottish"*. The denotation of *"Scottish king"* is simply the conjunction of denotations of *"Scottish"* and *"king"*. In FOL terms, *"Duncan was a Scottish king"* can be translated as *king1(d)* $\wedge$ *Scottish1(d)*. In HPSG this logical conjunction is represented as the set union of the *psoa*s corresponding to *"Scottish"* and *"king"*. Adding the relevant CONTENT values to the lexical entry for *"Scottish"* gives the AVM in figure 60. The INDEX value of the modified N' is unified with that of the of adjective, giving an effect similar in effect to the lambda-expression $\lambda P[\lambda x\ (Scottish1(x) \wedge P(x))]$ (where P corresponds to the RESTRICTION value of the N' and x to the INDEX value).

We can accommodate both kinds of adjective by defining the sort *adjective-wd* (a partition of *word*) as shown in figure 61. The common property possessed by all adjectives is that their CONTENT value contains an object of sort *psoa*. The value of the semantic argument of the *psoa* is supplied

[43]Many prepositions also have this dual function.

(i) *"The children are in the park."* (Predicative)

(ii) *"The children in the park are playing on the grass."* (Attributive)

In addition, some adjectives have only a predicative rôle, while others have only an attributive one.

(i) *"An utter fool"* (Attributive)

(ii) * *"That fool is utter."*

(iii) *"The door is ajar."* (Predicative)

(iv) * *"The ajar door"*

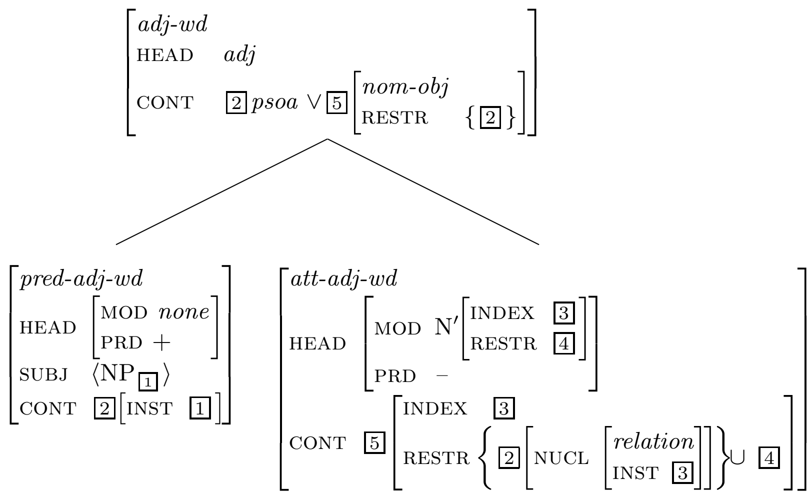


Figure 61: Sortal hierarchy for adjectival words.

by some external constituent – by the subject in the case of a predicative adjective, by the modified constituent in the case of an attributive one.

To make use of attributive adjectives, we need to define the sort of phrases in which they can occur – the sort *head-adjunct-phrase*.

Head-Adjunct Phrase

$$\textit{hd-adj-ph} \Longrightarrow \begin{bmatrix} \text{CONTENT} & \boxed{1} \\ \text{HD-DTR} & \left\langle \begin{bmatrix} \textit{phrase} & \\ \text{SYNSEM} & \boxed{2} \end{bmatrix} \right\rangle \\ \text{NON-HD-DTRS} & \left\langle \begin{bmatrix} \textit{phrase} & \\ \text{HEAD} & \begin{bmatrix} \text{MOD} & \boxed{2} \end{bmatrix} \\ \text{CONT} & \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix}$$

Head-adjunct phrases have the same *syntactic* category as their head daughter (because they are a subsort of *headed-phrase*), but they inherit the CONTENT value of the adjunct daughter. The values for HEAD and CONTENT attributes in the N′ *"Scottish king"* are shown in figure 62.[44]

8.2 PP adjuncts

PPs may also function as N′ adjuncts, in phrases such as *"an actor in London"*. Here, the preposition *"in"*, as the head of the PP, needs to be specified as modifying N′. As can be seen from figure 63, the principal difference between attributive adjectives and attributive prepositions is that the latter take an NP argument whose INDEX value is unified with the value of the LOCATION attribute. The

[44] To conserve space we have only included information about the immediate constituents of the phrase. The values for the words *"Scottish"* and *"king"* are identical to the daughter in figure 62, apart from the value for LEX, which will be minus in both cases.

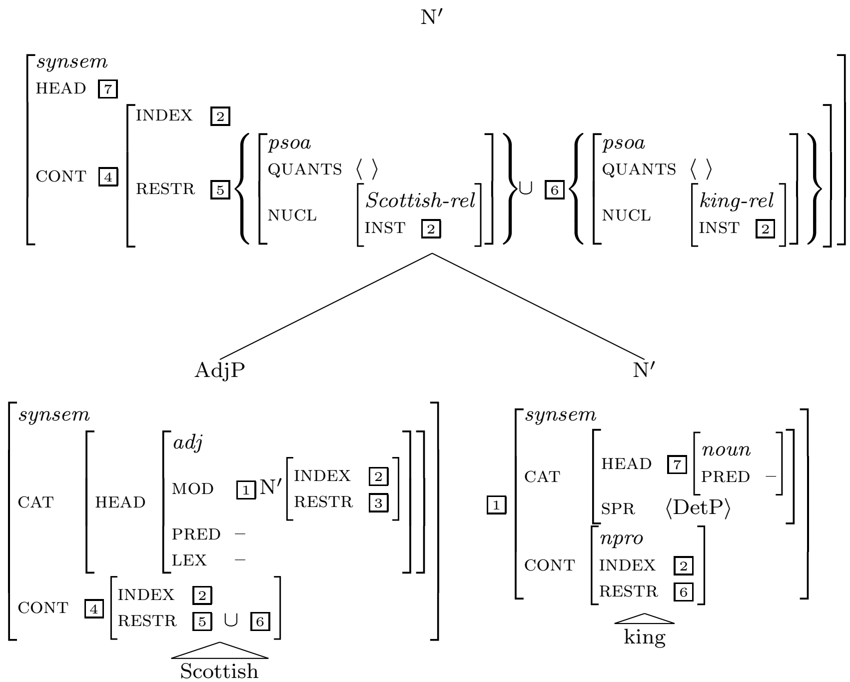


Figure 62: CONTENT and HEAD values for the N′ *"Scottish king"*.

modified N′, as before, contributes its INDEX value to the LOCATEE role and to the INDEX of the preposition itself.

$$\begin{bmatrix} \textit{word} \\ \text{PHON} & \langle \textit{in} \rangle \\ \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \textit{prep} \\ \text{MOD} & \text{N}' \begin{bmatrix} \textit{npro} \\ \text{INDEX} & \boxed{1} \\ \text{RESTR} & \boxed{2} \end{bmatrix} \\ \text{PRED} & - \end{bmatrix} \\ \text{SUBJ} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle \text{NP}_{\boxed{3}} \rangle \end{bmatrix} \\ \text{CONTENT} & \begin{bmatrix} \text{INDEX} & \boxed{1} \\ \text{RESTR} & \left\{ \begin{bmatrix} \textit{psoa} \\ \text{QUANTS} & \langle\ \rangle \\ \text{NUCL} & \begin{bmatrix} \textit{in-rel} \\ \text{LOCATEE} & \boxed{1} \\ \text{LOCATION} & \boxed{3} \end{bmatrix} \end{bmatrix} \right\} \cup \boxed{2} \end{bmatrix} \end{bmatrix}$$

Figure 63: SYNSEM value for an attributive preposition.

The PP inherits its HEAD and CONTENT values from the preposition and the constraints on *head-adjunct-phrase* ensure that the value of the whole phrase's *content* will be as in figure 64, where $\boxed{3}$ is the *index* of the NP *"London"*.

$$\begin{bmatrix} \text{INDEX} & \boxed{1} \\ \text{RESTR} & \left\{ \begin{bmatrix} \textit{psoa} \\ \text{QUANTS} & \langle\ \rangle \\ \text{NUCL} & \begin{bmatrix} \textit{actor-rel} \\ \text{INST} & \boxed{1} \end{bmatrix} \end{bmatrix}, \begin{bmatrix} \textit{psoa} \\ \text{QUANTS} & \langle\ \rangle \\ \text{NUCL} & \begin{bmatrix} \textit{in-rel} \\ \text{LOCATEE} & \boxed{1} \\ \text{LOCATION} & \boxed{3} \end{bmatrix} \end{bmatrix} \right\} \end{bmatrix}$$

Figure 64: CONTENT value of the phrase *"actor in London"*.

Exercise 18

Prepositional phrases come in three flavours.

1. *those that occur in argument positions and whose interpretations are identical to those of NPs, e.g. "Toby offered a drink to Andrew"*
2. *predicative PPs that occur as complements to "be", e.g "Hermes is in Rome"*
3. *attributive PPs, e.g. "Some men in Rome are Italian"*

Provide lexical entries for each of the three type of preposition. prepositions.

8.3 Relative clauses

Relative clauses form a major class of N′ adjuncts. They include ***wh*-relatives**, which can be subdivided into **finite** (59) and **infinitival** (60), **non-*wh*-relatives**, which can be divided into ***that*-relatives** (61) and **contact-relatives** (62). The complete range of relative clauses is far too complex to receive comprehensive coverage here and we will restrict ourselves here to a brief discussion of finite *wh*-relatives.

(59) *"the book which is on the table"*

(60) *"a shelf on which to put the book"*

(61) *"the book that is on the table"*

(62) *"the shelf I put the book on"*

In terms of the way in which the MOD feature is inherited, relative clauses bear a certain similarity to PP adjuncts. In PPs it is the head preposition that is lexically specified as containing the MOD attribute which is inherited by the prepositional phrase. In relative clauses, it is the highest verb that is lexically specified for MOD.[45] Example (61) has the distribution of MOD features shown in figure 65. Since MOD is a HEAD feature, it is shared by all the verb's projections.

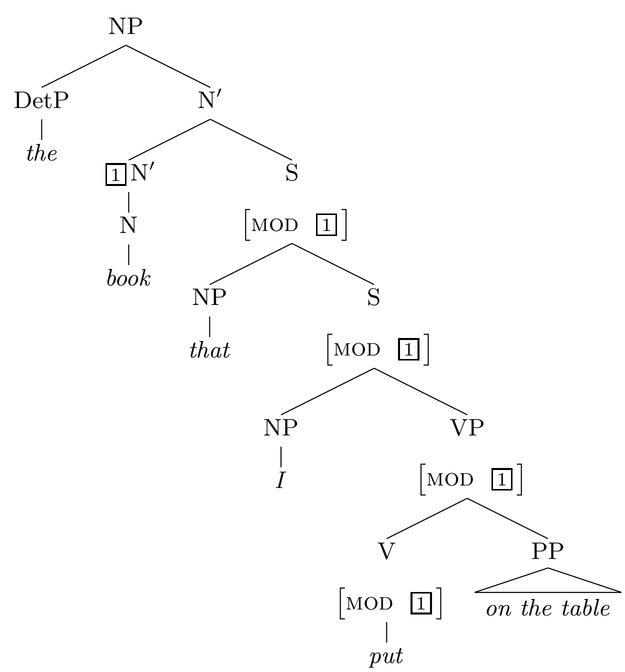


Figure 65: Distribution of the MOD feature in a relative clause.

We need to be able to distinguish between relative clauses and other sorts of clause which have different syntactic characteristics. In addition to relative clauses, we have imperative clauses (63), declarative clauses (64) and interrogative clauses (65).

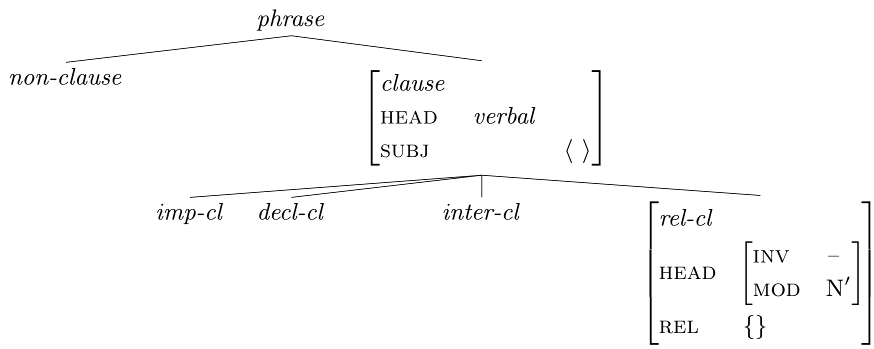
(63) *"Eat your toast."*

(64) *"You are eating something."*

(65) *"What are you eating?"*

[45] This entails that, like adjectives and prepositions, each verb will come in (at least) two forms, one with the feature specification [MOD *synsem*] and one with [MOD *none*], where *synsem* and *none* are partitions of *modifier*. The former function as adjuncts, the latter as 'ordinary' verbs.

This suggests a partition of the sort *phrase* along a dimension of **clausality** as in the following hierarchy.



With the exception of *rel-cl*, all other clause types are defined as [MOD *none*] and cannot function as adjuncts. Relative clauses are constrained to modify N′ and are precluded from containing Subject-Auxiliary Inversion.

8.3.1 *Wh*-words and Pied-piping

Before looking at relative clauses proper, we need say something about *wh*-words. These lexical items get their name because many of them are spelt with the initial letters *"wh"*, e.g. *"who"*, *"whose"*, *"what"*, *"which"*, *"when"* (but also *"how"*). They can occur both in relative clauses and in questions and, in HPSG terms, are characterised by the NON-LOCAL features REL and QUE mentioned in section 6. Relative pronouns have a non-empty value for REL, as illustrated by the lexical entry for relative *"who"* shown in figure 66, and interrogative pronouns have a non-empty value for QUE, as shown in the lexical entry for interrogative *"who"* in 67.

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{who} \rangle \\ \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SUBJ} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{LEX} & + \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \textit{nom-obj} & \\ \text{INDEX} & \boxed{1} \end{bmatrix} \\ \text{REL} & \{\boxed{1}\} \\ \text{QUE} & \{\ \} \end{bmatrix}$$

Figure 66: Lexical entry for relative *"who"*.

It is necessary to distinguish between relative and interrogative categories, not only because they require different semantics (as is apparent from a comparison of figures 66 and 67), but also because their distributions differ – *"what I like"*, for example is good as an interrogative (e.g. *"They asked what I like"*), but bad as a relative clause in many varieties of English (e.g. * *"the man what I like"*). Similarly, *"which"* is an interrogative determiner (e.g. *"Which book did you buy?"*), but not a relative one (e.g. * *"the book which book you bought"*).[46]

[46] The *"which"* that occurs in *"the book which you bought"* is an NP, not a determiner. *"Which"* is categorially ambiguous.

$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{who} \rangle \\ \text{CAT} & \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{SUBJ} & \langle\ \rangle \\ \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \\ \text{LEX} & + \end{bmatrix} \\ \text{CONT} & \boxed{1} \begin{bmatrix} \textit{which-quant} & \\ \text{REST-IND} & \begin{bmatrix} \text{INDEX} & \boxed{2}\ \textit{3rdsing} \\ \text{RESTR} & \left\{ \begin{bmatrix} \textit{psoa} & \\ \text{QUANTS} & \langle\ \rangle \\ \text{NUCL} & \begin{bmatrix} \textit{person-rel} & \\ \text{INST} & \boxed{2} \end{bmatrix} \end{bmatrix} \right\} \end{bmatrix} \end{bmatrix} \\ \text{REL} & \{\} \\ \text{QUE} & \{ \boxed{1} \} \end{bmatrix}$$

Figure 67: Lexical entry for interrogative *"who"*.

The features REL and QUE are distributed by principles very similar to those for SLASH and QUANTS – the REL and QUE values of a word are the disjoint union of the REL and QUE values of its arguments.

The *Wh*-Amalgamation Principle

$$\begin{bmatrix} \textit{word} & \\ \text{ARG-STR} & \left\langle \begin{bmatrix} \text{REL} & \boxed{3} \\ \text{QUE} & \boxed{4} \end{bmatrix}, \ldots, \begin{bmatrix} \text{REL} & \boxed{n} \\ \text{QUE} & \boxed{m} \end{bmatrix} \right\rangle \\ \text{REL} & \boxed{3} \uplus, \ldots, \uplus \boxed{n} \\ \text{QUE} & \boxed{4} \uplus, \ldots, \uplus \boxed{m} \end{bmatrix}$$

The REL and QUE values of the head daughter are inherited by the mother according to the following principle.

The *Wh*-Inheritance Principle

$$\textit{head-nexus-ph} \implies \begin{bmatrix} \text{REL} & \boxed{1} \\ \text{QUE} & \boxed{2} \\ \text{HD-DTR} & \begin{bmatrix} \text{REL} & \boxed{1} \\ \text{QUE} & \boxed{2} \end{bmatrix} \end{bmatrix}$$

The *Wh*-Inheritance Principle has the effect of ensuring that REL and QUE values of a word somewhere in the head daughter of a *head-nexus-phrase* are inherited by the mother. For example, the PP *"in which"* (in a phrase such as *"the house in which I live"*) contains a *wh*-word. *"Which"* here is and NP and has a lexical entry identical (apart from its PHON value) to that of *"who"* in figure 66.

The tree in figure 68 shows the values for the feature REL in the *wh*-phrase *"in which"*.[47] Since the preposition takes only one argument, its own REL value is identical to that of the argument. The REL value on the noun *"which"* comes from its lexical entry. It is shared with the mother NP as a consequence of the *Wh*-Inheritance Principle. The preposition *"in"* has the same REL value by virtue of the *Wh*-Amalgamation Principle, and the PP acquires the same value for REL again as a consequence of the *Wh*-Inheritance Principle.

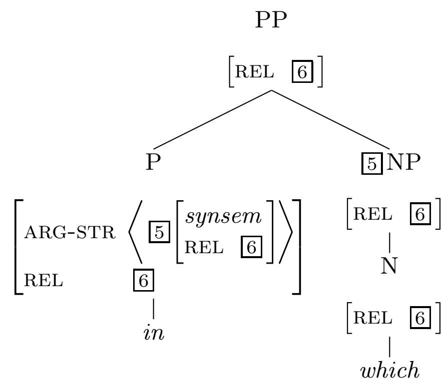


Figure 68: Tree showing distribution of REL values in the PP *"in which"*.

A clause like *"in which I live"*, when compared with *"I live in a house"*, seems to show that the 'movement' of the *wh*-word has dragged the preposition along with it to the front of the clause. This phenomenon is therefore known (somewhat whimsically) as **Pied-piping**. In HPSG, where no movement is involved, Pied-piping corresponds to a *wh*-word being contained in some other constituent, with its REL and QUE values being passed up to the mother.

Pied-piping may affect indefinitely large constituents, such as the phrase

"whose friends' mother's ... uncle's brother"

in examples such as such as

"the guy whose friends' mother's ... uncle's brother I know",

in which the non-empty REL value of the *wh*-determiner *"whose"* is inherited by the whole phrase.

Not all constituent types can host non-empty REL or QUE values. In English verbal, projections may not do so, as evidenced by the ungrammaticality of NPs such as * *"the man to see whom I went"*, containing the illicit VP *"to see whom"*. We impose this restriction by a constraint to the effect that all objects of sort *clause* must have empty specifications for the features REL and QUE.

$$\textit{clause} \implies \begin{bmatrix} \text{REL} & \{\ \} \\ \text{QUE} & \{\ \} \end{bmatrix}$$

The interaction of this constraint with the *Wh*-Inheritance Principle ensures that no VP or verb will have a non-empty REL value.

8.3.2 The Relative Clause

We have seen how the feature MOD functions in a relative clause, and how the feature REL is related to the occurrence of *wh*-words. We now show how these two pieces of information come together to determine the top of the relative clause. We define the sort *wh-rel-cl* as a subsort of *rel-cl*, with the following constraint.

[47] QUE, whose values in this example are all the empty set, has been omitted from the tree for clarity.

$$wh\text{-}rel\text{-}cl \implies \begin{bmatrix} \textit{hd-filler-ph} & \\ \text{HD-DTR} & \begin{bmatrix} \text{MOD} & \text{N}'_{\boxed{1}} \\ \text{SUBJ} & \textit{list(gap-ss)} \end{bmatrix} \\ \text{NON-HD-DTR} & \left\langle \begin{bmatrix} \text{REL} & \{\boxed{1}\} \end{bmatrix} \right\rangle \end{bmatrix}$$

This constraint requires that the INDEX value passed up in the REL value of the *wh*-phrase be identified with the INDEX value of the modified N′.

wh-rel-cl has two subsorts: *su(bject)-rel-cl* and *non-subject-rel-cl* (*ns-rel-cl*), with the following constraints:

$$su\text{-}wh\text{-}rel\text{-}cl \implies \begin{bmatrix} \text{HD-DTR} & \begin{bmatrix} \text{SUBJ} & \left\langle \begin{bmatrix} \textit{gap-ss} & \\ \text{LOC} & \boxed{1} \end{bmatrix} \right\rangle \end{bmatrix} & \\ \text{REL} & \boxed{6}\ \textit{non-hd-dtr} & \left\langle \begin{bmatrix} \text{LOC} & \boxed{1} \\ \text{REL} & \boxed{6} \end{bmatrix} \right\rangle \end{bmatrix}$$

$$ns\text{-}wh\text{-}rel\text{-}cl \implies \begin{bmatrix} \text{HD-DTR} & \begin{bmatrix} \text{SUBJ} & \langle\ \rangle \end{bmatrix} \end{bmatrix}$$

The former defines relative clauses where a *wh*-phrase is in subject position, as in the N′ *"man who likes Toby"*, whose relevant features and values are shown in figure 69.

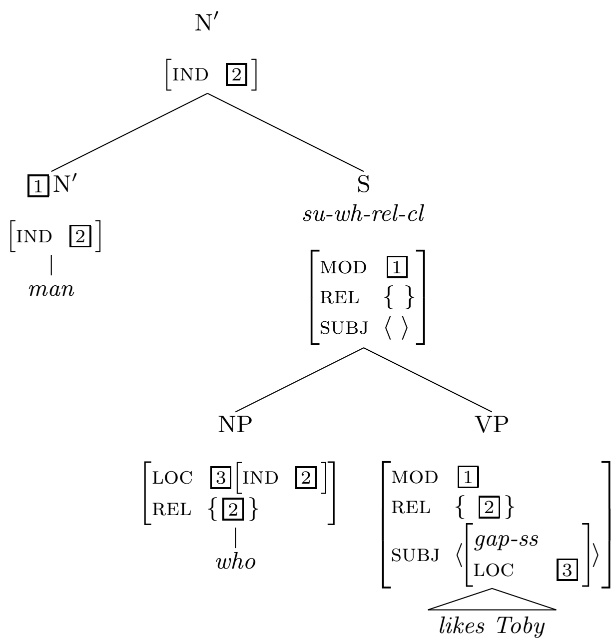


Figure 69: The N′ *"man who likes Toby"*.

The structure dominated by S is a normal *head-filler-phrase*, except that in addition it contains a feature specification for MOD. The REL value of S is empty because it is of sort *rel-clause*. The constraint on *wh-rel-cl* ensures that the membership of the REL set is identified with the *index* value of the MOD feature and the REL feature does not propagate any higher because the NP containing it is not the head daughter. The INDEX value of MOD in S is identified with the index of the N′ *"man"* and with the INDEX value of the NP *"who"*, ensuring that the argument of the *man* relation and the LIKER rôle of the *like* relation bound to the same individual.

The revisions that we have made to the clausality dimension are summarised in figure 70.

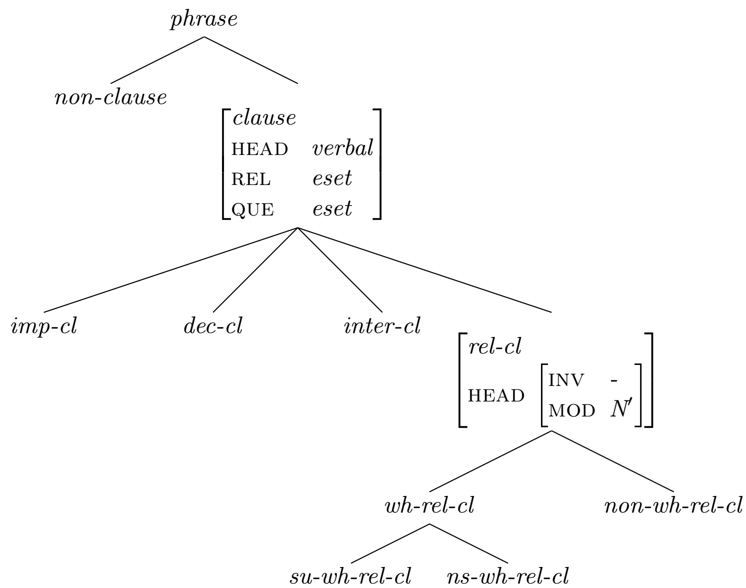


Figure 70: Revised sortal hierarchy for clausality

The analysis developed so far accounts correctly for the structure sharing of INDEX values, but the CONTENT value of a verbal projection is not of the right sort to contribute correctly to the content value of the phrase as a whole.[48] There does not seem to be any independent motivation to justify assigning verbs one kind of content in relative clauses and a different kind of content in other contexts, so an alternative is to make the required adjustment at the phrasal level by defining head-relative phrases as a distinct kind of phrase – *head-rel-phrase* – which forms a subsort of *head-adjunct-phrase*, and to impose the required constraints on *head-rel-phrase* as follows.

$$
\textit{hd-rel-ph} \quad \Longrightarrow \quad \begin{bmatrix} \text{HEAD} & \textit{noun} \\ \text{CONT} & \begin{bmatrix} \text{INDEX} & \boxed{1} \\ \text{RESTR} & \boxed{2} \uplus \{\boxed{3}\} \end{bmatrix} \\ \text{HD-DTR} & \begin{bmatrix} \text{INDEX} & \boxed{1} \\ \text{RESTR} & \boxed{2} \end{bmatrix} \\ \text{NON-HD-DTR} & \left\langle \begin{bmatrix} \text{SUBJ} & \langle\ \rangle \\ \text{CONT} & \begin{bmatrix} \text{NUCLEUS} & \boxed{3} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}
$$

The CONTENT values for the phrase *"man who likes Toby"* which result from this analysis are shown in figure 71. The CONTENT value of the whole phrase is equivalent to the FOL expression $\lambda x(man1(x) \wedge like1(x,t))$.

[48] Recall that the previous instances of N′ modifiers (adjectives and preposition) had CONTENT values of sort *nominal object*, whereas verbal projections have CONTENT values of sort *psoa*.

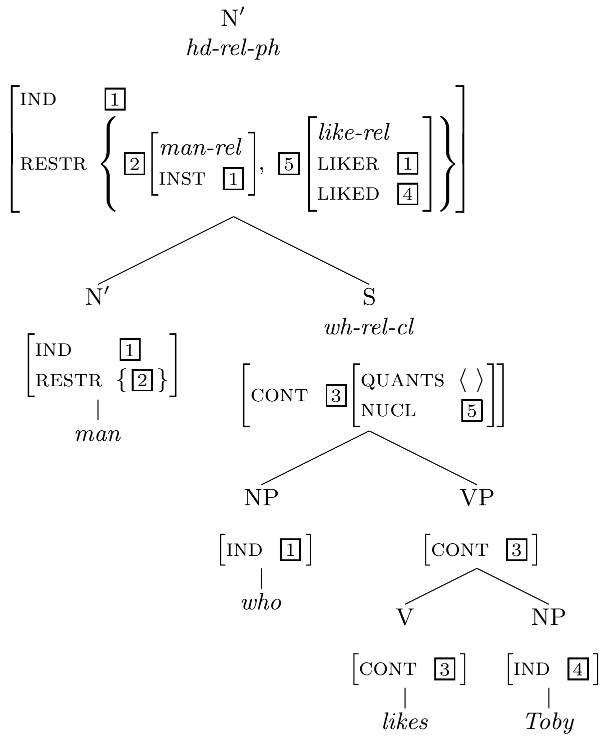


Figure 71: The CONTENT values for the N′ *"man who likes Toby"*.

The phrase *"who likes Toby"* is an example of what is called a **subject relative clause**. That is to say, the *wh*-word fills the rôle of the subject NP of the relative clause. The sentence *"who Toby likes"* is an example of an **object relative clause** – the *wh*-phrase fills the rôle of the direct object of the relative clause. Non-subject relatives introduce an additional complexity, since the relative clause contains a SLASH dependency between the *wh* filler and 'gap'. This is illustrated in figure 72.

The structure dominated by S_2 is a normal unbounded dependency construction, except that its head verb also contains the feature specification [MOD N′]. MOD and SLASH values are both inherited from the head daughter (by the Head Feature Principle and Slash Inheritance Principle respectively). The SLASH dependency is terminated in the *head-filler-phrase* dominated by S_1. The constraint on *wh-rel-cl* ensures that the content of the REL set of the filler NP *"who"* is identified with the INDEX value of the MOD feature and that the REL feature does not propagate any higher because the NP containing it is not the head daughter. The INDEX value of MOD in S_1 is identified with the index of the N′ *"man"*, ensuring that the CONTENT of the whole phrase, shown in figure 72, has the argument of the *man* relation and the LIKED rôle of the *like* relation bound to the same individual, as in the equivalent FOL expression $\lambda x(man1(x) \wedge like1(t, x))$.

This concludes our discussion of relative clauses. The analysis that we have outlined here extends without modification to other non-subject relatives, such as *"the money which Andrew gave Toby"*, and to more deeply embedded 'gaps', such as *"the money which Portia said that Andrew gave to Toby"*.

Exercise 19

Draw a tree, augmented with AVMs to show the structure of the following phrase

1. *"the man to whom Andrew gave a drink"*

Exercise 20

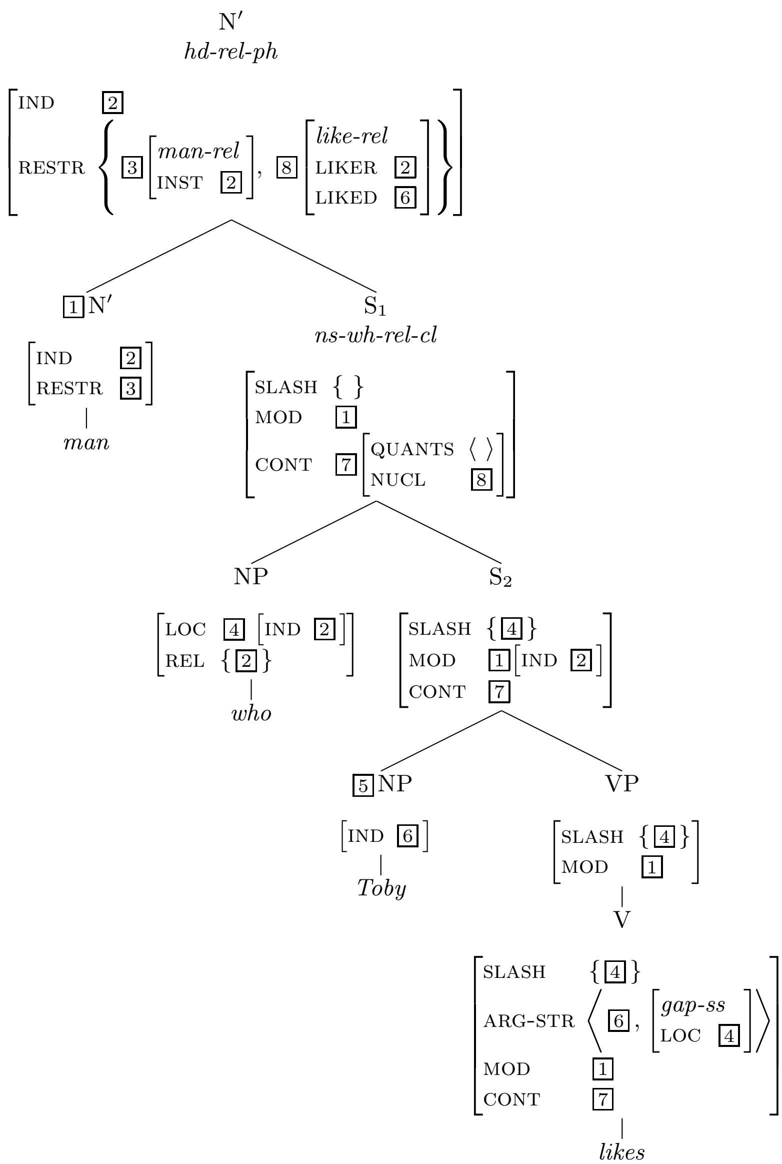


Figure 72: The N′ *"man who Toby likes"*.

Assume that the lexical entry for the possessive pronoun "my" is as shown below.[49] *Use this to provide the basis for an analysis of the phrase "a woman whose book Portia borrowed".*

$$
\begin{bmatrix}
\text{PHON} & \langle \textit{my} \rangle \\
\text{HEAD} & \begin{bmatrix} \textit{det} & \\ \text{SPEC} & \text{N}'\!:\!\begin{bmatrix} \text{IND} & \boxed{1} \\ \text{RESTR} & \boxed{2} \end{bmatrix} \end{bmatrix} \\
\text{ARG-STR} & \langle\ \rangle \\
\text{CONT} & \begin{bmatrix} \text{INDEX} & \boxed{1} \begin{bmatrix} \text{PER} & \textit{1} \\ \text{NUM} & \textit{sing} \end{bmatrix} \end{bmatrix} \\
\text{QSTORE} & \left\{ \begin{bmatrix} \textit{the-quant} & \\ \text{REST-IND} & \begin{bmatrix} \text{INDEX} & \boxed{3} \\ \text{RESTR} & \begin{bmatrix} \textit{poss-rel} & \\ \text{POSSESSOR} & \boxed{1} \\ \text{POSSESSED} & \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\}
\end{bmatrix}
$$

[49] The notation N′: is used to indicate the CONTENT value of the N′. The sort *the-quant* is the counterpart of the FOL translation of the definite determiner discussed in section 6.9.