The HPSG Formalism[*]

1 The Data Structures

1.1 AVMs

In HPSG, linguistic objects are described by typed attribute value matrices like the one shown in (1), which is part of the information lexically available for the verb *walks*.

(1)
$$\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{word}} \\
\textsc{phon} & \left\langle \textit{walks} \right\rangle \\
\textsc{synsem} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{synsem}} \\
\textsc{local} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{local}} \\
\textsc{cat} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{category}} \\
\textsc{subj} & \left\langle \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{synsem}} \\
\textsc{local} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{local}} \\
\textsc{cont} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{nom-obj}} \\
\textsc{index} & \boxed{1} \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{ref}} \\
\textsc{num} & \textit{sing} \\
\textsc{per} & \textit{3rd}
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right] \right\rangle
\end{array}\right] \\
\textsc{content} & \left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{content}} \\
\textsc{walker} & \boxed{1}
\end{array}\right]
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

The things in capital letters are the **attributes** (also called features), which are followed by their **values**. Values can either be simple (or atomic), like *sing(ular)*, which is the value of the NUM(BER) attribute, or they can be complex, i.e. their value can again be a complex attribute value matrix, like that for INDEX in the figure.

[*] Document history: first version by Susanne Riehemann (4/10/95); revised with help from Ivan Sag (5/1/95). Further revisions by Sag on 1/10/2002.

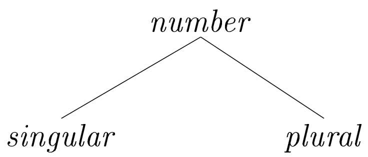
1.2 Types and Hierarchies

These attribute value matrices (AVM diagrams, or just AVMs for short) are typed (sometimes also called 'sorted') – for example, the value of the INDEX attribute is of type *referential* (abbreviated as *ref*). The type labels appear in italics in the upper left corner of an AVM. They are sometimes omitted in figures when they don't contribute relevant information.

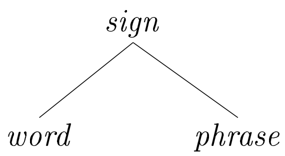
A grammar contains a specificationn of which features are appropriate for each type and what kind of value each feature must have. This is called imposing 'appropriateness conditions'. For example, things of type *index* can only have the attributes NUM(BER), PER(SON) and GEND(ER) and each of these features has certain values that are appropriate for it, e.g. *sg* and *pl* for NUM and *1st*, *2nd* and *3rd* for PER.

Some types stand in a hierarchical relationship to each other (see e.g. (2) and (3)), and since atomic values are also types (atomic types) they do, too. The relation between atomic types and other types will be clearer once we get to the graph notation.

(2)



(3)



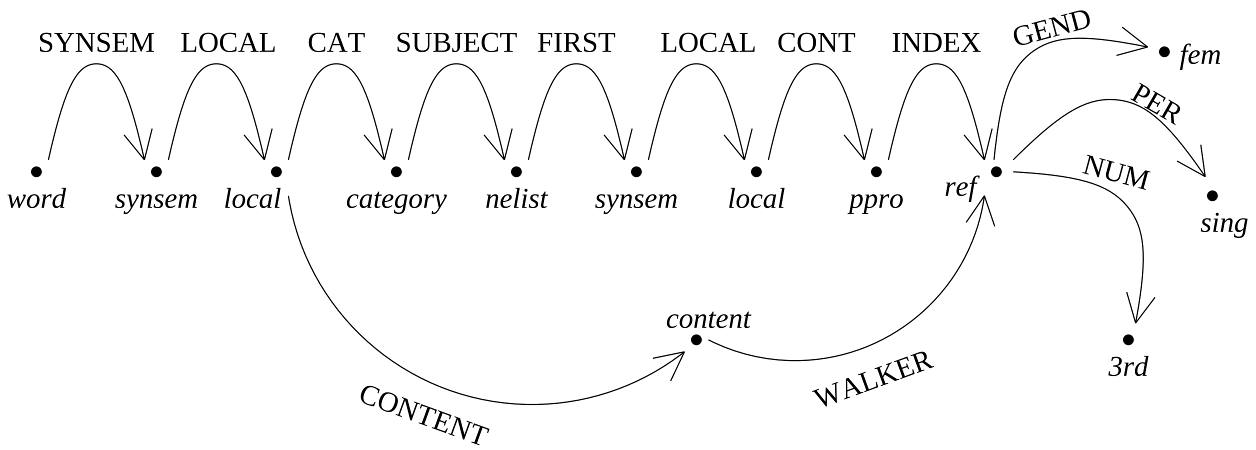
The types which contain more information are lower in the hierarchy, while the more general ones are higher up. Of course these hierarchies can get more complex when the subtypes themselves have further subtypes. Constraints that hold for several types have to be stated only once on a common supertype and are inherited by (instances of) the subtypes, which allows us to express generalizations and make our grammars and especially the lexicon more elegant.

1.3 Objects vs. Descriptions

In HPSG a clear distinction is made between linguistic objects and their descriptions – objects are assumed to be **modeled** by **feature structures**, which are usually treated as graphs, while these can be **described** by formulas that can be given in an **attribute value matrix (AVM)** notation.

For example, essentially the same information that is contained in (1) can be represented as a graph, as in (4):

(4)



In these graphs the arcs are labeled with the feature names, and the nodes are labeled with the type names. This shows the relationship between atomic types (= values) and other types – in such a graph both are labels for nodes.

These have been taken to be alternative notations for the same thing by many people (e.g. Shieber 1986 in your reading), and AVMs are often referred to as 'feature structures', but it is important to keep in mind that there is an important conceptual distinction drawn between the two in HPSG.

The feature structures are assumed to be complete models of the linguistic objects, and therefore cannot be underspecified or partial in any way. Formally, this is called 'totally well-typed and sort-resolved (or 'type-resolved')' – every feature that is admissible for a given type has to actually be present, and all their values have to belong to a maximal type. This means that for example the GENDER feature cannot be underspecified as *gend*, but has to have either *fem* or *masc* as its value when it is part of a feature structure modeling a linguistic object. So, the thing in (4) could not really be part of an HPSG feature structure – it is not fully specified (e.g. it is missing information about PHONOLOGY), and can therefore not be a feature structure; and it is in the wrong form to be a description.

1.4 Structure Sharing

The little boxed numbers in AVMs, called tags, indicate structure sharing. This means that the values of the attributes marked with these tags are supposed to be token-identical, which is stronger than merely providing these values with the same feature specifications. We'll see later that this is an important distinction, because it affects the results of unification.

The concept of 'sharing structure' can be seen quite easily in the graph notation – note that the two arcs actually lead to the same node. These feature structures are also called 'reentrant'.

2 Subsumption

AVM descriptions can be as partial as you like, and if you want to describe general properties of linguistic objects they can be very underspecified. In fact, they can be partially ordered according to their specificity. For example, (5)

(5) $\begin{bmatrix}\text{NUM} & \textit{sing}\end{bmatrix}$

is more general than (or subsumes) (6)

(6) $\begin{bmatrix}\text{NUM} & \textit{sing} \\ \text{PER} & \textit{3rd}\end{bmatrix}$

(6) contains more information, and therefore imposes stricter constraints and describes fewer linguistic objects.
The same kind of relation also holds when the types stand in a hierarchical relationship – for example, since *num* is more general than *sing*, (7) is more general than (8).

(7) $\begin{bmatrix}\text{NUM} & \textit{num}\end{bmatrix}$

(8) $\begin{bmatrix}\text{NUM} & \textit{sing}\end{bmatrix}$

But not every pair of AVMs stands in a subsumption relation – if the two AVMs are either mentioning different attributes, or else if they contain conflicting information, neither one subsumes the other.

3 Identity and Unification

IDENTITY is fundamental to HPSG (and other constraint-based grammars). A grammar contains many constraints on feature structures and these are most often stated identification of one feature value with some other feature value.
This is often put in terms of information from two or more sources (2 distinct constraints) being combined (or 'unified'). In order to compute the result of unifying the information from two AVM descriptions, one just puts together the information from both, without adding anything more. This results in the most general description that is compatible with the two inputs. The effect of this is the same as logical 'and' (&): the interpretation of 'D_1 & D_2' is just the intersection of the interpretation of D_1 and the interpretation of D_2. That is, the set of feature structures that satisfies 'D_1 & D_2' is the intersection of the satisfiers of D_1 and the satisfiers of D_2. If the information specified in D_1 is incompatible with the information specified in D_2, unification is said to 'fail' – the set of satisfiers of 'D_1 & D_2' will be empty.
For example, (9) unified with (10) gives (11) ((9) & (10) = (11)):

(9) $\begin{bmatrix}\textsc{num} & \textit{sing}\end{bmatrix}$

(10) $\begin{bmatrix}\textsc{per} & \textit{3rd}\end{bmatrix}$

(11) $\begin{bmatrix}\textsc{num} & \textit{sing}\\ \textsc{per} & \textit{3rd}\end{bmatrix}$

But (12) & (13) fails, because the values for the NUM feature are incompatible:

(12) $\begin{bmatrix}\textsc{num} & \textit{sing}\end{bmatrix}$

(13) $\begin{bmatrix}\textsc{num} & \textit{plur}\end{bmatrix}$

Now we can see more clearly why structure sharing is not the same as having the information present twice – it makes a difference when unifying something into one part of it:

(14) $$\begin{bmatrix}\mathrm{A} & \begin{bmatrix}\mathrm{B} & a\end{bmatrix}\\ \mathrm{C} & \begin{bmatrix}\mathrm{D} & \begin{bmatrix}\mathrm{B} & a\end{bmatrix}\end{bmatrix}\end{bmatrix} \;\&\; \begin{bmatrix}\mathrm{C} & \begin{bmatrix}\mathrm{D} & \begin{bmatrix}\mathrm{E} & b\end{bmatrix}\end{bmatrix}\end{bmatrix} = \begin{bmatrix}\mathrm{A} & \begin{bmatrix}\mathrm{B} & a\end{bmatrix}\\ \mathrm{C} & \begin{bmatrix}\mathrm{D} & \begin{bmatrix}\mathrm{B} & a\\ \mathrm{E} & b\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

But

(15) $$\begin{bmatrix}\mathrm{A} & \boxed{1}\begin{bmatrix}\mathrm{B} & a\end{bmatrix}\\ \mathrm{C} & \begin{bmatrix}\mathrm{D} & \boxed{1}\end{bmatrix}\end{bmatrix} \;\&\; \begin{bmatrix}\mathrm{C} & \begin{bmatrix}\mathrm{D} & \begin{bmatrix}\mathrm{E} & b\end{bmatrix}\end{bmatrix}\end{bmatrix} = \begin{bmatrix}\mathrm{A} & \boxed{1}\begin{bmatrix}\mathrm{B} & a\\ \mathrm{E} & b\end{bmatrix}\\ \mathrm{C} & \begin{bmatrix}\mathrm{D} & \boxed{1}\end{bmatrix}\end{bmatrix}$$

4 Notation and Abbreviations

4.1 Lists and Sets

Lists are written in angle brackets: $\langle\ \rangle$
HPSG lists are just like the standard lists you're used to – they are ordered, and the same object can be on the list twice.

Sets are written in curly brackets: { }
HPSG sets are a little different from what you're probably familiar with – it is possible for two descriptions in the set to actually describe the same object. So, a set with three descriptions in it could have three objects, two objects, or even just one object in it.

4.2 Types/Sorts

Remember that Pollard and Sag 1994 uses a different notation for types – you'll find them on the bottom corner outside the AVM bracket. This is purely a notational difference, and (16) is exactly the same description as (17):

(16)
$$\begin{bmatrix} \textit{word} & \\ \text{PHON} & \langle \textit{walks} \rangle \\ \text{SYNSEM} & \begin{bmatrix} \textit{synsem} & \\ \text{LOCAL} & \begin{bmatrix} \textit{local} & \\ \text{CAT} & \begin{bmatrix} \textit{category} & \\ \text{SUBJ} & \left\langle \begin{bmatrix} \textit{synsem} & \\ \text{LOCAL} & \begin{bmatrix} \textit{local} & \\ \text{CONT} & \begin{bmatrix} \textit{nom-obj} & \\ \text{INDEX} & \boxed{1} \begin{bmatrix} \textit{ref} & \\ \text{NUM} & \textit{sing} \\ \text{PER} & \textit{3rd} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{CONTENT} & \begin{bmatrix} \textit{content} & \\ \text{WALKER} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(17)
$${}_{\textit{word}}\begin{bmatrix} \text{PHON} & \langle \textit{walks} \rangle \\ \text{SYNSEM} & {}_{\textit{synsem}}\begin{bmatrix} \text{LOCAL} & {}_{\textit{local}}\begin{bmatrix} \text{CAT} & {}_{\textit{category}}\begin{bmatrix} \text{SUBJ} & \left\langle {}_{\textit{synsem}}\begin{bmatrix} \text{LOCAL} & {}_{\textit{local}}\begin{bmatrix} \text{CONT} & {}_{\textit{nom-obj}}\begin{bmatrix} \text{INDEX} & \boxed{1}\; {}_{\textit{ref}}\begin{bmatrix} \text{NUM} & \textit{sing} \\ \text{PER} & \textit{3rd} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix} \\ \text{CONTENT} & {}_{\textit{content}}\begin{bmatrix} \text{WALKER} & \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

4.3 Paths

It is common to leave out sort labels when they are not necessary and to avoid brackets by using a path notation within AVMs. If you compare (18) to (17) (=1) you will see that for example SYNSEM|LOCAL has been put together, since the bracketing would not be helpful because there is only one attribute mentioned. Such paths can get as long as you like, e.g. you can say that *sing* is the value of SYNSEM|LOCAL|CAT|SUBJ|LOCAL|CONT|INDEX|NUM

(18)
$$\begin{bmatrix} \text{PHON} & \langle \textit{walks} \rangle \\ \text{SYNSEM}\,|\,\text{LOCAL} & \begin{bmatrix} \text{CAT}\,|\,\text{SUBJ} & \left\langle \text{LOCAL}\,|\,\text{CONT} \begin{bmatrix} \textit{nom-obj} & \\ \text{INDEX} & \boxed{1}\begin{bmatrix}\textit{ref} & \\ \text{NUM} & \textit{sing} \\ \text{PER} & \textit{3rd}\end{bmatrix}\end{bmatrix}\right\rangle \\ \text{CONTENT} & \begin{bmatrix}\text{WALKER } \boxed{1}\end{bmatrix} \end{bmatrix}\end{bmatrix}$$

4.4 Abbreviations

Some further common abbreviations are:

Abbreviation:	Abbreviated AVM Diagram:
$\text{NP}_{\boxed{i}}$	$\begin{bmatrix}\text{LOCAL} & \begin{bmatrix}\text{CATEGORY} & \begin{bmatrix}\text{HEAD} & \textit{noun} \\ \text{SPR} & \langle\,\rangle \\ \text{COMPS} & \langle\,\rangle\end{bmatrix} \\ \text{CONTENT}\,|\,\text{INDEX} & \boxed{i}\end{bmatrix}\end{bmatrix}$
S:$\boxed{i}$	$\begin{bmatrix}\text{LOCAL} & \begin{bmatrix}\text{CATEGORY} & \begin{bmatrix}\text{HEAD} & \textit{verb} \\ \text{SUBJ} & \langle\,\rangle \\ \text{COMPS} & \langle\,\rangle\end{bmatrix} \\ \text{CONTENT} & \boxed{i}\end{bmatrix}\end{bmatrix}$
VP:$\boxed{i}$	$\begin{bmatrix}\text{LOCAL} & \begin{bmatrix}\text{CATEGORY} & \begin{bmatrix}\text{HEAD} & \textit{verb} \\ \text{SUBJ} & \langle\ \textit{synsem}\ \rangle \\ \text{COMPS} & \langle\,\rangle\end{bmatrix} \\ \text{CONTENT} & \boxed{i}\end{bmatrix}\end{bmatrix}$

$\langle\ \textit{synsem}\ \rangle$ means that the list has only one element, which is of type *synsem*.
Alternative notations for this are $\langle\ [\textit{synsem}]\ \rangle$, $\langle\ [\quad]\ \rangle$ or $\langle\ \text{X}\ \rangle$
In general, subscripts are always taken to refer to the INDEX value, and tags after a colon refer to the CONTENT value.

Sometimes more information is added about the INDEX:

$$\mathrm{NP}_{\boxed{i}[\mathit{3rd},\mathit{sing}]} \qquad \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CATEGORY} & \begin{bmatrix} \text{HEAD} & \mathit{noun} \\ \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{CONTENT} \mid \text{INDEX} & \boxed{i}\begin{bmatrix} \text{PER} & \mathit{3rd} \\ \text{NUM} & \mathit{sing} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Sometimes the values of frequently needed attributes such as CASE or VFORM are put in square brackets behind the abbreviation, e.g.:

$$\mathrm{NP}[\mathit{nom}] \qquad \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CATEGORY} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \mathit{noun} & \\ \text{CASE} & \mathit{nom} \end{bmatrix} \\ \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\mathrm{VP}[\mathit{fin}] \qquad \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CATEGORY} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \mathit{verb} & \\ \text{VFORM} & \mathit{fin} \end{bmatrix} \\ \text{SUBJ} & \langle\ \mathit{synsem}\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

These abbreviation are often combined:

$$\mathrm{NP}[\mathit{nom}]_{\boxed{i}[\mathit{3rd},\mathit{sing}]} \qquad \begin{bmatrix} \text{LOCAL} & \begin{bmatrix} \text{CATEGORY} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \mathit{noun} & \\ \text{CASE} & \mathit{nom} \end{bmatrix} \\ \text{SPR} & \langle\ \rangle \\ \text{COMPS} & \langle\ \rangle \end{bmatrix} \\ \text{CONTENT} \mid \text{INDEX} & \boxed{i}\begin{bmatrix} \text{PER} & \mathit{3rd} \\ \text{NUM} & \mathit{sing} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

5 Changes from HPSG II to the 5/1/95 Version of HPSG III

5.1 Valence

In HPSG II, i.e. the first 8 chapters of Pollard and Sag 1994, we have a SUBCAT list:

$$\left[\textsc{synsem}\,|\,\textsc{local}\,|\,\textsc{category}\,|\,\textsc{subcat} \quad \textit{list(synsem)}\right]$$

For example, a simple transitive verb would have first the subject and then the object on the SUBCAT list:

$$\left[\textsc{synsem}\,|\,\textsc{local}\,|\,\textsc{category}\,|\,\textsc{subcat} \quad \langle\ \mathrm{NP}[\textit{nom}],\ \mathrm{NP}[\textit{acc}]\ \rangle\right]$$

In HPSG III this is replaced by separate attributes for subjects, complements, and specifiers:

$$\left[\textsc{synsem}\,|\,\textsc{local}\,|\,\textsc{category} \quad \left[\begin{array}{ll} \textsc{subj} & \textit{list(synsem)} \\ \textsc{comps} & \textit{list(synsem)} \\ \textsc{spr} & \textit{list(synsem)} \end{array}\right]\right]$$

For example:

$$\left[\textsc{synsem}\,|\,\textsc{local}\,|\,\textsc{category} \quad \left[\begin{array}{ll} \textsc{subj} & \langle \mathrm{NP}[\textit{nom}]\ \rangle \\ \textsc{comps} & \langle \mathrm{NP}[\textit{acc}]\ \rangle \end{array}\right]\right]$$

5.2 DTRS

The attribute DTRS has been eliminated, and the attributes HD-DTR, SUBJ-DTR, COMP-DTRS, and SPR-DTR have become direct attributes of *phrases* (for details, see your "hierarchy of headed phrases" handout).

5.3 ARG-ST

There is a new attribute ARG-ST for 'argument structure', which takes a list of *synsem* objects as its value. For now make it an attribute at the same level as the valence features SUBJ, COMPS, and SPR:

$$\left[\text{SYNSEM} \mid \text{LOCAL} \mid \text{CATEGORY} \quad \begin{bmatrix} \text{HEAD} & \textit{head} \\ \text{SUBJ} & \textit{list(synsem)} \\ \text{COMPS} & \textit{list(synsem)} \\ \text{SPR} & \textit{list(synsem)} \\ \text{ARG-ST} & \textit{list(synsem)} \end{bmatrix}\right]$$

5.4 Relations

The information about the semantic relation of a verb got moved from being the value of an attribute RELATION to being a subsort of *content*.

$$\left[\text{SYNSEM} \mid \text{LOCAL} \mid \text{CONT} \quad \begin{bmatrix} \text{RELATION} & \textit{love} \\ \text{LOVER} & \\ \text{LOVED} & \end{bmatrix}\right]$$

becomes:

$$\left[\text{SYNSEM} \mid \text{LOCAL} \mid \text{CONT} \quad \begin{bmatrix} \textit{love-rel} \\ \text{LOVER} \\ \text{LOVED} \end{bmatrix}\right]$$