The building blocks of HPSG grammars

1. From a linguistic perspective
2. From a formal perspective

HPSG grammars from a linguistic perspective

From a linguistic perspective, an HPSG grammar consists of

a) a lexicon licensing basic words
b) lexical rules licensing derived words
c) immediate dominance (ID) schemata licensing constituent structure
d) linear precedence (LP) statements constraining word order
e) a set of grammatical principles expressing generalizations about linguistic objects

Why implement an HPSG theory?

Implementing grammars can be very valuable in terms of

a) providing feedback for a rigid and complete formalization of a linguistic theory.
b) stimulating system development to enhance the link between theory and implementation and to improve performance.

For this to work,

- all differences between a linguistic theory and its implementation need to be documented, and
- the system should support a clear, tractable, and formally meaningful way of implementing close to the linguistic theory.

Recoding a linguistic theory in terms of some unrelated or lower-level computer language makes it difficult to obtain meaningful feedback from the implementation for linguistics.
HPSG grammars from a formal perspective

From a formal perspective, an HPSG grammar consists of

- the **signature** as declaration of the domain, and
- the **theory** constraining the domain.

Models of linguistic objects

- The objects are modelled by feature structures, which are depicted as directed graphs.

- Since these models represent objects in the world (and not knowledge about the world), they are total with respect to the ontology declared in the signature. Technically, one says that these feature structures are
  - **totally well-typed**: Every node has all the attributes appropriate for its type and each attributes has an appropriate value.
  - **sort-resolved**: Every node is of a maximally specific type.

The signature

- defines the ontology (‘declaration of what exists’):
  - which kind of objects are distinguished, and
  - which properties of which objects are modelled.

- consists of
  - the type (or sort) hierarchy and
  - the appropriateness conditions, defining which type has which appropriate attributes (or features) with which appropriate values.

Descriptions

A description language and its abbreviating **AVM notation** is used to talk about sets of objects. Descriptions consists of three building blocks:

- **Type** descriptions single out all objects of a particular type, e.g., word

- **Attribute-value pairs** describe objects that have a particular property. The attribute must be appropriate for the particular type of object, and the value can be any kind of description, e.g., `[spouse | name mary]`

- **Tags** (structure sharing) to specify **token identity**, e.g. □
Descriptions (cont.)

Complex descriptions are obtained by combining descriptions with the help of conjunction ($\land$), disjunction ($\lor$) and negation ($\lnot$). In the AVM notation, conjunction is implicit.

A **theory** (in the formal sense) is a set of description language statements, often referred to as the constraints.

- The theory singles out a subset of the objects declared in the signature, namely those which are grammatical.
- A linguistic object is admissible with respect to a theory iff it satisfies each of the descriptions in the theory and so does each of its substructures.

Motivating SUBCAT

(1) a. *I laugh. ($<$NP$>$)
   b. I saw him. ($<$NP NP$>$)
   c. I give her the book. ($<$NP NP NP$>$)
   d. I said that she left. ($<$NP S[that]$>$)

Cannot always be derived from semantics:

(2) a. Paul ate a steak. ($<$NP$>$)
   b. * Paul ate. ($<$NP NP$>$)
(3) a. Paul devoured a steak. ($<$NP$>$)
   b. * Paul devoured

An ontology of linguistic objects

```
<table>
<thead>
<tr>
<th>sign</th>
<th>list(phonstring)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHON</td>
<td>SYNSEM synsem</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

word

```

<table>
<thead>
<tr>
<th>synsem</th>
<th>local</th>
<th>non-local</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCAL</td>
<td>category</td>
<td>content</td>
</tr>
<tr>
<td>context</td>
<td>category</td>
<td>content</td>
</tr>
<tr>
<td>CONTEXT</td>
<td>category</td>
<td>content</td>
</tr>
<tr>
<td></td>
<td>HEAD head</td>
<td>SUBCAT</td>
</tr>
<tr>
<td></td>
<td>list(synsem)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Part-of-speech

```

<table>
<thead>
<tr>
<th>head</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

| functional |
|           |
|           |

| substantive |
|            |
|            |

<table>
<thead>
<tr>
<th>verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>VFORM</td>
</tr>
<tr>
<td>vform</td>
</tr>
<tr>
<td>AUX</td>
</tr>
<tr>
<td>INV</td>
</tr>
<tr>
<td>boolean</td>
</tr>
<tr>
<td>boolean</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>noun</th>
</tr>
</thead>
<tbody>
<tr>
<td>CASE</td>
</tr>
</tbody>
</table>

| preposition |
|            |
| PFORM pform|
|            |

```
Motivating CASE

(7) a. He left.
   b. * Him left.

(8) a. She sees him.
   b. * She sees he.

Motivating VFORM

(4) a. Peter will win the race. (base form)
    b. * Peter will won the race.
    c. * Peter will to win the race.

(5) a. Peter has won the race. (past participle)
    b. * Peter has win the race.
    c. Peter has to win the race. (→ different verb)

(6) a. Peter seems to win the race. (to-infinitive)
    b. * Peter seems win the race.
    c. * Peter seems won the race.

Indices

- index
- person
- number
- gender
- referential
- there
- it
- first
- second
- third
- singular
- plural
- masculine
- feminine
- neuter
### Semantic representations

![Semantic representations diagram](image)

### Auxiliary data structures

![Auxiliary data structures diagram](image)

### Abbreviations for describing lists

- `empty-list` is abbreviated as `e-list, <>`
- `non-empty-list` is abbreviated as `ne-list`
- `\langle \ldots \rangle` is abbreviated as `\langle \ldots \rangle`
- `\langle \text{HEAD} \text{T} \text{TAIL} \text{TAIL} \rangle` is abbreviated as `\langle \text{HEAD} \text{TAIL} \text{TAIL} \rangle`

Attention: `\langle T \rangle` and `\langle B \rangle` describe all lists of length one!
### Abbreviations of common AVMs

Pollard and Sag (1994) use some abbreviations to describe synsem objects:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Abbreviated AVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>synsem</td>
</tr>
<tr>
<td>S</td>
<td>synsem</td>
</tr>
<tr>
<td>VP</td>
<td>synsem</td>
</tr>
</tbody>
</table>

The basic lexicon is defined by the *Word Principle* as part of the theory. It defines which of the ontologically possible words are grammatical:

\[
\text{word} \rightarrow \text{lexical-entry}_1 \lor \text{lexical-entry}_2 \lor \ldots
\]

with each of the lexical entries being descriptions, such as e.g.:

\[
\begin{array}{l}
\text{word} \\ 
\text{PHON }<\text{laughs}> \\ 
\text{SYNSEM}[\text{LOC}]
\end{array}
\]

\[
\begin{array}{l}
\text{word} \\ 
\text{PHON }<\text{drinks}> \\ 
\text{SYNSEM}[\text{LOC}]
\end{array}
\]

\[
\begin{array}{l}
\text{word} \\ 
\text{PHON }<\text{drinker}> \\ 
\text{SYNSEM}[\text{LOC}]
\end{array}
\]

\[
\begin{array}{l}
\text{word} \\ 
\text{PHON }<\text{drunken}> \\ 
\text{SYNSEM}[\text{LOC}]
\end{array}
\]
A very first sketch of an example

she

drinks wine

An ontology of phrases

constituent-structure

head-struc

head-marker-struc

head-adjunct-struc

coordinate-structure

sign

phrase

phrase

word

elast
Subcat Principle:

The symbol \(<\) standing for list concatenation, i.e., \(\text{append}\), defined as follows:

\[
\begin{align*}
e\text{-list} & : = \text{first } + \text{list} \\
\text{first } & : = \text{first} \\
\text{list } & : = \text{rest } + \text{list}.
\end{align*}
\]

An auxiliary relation: synsem2sign/2

The call to \(\text{synsem2sign/2}\) is needed to relate the synsem objects on the subcat to sign objects containing those synsem objects as value of their SYNSEM attribute. It is defined as follows:

\[
\begin{align*}
\text{synsem2sign}(\text{e-list}) & : = \text{e-list}. \\
\text{synsem2sign}(\text{first}) & : = \text{first}\left[\text{synsem}\right].
\end{align*}
\]
Immediate Dominance Principle (for English):

\[
\text{phrase \quad \text{[DTRs headed-struct] \rightarrow \quad \mathbf{\vdots}}
\]

\[
\mathbf{\vdots}
\]

\[
\mathbf{\vdots}
\]

\[
\text{DTRs \quad \text{head-mark-struct}} \quad \text{[MARKER-DTR[SYNSEM|LOC|CAT|HEAD marked]}} \quad \text{(Head-Marker)}
\]

\[
\text{DTRs \quad \text{head-adjunct-struct}} \quad \text{[ADJ-DTR[SYNSEM|LOC|CAT|HEAD|MOD \text{mod}]}} \quad \text{(Head-Adjunct)}
\]

\[
\text{\ldots continued on next page}
\]

Lexical entry of an attributive adjective

Version without redundant specifications
Sketch of an example for a head-adjunct structure

Sketch of an example with an inverted auxiliary

Sketch of an example with an auxiliary

SPEC Principle:

\[
\text{phrase} \rightarrow \text{DTRS} \left( \text{MARKER-DTR} \lor \text{COMP-DTR}[\text{FIRST}] \right) \text{SYNSEM}[\text{LOC}\text{CAT}][\text{HEAD functional}] \]

\[
\rightarrow \text{DTRS} \left( \text{MARKER-DTR} \lor \text{COMP-DTR}[\text{FIRST}] \right) \text{SYNSEM}[\text{LOC}\text{CAT}][\text{HEAD} \text{SPEC }] \text{[SPEC } \right)
\]
Marking Principle:

\[
\text{phrase} \Rightarrow \begin{cases} 
\text{SYNSEM|LOC|CAT|MARKING} & \text{head-mark-struc} \\
\text{DTRS} & \text{ MARKER-DTR|SYNSEM|LOC|CAT|MARKING} \\
\text{SYNSEM|LOC|CAT|MARKING} & \text{¬head-mark-struc} \\
\text{DTRS} & \text{ HEAD-DTR|SYNSEM|LOC|CAT|MARKING} 
\end{cases}
\]

Sketch of an example for a head-marker structure

Lexical entry of the marker *that*

\[
\text{PHON} <\text{that}> \\
\text{SYNSEM|LOC|CAT} \\
\text{HEAD} \text{ SPEC LOC|CAT} \\
\text{SUBCAT} \text{ MARKING} <\text{that}>
\]

\[
\text{PHON} <\text{John,laughs}> \\
\text{SYNSEM|LOC|CAT} \\
\text{HEAD} \text{ SPEC LOC|CAT} \\
\text{SUBCAT} <\text{John}>
\]

\[
\text{PHON} <\text{John}> \\
\text{SYNSEM|LOC|CAT} \\
\text{HEAD} \text{ SPEC LOC|CAT} \\
\text{SUBCAT} <\text{John}>
\]

\[
\text{PHON} <\text{John}> \\
\text{SYNSEM|LOC|CAT} \\
\text{HEAD} \text{ SPEC LOC|CAT} \\
\text{SUBCAT} <\text{John}>
\]