Head-Driven Phrase Structure Grammar (HPSG)

The main mechanism of HPSG for linguistic analysis lies in its feature architecture:
- Trees, bar-level distinctions, subcategorization, agreement, etc. are handled via feature structures.

Rough comparison to LFG:
- Similar: use of feature structures, highly lexical
- Different: only one layer of syntax, functions are not primitives, stronger reliance on structure-sharing

Preview: lexical entry for letter

(This entry more or less follows Carnie (2013) / Sag & Wasow (1999), which most of the rest of the slides don’t.)
Introduction

**Phrases Raising/Equi UDCs**

**Motivation**

**Unpacking letter**

- SYN: holds syntactic information (vs. PHON or SEM)
- SYNSEM is also commonly used (e.g., these slides)
- HEAD: category & inflectional properties
- SPR & COMPS: subcategorization properties
  - Sometimes 3 lists (SUBJ, SPR, COMPS), sometimes 1 list (SUBCST)
  - Note that detailed properties of selected items can be specified

Components: from a linguistic perspective

**HPSG grammars**

An HPSG grammar, from a linguistic perspective, 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

Lexical rules

e.g., plural rule (adapted from Carnie 2013):

```
[ | PRON | PRON | ]
[ | ARG-ST (COUNT s) | ]
```

⇒

```
[ | word | syn | head | agr | num | pl | ]
```

Any feature not specified is unchanged
**Grammatical principles**

Grammatical principles allow one to generalize over entire classes of objects
e.g., every complete sentence in English is tensed

These can be interpreted as constraints on the acceptable feature structures
- HPSG is constraint-based: grammars are sets of constraints which hold simultaneously
- This determines the collections of admissible linguistic structures
  - Constraints do not define an order of the derivation or generation of linguistic signs

---

**HPSG grammars**

An HPSG grammar formally consists of
- the **signature** as declaration of the domain, and
- the **theory** constraining the domain.

---

**The signature of an HPSG grammar**

The **signature**
- defines the ontology ('declaration of what exists'):
  - which kind of objects are distinguished, and
  - which properties of which objects are modeled.
- consists of
  - the **type hierarchy** (or sort hierarchy) and
  - the **appropriateness conditions**, defining which type has which appropriate attributes (or features) with which appropriate values.
**Models of linguistic objects**

What do the mathematical structures used as model for HPSG theories look like?

- The objects are modeled by typed feature structures, which can be notated as directed graphs.
- These models represent objects in the world, so they are total with respect to the ontology declared in the signature.
- Naturally, knowledge about objects in the world is partial.
- Formally speaking, the feature structures used as models are:
  - **totally well-typed**: Every type has every one of the attributes and their values which are appropriate for it.
  - **sort-resolved**: Every type is maximally specific.

Type and sort as well as attribute and feature are used synonymously.

**An example for a typed feature structure**
The theory of an HPSG grammar

A **theory** 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.

How do we express a theory? Descriptions

- A **description language** describes sets of objects.
- The **AVM** (attribute value matrix) notation is used to compactly write down these descriptions.
- Descriptions consists of three building blocks:
  - **Type** descriptions single out all objects of a given type
    - Example: `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.
    - Example: `[spouse [name mary]]`
  - **Tags** (structure sharing) to specify token identity
    - Example: 

Descriptions (cont.)

- Complex descriptions are obtained through
  - conjunction (`∧`)
  - disjunction (`∨`)
  - negation (`¬`)
- In the AVM notation, conjunction is implicit.
An example AVM - The pronoun *she*

```
<table>
<thead>
<tr>
<th>PHON</th>
<th>&lt;she&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAT</td>
<td>local</td>
</tr>
<tr>
<td>SYNSEM</td>
<td>LOC</td>
</tr>
<tr>
<td>CONT</td>
<td></td>
</tr>
<tr>
<td>CONTEXT</td>
<td>BACKGR</td>
</tr>
</tbody>
</table>
```

Aspects of the signature

**Signs**

Aspects of the signature

```
sign
  PHON  list(phonstring)
  SYNSEM syssem
```

```
word
  phrase
    DTBS constituent-structure
```

```
syssem
  local
  non-local
```

```
local
  category
  CONTENT content
  CONTEXT context
```

```
category
  head
  subcat list(syssem)
  marking marking
```

Motivating subcat

1. a. *I laugh.*
   b. *I saw him.*
   c. *I give her the book.*
   d. *I said that she left.*

Cannot always be derived from semantics:

2. a. *Paul ate a steak.*
   b. *Paul ate.*

3. a. *Paul devoured a steak.*
   b. *Paul devoured*
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.
Motivating CASE

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

(8) a. She sees him. (acc)
    b. * She sees he.
### Auxiliary data structures

Alternative names for the attributes `first (ft)` and `rest (rt)` of `non-empty-list` are `head (hd)` and `tail (tl)`.

### Abbreviations for describing lists

- `empty-list` is abbreviated as `e-list, <>`
- `non-empty-list` is abbreviated as `ne-list`
- `⟨first | rest⟩` is abbreviated as `⟨1 | 2⟩`
- `⟨(... | i)⟩` is abbreviated as `⟨. . . | i⟩`
- `⟨first | rest⟩` is abbreviated as `⟨hd | tl⟩`

### Abbreviations for common AVMs

Pollard and Sag (1994) make use of the following abbreviations for describing `synsem` objects:

<table>
<thead>
<tr>
<th>Abbrev.</th>
<th>abbreviated AVM</th>
</tr>
</thead>
</table>
| `NP`    | `synsem`
| `S`     | `synsem`
| `VP`    | `synsem`
Theory: The Lexicon

The basic lexicon is defined by the Word Principle as part of the theory. It is an implicational statement defining 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, e.g.:

\[
\begin{array}{c}
\text{word} \\
\text{PHON} < \text{laughs}> \\
\text{SYNSEM/LOC} \begin{cases}
\text{HEAD} \\
\text{SUBCAT} \{ \text{NP}[\text{nom}_1], \text{3rd sing} \} \\
\text{CONTENT} < \text{laughs}_1>
\end{cases}
\end{array}
\]

An example lexicon

\[
\begin{array}{c}
\text{word} \rightarrow \\
\text{PHON} < \text{drinks}> \\
\text{SYNSEM/LOC} \begin{cases}
\text{HEAD} \\
\text{SUBCAT} \{ \text{NP}[\text{nom}_1], \text{3rd sing}, \text{NP}[\text{acc}_1], \text{plur} \} \\
\text{CONTENT} < \text{drinks}_1>
\end{cases}
\end{array}
\]

\[
\begin{array}{c}
\text{V} \\
\text{PHON} < \text{give}> \\
\text{SYNSEM/LOC} \begin{cases}
\text{HEAD} \\
\text{SUBCAT} \{ \text{NP}[\text{nom}_1], \text{plur}, \text{NP}[\text{acc}_1], \text{plur} \} \\
\text{CONTENT} < \text{give}_1>
\end{cases}
\end{array}
\]
Different kinds of phrasal constructions in HPSG

For phrases, the DTRS feature has a constituent-structure type:
The structure of a simple phrase

Sketch of an example for head-complement structures

Sketch of an example for head-complement structures with a ditransitive verb
Head-Feature Principle:

The head value of any headed phrase is structure-shared with the head value of the head daughter.

\[
\begin{align*}
\text{phrase} & \quad \text{dtrs} \quad \text{headed-structure} \\
\text{SYNSEM} & \quad \text{LOC} \quad \text{CAT} \quad \text{HEAD} \\
\text{dtrs} & \quad \text{HEAD-DTRS} \quad \text{SYNSEM} \quad \text{LOC} \quad \text{CAT} \quad \text{HEAD}
\end{align*}
\]

Subcategorization Principle:

In a headed phrase (i.e. a phrasal sign whose dtrs value is of sort head-struc), the subcat value of the head daughter is the concatenation of the phrase's subcat list with the list (in order of increasing obliqueness) of synsem values of the complement daughters.

Subcat Principle: (cont.)

\[
\begin{align*}
\text{dtrs} & \quad \text{headed-structure} \\
\text{SYNSEM} & \quad \text{LOC} \quad \text{CAT} \quad \text{SUBCAT} \\
\text{dtrs} & \quad \text{HEAD-DTRS} \quad \text{SYNSEM} \quad \text{LOC} \quad \text{CAT} \quad \text{SUBCAT} \\
\text{COMP-DTRS} & \quad \text{synsem2sign}(2)
\end{align*}
\]

with \(\oplus\) standing for list concatenation, i.e., append, defined as follows

\[
\begin{align*}
e\text{-list} & \quad \oplus \quad e\text{-list} := \quad e\text{-list} \\
\text{FIRST} & \quad \oplus \quad \text{FIRST} := \quad \text{FIRST} \\
\text{REST} & \quad \oplus \quad \text{REST} := \quad \text{REST}
\end{align*}
\]

and \(\text{synsem2sign}\) encoding conversion of synsem to sign lists defined as follows

\[
\begin{align*}
\text{synsem2sign}(e\text{-list}) & := \quad e\text{-list} \\
\text{FIRST} & := \quad \text{FIRST} \\
\text{REST} & := \quad \text{REST} \\
\text{synsem2sign} & := \quad \text{synsem2sign}
\end{align*}
\]
### Immediate Dominance Principle (for English):

\[
\begin{align*}
\text{phrase} & \rightarrow \\
& \begin{cases}
\text{SYNSEM}\{\text{LOC}\}\{\text{CAT}\} \\
\text{HEAD} \quad \left[ \begin{array}{l}
\text{verb} \quad \text{INV} \\
\text{subcat} \quad \rangle
\end{array} \right] \\
\text{head-comp-struc} \\
\text{head-dtr} \\
\text{phrase} \\
\text{comp-dtrs} \quad \langle \text{sign}\rangle \\
\text{dtrs} \\
\end{cases} \\
\text{(Head-Subject)} \\
\text{∨} \\
\begin{cases}
\text{SYNSEM}\{\text{LOC}\}\{\text{CAT}\} \\
\text{HEAD} \quad \left[ \begin{array}{l}
\text{verb} \quad \text{INV} \\
\text{subcat} \quad \rangle
\end{array} \right] \\
\text{head-comp-struc} \\
\text{head-dtr} \\
\text{word} \\
\text{dtrs} \\
\end{cases} \\
\text{(Head-Complement)}
\end{align*}
\]

### Head-Subject Phrases (Schema 1)

\[
\begin{align*}
\begin{cases}
\text{SYNSEM}\{\text{LOC}\}\{\text{CAT}\} \\
\text{HEAD} \quad \langle \text{mod} \rangle \\
\text{dtrs} \\
\end{cases} \\
\text{(Head-Adjunct)}
\end{align*}
\]
Head-Complement Phrases (Schema 2)

Head-Subject-Comp Phrases (Schema 3)

Sketch of an example with an auxiliary
Introduction  Signs  Phrases  Raising/Equi UDCs  Immediate Dominance Schemata

Sketch of an example with an inverted auxiliary

Complementizer as heads

Some verbs select a complementized sentential complement headed by a verb with a specific verb form:

(9) I demand that he leave/*leaves immediately.

Markers
Marking Principle:

In a headed structure, the MARKING value coincides with that of the marker daughter if there is one, and with that of the head daughter otherwise.

\[
\text{phrase} \quad \text{headed-structure} \rightarrow \\
\text{SYNSEM} | \text{LOC} | \text{CAT} | \text{MARKING} 1 \\
\text{head-mark-struc} \quad \text{DTRS} \\
\text{MARKER-DTRS} | \text{SYNSEM} | \text{LOC} | \text{CAT} | \text{MARKING} 1 \\
\lor \\
\text{SYNSEM} | \text{LOC} | \text{CAT} | \text{MARKING} 1 \\
\text{head-dtr} | \text{DTRS} \\
\text{HEAD} | \text{SYNSEM} | \text{LOC} | \text{CAT} | \text{MARKING} 1 \\
\]

Lexical entry of the marker that

Sketch of an example for a head-marker structure
**SPEC Principle:**

If a nonhead daughter in a headed structure bears a SPEC value, it is token-identical to the SYNSEM value of the head daughter.

\[ \text{phrase} \rightarrow \text{dtrs} \left( \text{marker-dtr} \lor \text{comp-dtrs} \mid \text{first} \right) \mid \text{synsem} \mid \text{loc} \mid \text{cat} \mid \text{head} \mid \text{spec} \]

**Determiners**

Common nouns subcategorize for their determiners:

\[
\text{PHON} \langle \text{book} \rangle \rightarrow \text{head} \left( \text{cat} \left( \text{loc} \left( \text{cat} \left( \text{head} \left( \text{det} \text{spec} \left( \text{loc} \left( \text{cat} \langle \rangle \rangle \right) \right) \right) \right) \right) \right) \right)
\]

**Determiners (cont.)**

Determiners select their nominal sister:

\[
\text{PHON} \langle \text{every} \rangle \rightarrow \text{det} \left( \text{cat} \left( \text{spec} \left( \text{loc} \left( \text{cat} \left( \text{head} \text{noun} \text{spec} \left( \text{det} \text{P} \right) \right) \right) \right) \right) \right)
\]
Sketch of the head-comp phrase *every book*

Lexical entry for the possessive pronoun *my*

Lexical entry for possessive *'s*
Sketch of an example for a determiner phrase

Lexical entry of an attributive adjective

Semantics Principle

In a headed phrase, the CONTENT value is token-identical to that of the adjunct daughter if the DTRS value is of sort head-adj-struc, and with that of the head daughter otherwise.
**Introduction**

**Signs**

**Phrases**

**Raising/Equi**

**UDCs**

**Adjuncts**

Sketch of an example for a head-adjunct structure

```
A
 \[\begin{array}{c}
\text{PHON}\bigg(\text{cont, book}\bigg)
\end{array}\bigg]\n\text{LOC}
\[\begin{array}{c}
\text{HEAD} \text{ subcat}\langle \text{NP}\rangle
\end{array}\bigg]\n\text{COST}\langle \text{B}\rangle
\]
```

Raising/Equi

Review of ID Schemata

```
\begin{align*}
\text{phrase} & \rightarrow \text{STBS headed-struc} \\
\vdash & \text{STBS} \\
\vdash & \text{SYNSEM}\text{LOC}\langle\text{CAT}angle
\begin{array}{c}
\text{HEAD} \text{ subcat}\langle \text{NP}\rangle
\end{array}\bigg]\n\text{COMP-STRUC} \text{ phrase}
\end{align*}
```

Towards an analysis: Unsaturated Complements

Where does it say in these ID schemata that every subcategorized item must be realized?

- In English, many verbs and adjectives subcategorize for an unsaturated complement.
- In other words, a complement can be specified as
  \[\text{SUBCAT}\langle \text{NP}\rangle\], rather than \[\text{SUBCAT}\langle \rangle\]
  - The Head-Subject Schema allows for this.
  - And this will give consider access to the lower subject, as well as its own subject.
### Subject oriented raising verbs

```
<table>
<thead>
<tr>
<th>word</th>
<th>PHON</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>synsem</td>
<td>local</td>
</tr>
<tr>
<td>CAT</td>
<td>HEAD</td>
</tr>
<tr>
<td>SUBCAT</td>
<td>NP</td>
</tr>
<tr>
<td>seeem</td>
<td>SOA-ARG</td>
</tr>
</tbody>
</table>
```

### Subject oriented equi verbs

```
<table>
<thead>
<tr>
<th>word</th>
<th>PHON</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>synsem</td>
<td>local</td>
</tr>
<tr>
<td>CAT</td>
<td>HEAD</td>
</tr>
<tr>
<td>SUBCAT</td>
<td>NP</td>
</tr>
<tr>
<td>try</td>
<td>TEVER</td>
</tr>
<tr>
<td>SOA-ARG</td>
<td>0</td>
</tr>
</tbody>
</table>
```

### Object oriented raising verbs

```
<table>
<thead>
<tr>
<th>word</th>
<th>PHON</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>synsem</td>
<td>local</td>
</tr>
<tr>
<td>CAT</td>
<td>HEAD</td>
</tr>
<tr>
<td>SUBCAT</td>
<td>NP</td>
</tr>
<tr>
<td>believe</td>
<td>BELIEVER</td>
</tr>
<tr>
<td>SOA-ARG</td>
<td>0</td>
</tr>
</tbody>
</table>
```
Object oriented equi verbs

Analysis of: *It wants to rain

Analysis of: It seems to rain

Analysis of: *It wants to rain
Unbounded dependency constructions (UDCs)

- To account for UDCs, we will use the feature slash.
  - Named after the categorial grammar categories (S/NP as an S missing an NP).
  - This is a non-local feature which:
    - originates with a trace,
    - gets passed up the tree,
    - and is finally bound by a filler.

The bottom of a UDC: Traces

- phonologically null, but structure-shares LOCAL and slash
- we'll talk about TO-BIND later

An example for a strong UDC

The bottom of a UDC: Traces
Traces

Because the local value of a trace is structure-shared with the slash value, constraints on the trace will be constraints on the filler.

- For example, hates specifies that its object be accusative, and this case information is local.
- So, the trace has \([\text{SYNSEM}|\text{LOCAL}|\text{CAT}|\text{HEAD}|\text{CASE}\ \text{acc}\) as part of its entry, and thus the filler will also have to be accusative.

(10) *He/, Him, John likes –

The middle of a UDC
The Nonlocal Feature Principle (NFP)

For each nonlocal feature, the inherited value on the mother is the union of the inherited values on the daughter minus the to-bind value on the head daughter.

- In other words, the slash information (which is part of inherited) percolates “up” the tree.
- This allows all the local information of a trace to “move up” to the filler.

The top of a UDC: Filler-head structures
Filler-head schema

\[
\text{phrase} \quad \begin{array}{c}
\text{DTRs} \quad \text{head-filler-struc}
\end{array} \\
\begin{array}{c}
\text{HEAD-DTR}|\text{SYNSEM} \\
\text{LOC}|\text{CAT} \\
\text{HEAD} \\
\text{SUBCAT}() \\
\text{INHERITED}|\text{SLASH} \\
\text{TO-BIND}|\text{SLASH}\end{array} \\
\begin{array}{c}
\text{verb} \\
| \text{VFORM} fn \\
| \text{fin} \\
\text{subcat}⟨⟩ \\
\text{head} \\
\text{cat} \\
\text{loc} \\
\text{nonloc} \end{array}
\begin{array}{c}
\text{filler-DTR}|\text{SYNSEM}|\text{LOCAL} \end{array}
\]
Filler and trace are identified as the exact same thing (as far as their local structure is concerned).
The trace is "bound" by the TO-BIND feature; this prevents the SLASH value from going any higher in the tree.
Only saturated finite verbs (i.e., sentences) license such structures.

Example for a structure licensed by the filler-head schema:

```
[LOCAL [NLOC INHERITED SLASH ()]]
```

The analysis of the strong UDC example:

```
[S [NLOC INHERITED SLASH ()]
 [LOCAL [NLOC INHERITED SLASH [TO-BIND SLASH {1}]]]
]
```