

Head-Driven Phrase Structure Grammar

Petr Horáček, Eva Zámečnicková and Ivana Burgetová

Department of Information Systems
Faculty of Information Technology
Brno University of Technology
Božetěchova 2, 612 00 Brno, CZ



- **Introduction**

- **Introduction**
- **Sign and AVM**

- **Introduction**
- **Sign and AVM**
- **HPSG Principles**



- **Introduction**
- **Sign and AVM**
- **HPSG Principles**
- **Lexicon**



- **Introduction**
- **Sign and AVM**
- **HPSG Principles**
- **Lexicon**
- **Examples**

- **Introduction**
- Sign and AVM
- HPSG Principles
- Lexicon
- Examples



Head-Driven Phrase Structure Grammar (HPSG)

- Generative grammar
- Non-derivational
 - No notion of deriving one structure from another (such as transformations).
- Declarative constraints
- Unification-based



Head-Driven Phrase Structure Grammar (HPSG)

- Generative grammar
 - Non-derivational
 - No notion of deriving one structure from another (such as transformations).
 - Declarative constraints
 - Unification-based
-
- Influenced by GPSG.
 - Sometimes considered a direct successor to GPSG, but there is influence from other formalisms as well (such as LFG).



Head-Driven Phrase Structure Grammar (HPSG)

- Generative grammar
 - Non-derivational
 - No notion of deriving one structure from another (such as transformations).
 - Declarative constraints
 - Unification-based
-
- Influenced by GPSG.
 - Sometimes considered a direct successor to GPSG, but there is influence from other formalisms as well (such as LFG).
 - Emphasis on precise mathematical modeling of linguistic entities.
 - Suitable for computer implementations, often used in practice in NLP.



Components of HPSG

- 1 Grammar principles
- 2 Grammar rules
- 3 Lexical entries



Components of HPSG

- 1 Grammar principles
- 2 Grammar rules
- 3 Lexical entries

- All these components are formalized as **typed feature structures**.

- Introduction
- **Sign and AVM**
- HPSG Principles
- Lexicon
- Examples

Sign

- Basic HPSG type
- **Collection of information**, including
 - Phonology
 - Syntax
 - Semantics
- Every constituent admitted by HPSG is of type sign.
- Constituents have to conform to grammatical principles.

Sign

- Basic HPSG type
- **Collection of information**, including
 - Phonology
 - Syntax
 - Semantics
- Every constituent admitted by HPSG is of type sign.
- Constituents have to conform to grammatical principles.
- Two subtypes, further conforming to different constraints.

Sign subtypes

- 1 Word
 - Conforming to lexical entries
- 2 Phrase
 - Conforming to grammar rules

- Sign is usually represented by **attribute-value matrix** (AVM).

Attribute-Value Matrix

$$\begin{bmatrix} \textit{type} \\ \text{ATTRIBUTE} & \textit{value} \\ \text{ATTRIBUTE} & \textit{value} \\ \vdots \end{bmatrix}$$

- Note: AVM notations may vary, there may be additional information, type may be omitted, . . .

- Sign is usually represented by **attribute-value matrix** (AVM).

Attribute-Value Matrix

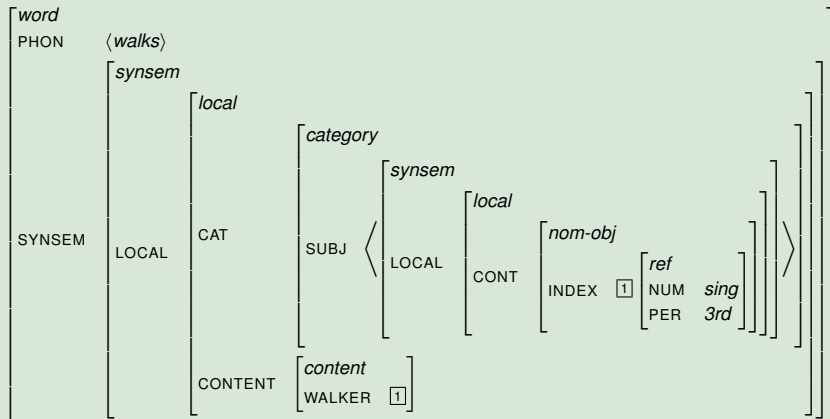
$$\begin{bmatrix} \textit{type} \\ \text{ATTRIBUTE} & \textit{value} \\ \text{ATTRIBUTE} & \textit{value} \\ \vdots \end{bmatrix}$$

- Note: AVM notations may vary, there may be additional information, type may be omitted, . . .

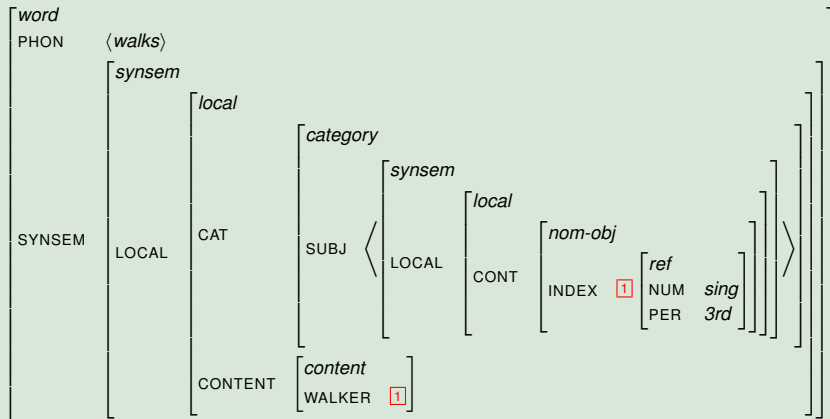
Types of Values

- 1 Atomic
- 2 Complex – the value is itself a feature structure (another AVM)

Example



Example



Coreferential tag – indicates that certain substructures are identical.

- Here, the AVM which would be the value for **WALKER** is identical to the tagged AVM in **INDEX** (number and person must match).

- PHON (phonology) – list of phonological descriptions

- PHON (phonology) – list of phonological descriptions
- SYNSEM (syntax and semantics) – another AMV of type *synsem*

SYNSEM

- HEAD – encodes syntactical features that head and its phrasal constituent have in common
 - Includes information such as part-of-speech, inflectional properties.

- PHON (phonology) – list of phonological descriptions
- SYNSEM (syntax and semantics) – another AMV of type *synsem*

SYNSEM

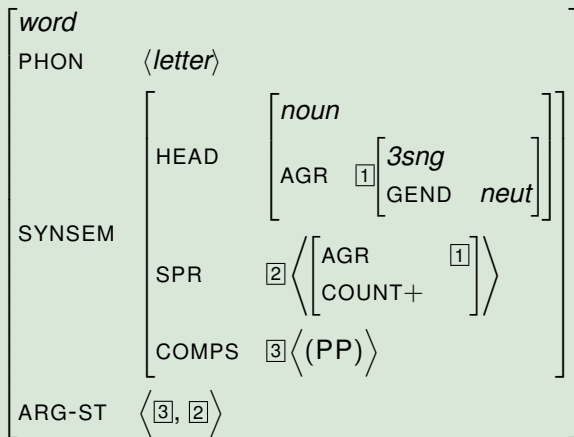
- HEAD – encodes syntactical features that head and its phrasal constituent have in common
 - Includes information such as part-of-speech, inflectional properties.
- SPR – element that may appear as the specifier in a constituent
- COMPS – elements that may appear as the complements
- ...

- PHON (phonology) – list of phonological descriptions
- SYNSEM (syntax and semantics) – another AMV of type *synsem*

SYNSEM

- HEAD – encodes syntactical features that head and its phrasal constituent have in common
 - Includes information such as part-of-speech, inflectional properties.
 - SPR – element that may appear as the specifier in a constituent
 - COMPS – elements that may appear as the complements
 - ...
-
- ARG-ST (argument structure) – ordered list of arguments required by the sign
 - Ordered lists are denoted by angled brackets $\langle \rangle$

Example



- Introduction
- Sign and AVM
- **HPSG Principles**
- Lexicon
- Examples

- We **combine** information from two AVM descriptions.
- Similar to feature unification in GPSG.

- We **combine** information from two AVM descriptions.
- Similar to feature unification in GPSG.

Example

[NUM *sing*]

[PER *3rd*]

- We **combine** information from two AVM descriptions.
- Similar to feature unification in GPSG.

Example

$$\left[\text{NUM} \quad \textit{sing} \right]$$
$$\left[\text{PER} \quad \textit{3rd} \right]$$
$$\left[\begin{array}{l} \text{NUM} \quad \textit{sing} \\ \text{PER} \quad \textit{3rd} \end{array} \right]$$

- We **combine** information from two AVM descriptions.
- Similar to feature unification in GPSG.

Example

$$\left[\text{NUM } \textit{sing} \right]$$
$$\left[\text{PER } \textit{3rd} \right]$$
$$\left[\begin{array}{l} \text{NUM } \textit{sing} \\ \text{PER } \textit{3rd} \end{array} \right]$$

- If features contradict each other, unification **fails**.

Example

$$\left[\text{NUM } \textit{sing} \right]$$
$$\left[\text{NUM } \textit{plur} \right]$$



- Grammar rules and **principles** determine well-formed expressions of a language.
- Formally, principles are implemented by **feature structures**.
 - This means we can also describe them using AVMs.

- Grammar rules and **principles** determine well-formed expressions of a language.
- Formally, principles are implemented by **feature structures**.
 - This means we can also describe them using AVMs.

Some HPSG Principles

- **Head Feature Principle**
- **Valence Principle**
- Immediate Dominance Principle
- Argument Realization Principle
- ...

- Grammar rules and **principles** determine well-formed expressions of a language.
- Formally, principles are implemented by **feature structures**.
 - This means we can also describe them using AVMs.

Some HPSG Principles

- **Head Feature Principle**
 - **Valence Principle**
 - Immediate Dominance Principle
 - Argument Realization Principle
 - ...
-
- Checking of principles is done by **unification** – if unification between the feature structures of the principle and a particular sign fails, then the principle is not satisfied.



Head Feature Principle

The HEAD value of a headed phrase is identified with that of its head-child.

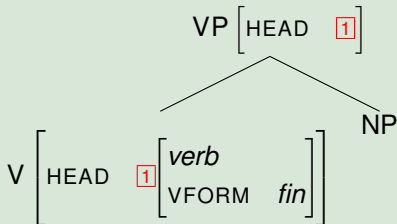
- Ensures that the HEAD properties (part-of-speech, verb inflection, . . .) of head are projected onto headed phrases.

Head Feature Principle

The HEAD value of a headed phrase is identified with that of its head-child.

- Ensures that the HEAD properties (part-of-speech, verb inflection, ...) of head are projected onto headed phrases.

Example





Valence Principle

For each valence feature F , the F value of a headed phrase is the child's F value minus the realized non-head-children.

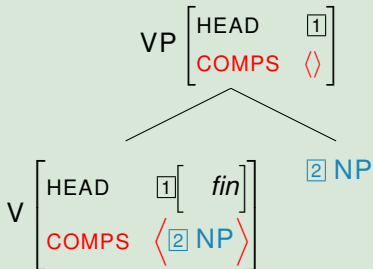
- “Checks off” the combinatorial requirements of lexical head, encoded through valence features (such as SPR, COMP).

Valence Principle

For each valence feature F , the F value of a headed phrase is the child's F value minus the realized non-head-children.

- “Checks off” the combinatorial requirements of lexical head, encoded through valence features (such as SPR , $COMP$).

Example



- Introduction
- Sign and AVM
- HPSG Principles
- **Lexicon**
- Examples

- **Lexical entries** in HPSG are also represented as **feature structures** (using AVM).

Example

gave

HEAD	$\left[\begin{array}{l} \textit{verb} \\ \text{VFORM } \textit{fin} \end{array} \right]$
SUBJ	$\langle \text{NP}[\textit{nom}] \rangle$
COMPS	$\langle \text{NP}[\textit{acc}], \text{PP}[\textit{to}] \rangle$

books

HEAD	\textit{noun}
SPEC	$\langle \text{DetP} \rangle$
COMPS	$\langle \rangle$

- Lexical entries in HPSG are also represented as feature structures (using AVM).

Example

gave

HEAD	$\left[\begin{array}{l} \textit{verb} \\ \text{VFORM } \textit{fin} \end{array} \right]$
SUBJ	$\langle \text{NP}[\textit{nom}] \rangle$
COMPS	$\langle \text{NP}[\textit{acc}], \text{PP}[\textit{to}] \rangle$

books

HEAD	\textit{noun}
SPEC	$\langle \text{DetP} \rangle$
COMPS	$\langle \rangle$

- Lexical entries are **fully inflected** (entries for *give*, *gave*, *given*...).



- If the lexicon were just an unorganized collection of lexical entries, there would be **redundancy**, important generalizations would be missed.



- If the lexicon were just an unorganized collection of lexical entries, there would be **redundancy**, important generalizations would be missed.

Horizontal redundancy

Separate entries for items related according to some recurrent pattern.

- For example plural inflection (*book* and *books*) or active and passive form of verb.



- If the lexicon were just an unorganized collection of lexical entries, there would be **redundancy**, important generalizations would be missed.

Horizontal redundancy

Separate entries for items related according to some recurrent pattern.

- For example plural inflection (*book* and *books*) or active and passive form of verb.

Vertical redundancy

Listing all linguistic information shared by whole classes of words in each entry separately.

- For example, all singular count nouns in English need a determiner.



- Hierarchical classification deals with vertical redundancy.

Hierarchical Classification

- We assign a type (**sort**) to words of specific categories.
- **Supersort** – category which covers a group of words.
- Constraints that are shared by category of words are assigned to supersort.
- Each lexical entry lists its sort – the constraints of the category are then **inherited** from its supersort.
 - We do not need to list these constraints for each entry separately.



- **Lexical rules** deal with **horizontal redundancy**.

Lexical Rules

- Generate new lexical entries from basic entries.
 - Reduces the number of entries we need to store.

- **Lexical rules** deal with **horizontal redundancy**.

Lexical Rules

- Generate new lexical entries from basic entries.
 - Reduces the number of entries we need to store.

Example

Passive lexical rule:

$$\left[\begin{array}{l} \textit{trans-verb} \\ \text{PHON} \quad \boxed{1} \\ \text{SUBJ} \quad \langle \text{NP}_i \rangle \\ \text{COMPS} \quad \langle \boxed{2}, \dots \rangle \end{array} \right] \rightarrow \left[\begin{array}{l} \textit{passive-verb} \\ \text{PHON} \quad F_{\textit{pass}}(\boxed{1}) \\ \text{SUBJ} \quad \langle \boxed{2} \rangle \\ \text{COMPS} \quad \langle \dots, (\text{PP}_i) \rangle \end{array} \right]$$

- Introduction
- Sign and AVM
- HPSG Principles
- Lexicon
- **Examples**

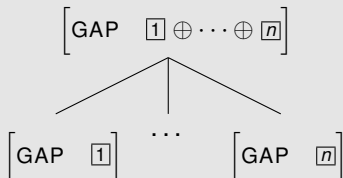


- To deal with long distance dependencies, HPSG uses **GAP feature**.
- ARG-ST – arguments required by the argument structure.
- **Missing arguments** appear in the value of GAP.

- To deal with long distance dependencies, HPSG uses **GAP feature**.
- ARG-ST – arguments required by the argument structure.
- **Missing arguments** appear in the value of GAP.

- The GAP feature percolates up to the parent node.

GAP Principle



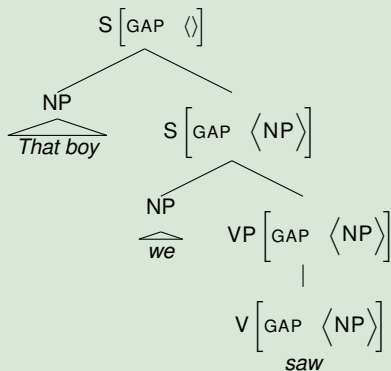
- In a well-formed sentences the GAP feature must be satisfied.
- We can apply the **head filler rule**.

Head Filler Rule

$$\begin{bmatrix} \textit{phrase} \\ \text{GAP} \langle \rangle \end{bmatrix} \rightarrow \boxed{1} \begin{bmatrix} \textit{phrase} \\ \text{GAP} \langle \rangle \end{bmatrix} \quad \text{H} \begin{bmatrix} \textit{phrase} \\ \text{FORM} \textit{fin} \\ \text{SPR} \langle \rangle \\ \text{GAP} \langle \boxed{1} \rangle \end{bmatrix}$$

Example

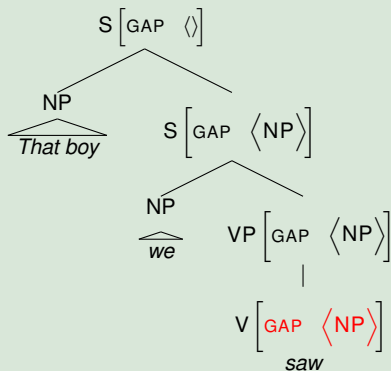
That boy we saw



- 1 The verb see requires a complement which is not present – GAP feature is filled (NP is required by ARG-ST of the verb).
- 2 GAP feature percolates up.
- 3 At the sentence level, we can apply the head filler rule – GAP becomes empty.

Example

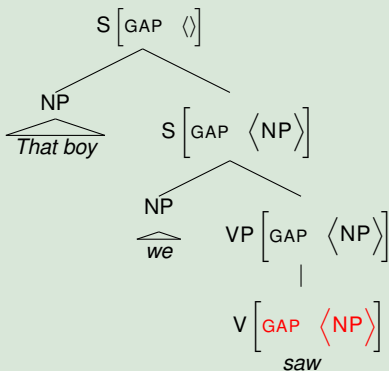
That boy we saw



- 1 The verb *see* requires a complement which is not present – GAP feature is filled (NP is required by ARG-ST of the verb).
- 2 GAP feature percolates up.
- 3 At the sentence level, we can apply the head filler rule – GAP becomes empty.

Example

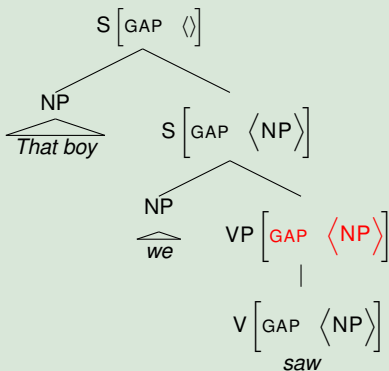
That boy we saw



- 1 The verb see requires a complement which is not present – GAP feature is filled (NP is required by ARG-ST of the verb).
- 2 **GAP feature percolates up.**
- 3 At the sentence level, we can apply the head filler rule – GAP becomes empty.

Example

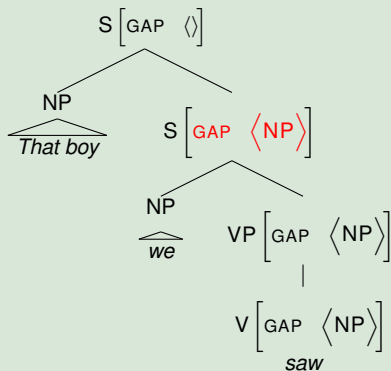
That boy we saw



- 1 The verb see requires a complement which is not present – GAP feature is filled (NP is required by ARG-ST of the verb).
- 2 **GAP feature percolates up.**
- 3 At the sentence level, we can apply the head filler rule – GAP becomes empty.

Example

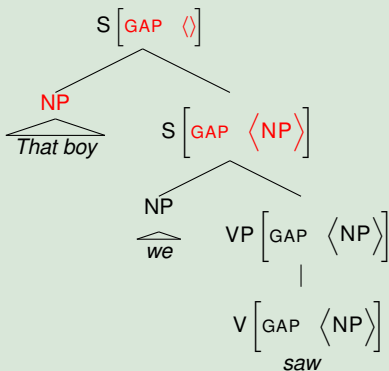
That boy we saw



- 1 The verb see requires a complement which is not present – GAP feature is filled (NP is required by ARG-ST of the verb).
- 2 **GAP feature percolates up.**
- 3 At the sentence level, we can apply the head filler rule – GAP becomes empty.

Example

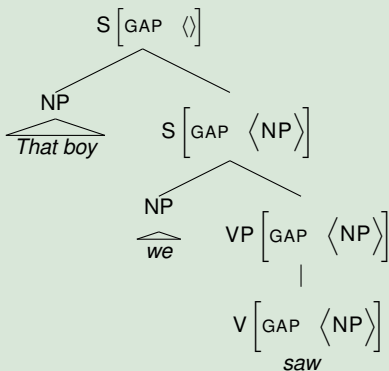
That boy we saw



- 1 The verb see requires a complement which is not present – GAP feature is filled (NP is required by ARG-ST of the verb).
- 2 GAP feature percolates up.
- 3 At the sentence level, we can apply the head filler rule – GAP becomes empty.

Example

That boy we saw



- 1 The verb see requires a complement which is not present – GAP feature is filled (NP is required by ARG-ST of the verb).
- 2 GAP feature percolates up.
- 3 At the sentence level, we can apply the head filler rule – GAP becomes empty.



James Allen:

Natural Language Understanding,
The Benjamin/Cummings Publishing Company. Inc., 2005



Andrew Carnie:

Syntax: A Generative Introduction,
Blackwell Publishing, Oxford, 2002



Carl Pollard, Ivan A. Sag:

Head-Driven Phrase Structure Grammar,
University of Chicago Press, 1994

Thank you for your attention!

End