Head-Driven Phrase Structure Grammar

Stefan Müller
Deutsche Grammatik
Institut für Deutsche und Niederländische Philologie
Fachbereich Philosophie und Geisteswissenschaften
FU Berlin
Stefan.Mueller@fu-berlin.de
September 23, 2009

Head-Driven Phrase Structure Grammar (Background)

• developed in the 80s as a successor of GPSG
• main publications Pollard and Sag, 1987, 1994, many contributions since then
  • syntactic theory
  • language typology
  • computational linguistics, grammar development
    (German, English, French, Norwegian, Japanese, Spanish, Persian, Maltese, Danish, Polish, Mandarin Chinese, …)
• Phonology, morphology, syntax, semantics, and pragmatics (information structure) are covered.
• since 1994 yearly conferences:
  conference volumes are published by CSLI online publications
• Web pages:
  http://hpsg.stanford.edu/ and
  http://hpsg.fu-berlin.de/HPSG-Bib/ (Literature)

Motivations for HPSG

• Increased Precision
• Framework for Integration
• Declarative, Constraint Satisfaction System
• Grammars that Scale Up
• Grammars that Can be Implemented
• Psycholinguistic Plausibility
Important Moments in the History of Linguistics – I

Chomsky (1968) speaking of early psycholinguistic findings in relation to the ‘derivational theory of complexity’ (DTC):

*The results show a remarkable correlation of the amount of memory and number of transformations.* (Chomsky, 1968)

Important Moments in the History of Linguistics – II

Fodor, Bever and Garrett (1974):

Experimental investigations of the psychological reality of linguistic structural descriptions have [...] proved quite successful.

Fodor, Bever and Garrett (1974):

Investigations of DTC [...] have generally proved equivocal. This argues against the occurrence of grammatical derivations in the computations involved in sentence recognition.

HPSG as response to the Fodor, Bever, Garrett dilemma

- HPSG recognizes the ‘linguistic structural descriptions’ whose psychological reality is established, e.g. phonological representations, semantic representations.
- HPSG defines these descriptions via structural definitions and ‘interface constraints’ (Jackendoff), thus eliminating grammatical derivations in FBG’s sense.

General Overview of the Framework

- lexicalized (head-driven), but with some phrasal elements
- sign-based (Saussure, 1916)
- typed feature structures (lexical entries, phrases, principles)
- multiple inheritance
- phonology, syntax, and semantics are represented in one description:

```
Phonology

Syntax

Semantics
```

```
word

loc

_dur

[inst X]

[grammatical]

[case]

[DET [case]]

[cat]

[subcat]

[case]

[head]

[case]

[grammatical]

[head]

[case]

```

```
Valency and Grammar Rules: PSG

- huge number of rules:
  \[ S \rightarrow NP, V \]
  \[ X \text{ schl"aft} ('sleeps') \]
  \[ S \rightarrow NP, NP, V \]
  \[ X Y \text{ liebt} ('loves') \]
  \[ S \rightarrow NP, PP["uber], V \]
  \[ X \text{"uber} y \text{ spricht} ('talks about') \]
  \[ S \rightarrow NP, NP, NP, V \]
  \[ X Y Z \text{ gibt} ('gives') \]
  \[ S \rightarrow NP, NP, PP[mit], V \]
  \[ X Y \text{ mit} Z \text{ dient} ('serves') \]
- verbs have to be used with the right rule

Valency and Grammar Rules: HPSG

- arguments represented as complex categories in the lexical entry of the head (similar to categorial grammar)
- Verb \textbf{SUBCAT}
  \[ schlafen \quad \langle NP \rangle \]
  \[ lieben \quad \langle NP, NP \rangle \]
  \[ sprechen \quad \langle NP, PP["uber] \rangle \]
  \[ geben \quad \langle NP, NP, NP \rangle \]
  \[ dienen \quad \langle NP, NP, PP[mit] \rangle \]

Example Tree with Valency Information (I)

\[ V[\text{SUBCAT} \emptyset] \]
\[ \text{NP} \]
\[ \text{V[SUBCAT} \emptyset]\]
\[ \text{Peter schl"aft} \]

\[ V[\text{SUBCAT} \emptyset] \text{ corresponds to a fully saturated phrase (VP or S)} \]
Valency and Grammar Rules: HPSG

• specific rules for head argument combination:
  \[ V[\text{SUBCAT } A] \rightarrow 1 V[\text{SUBCAT } A \oplus \langle 1 \rangle] \]

• \( \oplus \) is a relation that concatenates two lists:
  \[ \langle a, b \rangle = \langle a \rangle \oplus \langle b \rangle \]

• In the rule above a list is split in a list that contains exactly one element \( \langle 1 \rangle \) and a rest \( \langle A \rangle \).

• Depending on the valency of the head the rest may contain zero or more elements.

Generalization over Rules

• specific rules for head argument combinations:
  \[ V[\text{SUBCAT } A] \rightarrow 1 V[\text{SUBCAT } A \oplus \langle 1 \rangle] \]
  \[ A[\text{SUBCAT } A] \rightarrow 1 A[\text{SUBCAT } A \oplus \langle 1 \rangle] \]
  \[ N[\text{SUBCAT } A] \rightarrow 1 N[\text{SUBCAT } A \oplus \langle 1 \rangle] \]
  \[ P[\text{SUBCAT } A] \rightarrow 1 P[\text{SUBCAT } A \oplus \langle 1 \rangle] \]

• abstraction with respect to the order:
  \[ V[\text{SUBCAT } A] \rightarrow V[\text{SUBCAT } A \oplus \langle 1 \rangle] \]
  \[ A[\text{SUBCAT } A] \rightarrow A[\text{SUBCAT } A \oplus \langle 1 \rangle] \]
  \[ N[\text{SUBCAT } A] \rightarrow N[\text{SUBCAT } A \oplus \langle 1 \rangle] \]
  \[ P[\text{SUBCAT } A] \rightarrow P[\text{SUBCAT } A \oplus \langle 1 \rangle] \]

• generalized, abstract schema (H = head):
  \[ H[\text{SUBCAT } A] \rightarrow H[\text{SUBCAT } A \oplus \langle 1 \rangle] \]

Application of the Rules

• generalized, abstract schema (H = head):
  \[ H[\text{SUBCAT } A] \rightarrow H[\text{SUBCAT } A \oplus \langle 1 \rangle] \]

• possible instantiations of the schema:
  \[ V[\text{SUBCAT } A] \rightarrow V[\text{SUBCAT } A \oplus \langle 1 \textrm{NP} \rangle] \]
  \[ V[\text{SUBCAT } A] \rightarrow V[\text{SUBCAT } A \oplus \langle 1 \textrm{NP} \rangle] \]
  \[ N[\text{SUBCAT } A] \rightarrow N[\text{SUBCAT } A \oplus \langle 1 \textrm{Det} \rangle] \]

Representation of Valency in Feature Descriptions

\[ \text{gibt} \ (\text{’gives’}, \text{finite form}): \]
\[
\begin{array}{c}
\text{PHON} \\
\langle \text{gibt} \rangle \\
\text{PART-OF-SPEECH} \\
\text{verb} \\
\langle \text{NP[nom], NP[acc], NP[dat]} \rangle \\
\end{array}
\]

\[ \text{NP[nom], NP[acc] and NP[dat] are abbreviations of complex feature descriptions.} \]
Demo: Grammar 3

(1) a. der Mann schläft
    the man sleeps
    'The man sleeps'

b. der Mann die Frau kennt
    the man the woman knows
    'The man knows the woman.'
An Example

\[
\begin{array}{c}
\text{V[SUBCAT (\text{\textendash\textendash})]} \\
\text{C} \\
\text{H} \\
\text{NP[nom]} \\
\text{V[SUBCAT (I)]} \\
\end{array}
\]
\[
\begin{array}{c}
\text{NP[acc]} \\
\text{V[SUBCAT (II)]} \\
\text{C} \\
\text{H} \\
\text{NP[dat]} \\
\text{V[SUBCAT (I I)]} \\
\end{array}
\]

\begin{array}{c}
er das Buch dem Mann gibt \\
he the book the man gives \\
\end{array}

Part of the Structure in AVM Representation – PHON values (I)

\[
\begin{array}{c}
\text{PHON} \langle \text{er das Buch dem Mann gibt} \rangle \\
\text{HEAD-DTR} \langle \text{PHON} \langle \text{gibt} \rangle \rangle \\
\text{NON-HEAD-DTR} \langle \text{PHON} \langle \text{Mann} \rangle \rangle \\
\text{NON-HEAD-DTR} \langle \text{PHON} \langle \text{dem} \rangle \rangle \\
\end{array}
\]

Part of the Structure in AVM Representation – PHON values (II)

\[
\begin{array}{c}
\text{PHON} \langle \text{er das Buch dem Mann gibt} \rangle \\
\text{PHON} \langle \text{das Buch dem Mann gibt} \rangle \\
\text{HEAD-DTR} \langle \text{PHON} \langle \text{gibt} \rangle \rangle \\
\text{HEAD-DTR} \langle \text{PHON} \langle \text{Mann} \rangle \rangle \\
\text{NON-HEAD-DTR} \langle \text{PHON} \langle \text{dem} \rangle \rangle \\
\end{array}
\]

Partial Structure in Feature Structure Representation

\[
\begin{array}{c}
\text{PHON} \langle \text{dem Mann gibt} \rangle \\
\text{SUBCAT} \langle \text{NP[nom], NP[acc]} \rangle \\
\text{HEAD-DTR} \langle \text{PHON} \langle \text{gibt} \rangle \rangle \\
\text{P-O-S \text{ noun}} \\
\text{SUBCAT} \langle \text{head-argument-phrase} \rangle \\
\end{array}
\]

\[
\begin{array}{c}
\text{NON-HEAD-DTR} \langle \text{PHON} \langle \text{Mann} \rangle \rangle \\
\text{HEAD-DTR \ldots} \\
\text{NON-HEAD-DTR \ldots} \\
\text{head-argument-phrase} \\
\end{array}
\]
Head-Driven Phrase Structure Grammar
Head Argument Structures
Projection of Head Properties

Partial Structure in Feature Structure Representation

Projection of Head Properties

Feature Structure Representation: the HEAD Value

Different Heads Project Different Features
A Lexical Entry with Head Features

- A lexical entry contains the following:
  - phonological information
  - head information (part of speech, verb form, ...)
  - valency information: a list of descriptions of arguments

The Head Feature Principle

- In a headed structure the head features of the mother are identical to the head features of the head daughter.

headed-phrase → [HEAD [HEAD-DTR [HEAD]]]

- head-argument-phrase is a subtype of headed-phrase
  → All constraints apply to structures of this type as well.
- head-argument-phrase inherits properties of/constraints on headed-phrase.

Demo: Grammar 4

(2) a. der Mann schläft
    the man sleeps
    ‘The man sleeps’

b. der Mann die Frau kennt
    the man the woman knows
    ‘The man knows the woman.’

Semantics

- More recent work (in particular work in relation to computational implementations) uses Minimal Recursion Semantics (Copestake, Flickinger, Pollard and Sag, 2005).
The Representation of Relations with Feature Descriptions

\[ \text{love}(e,x,y) \quad \text{book}(x) \]

ARG0 event
ARG1 index
ARG2 index
love

Representation of the CONT Value

- possible data structure \((\text{CONT} = \text{CONTENT})\):
  \[
  \begin{array}{c}
  \text{PHON} \\
  \text{HEAD} \\
  \text{SUBCAT} \\
  \text{CONT} \\
  \end{array}
  \begin{array}{c}
  \text{list of phoneme strings} \\
  \text{head} \\
  \text{list} \\
  \text{mrs} \\
  \end{array}
  \]

- more structure:
  partition into syntactic and semantic information \((\text{CAT} = \text{CATEGORY})\):
  \[
  \begin{array}{c}
  \text{PHON} \\
  \text{CAT} \\
  \text{SUBCAT} \\
  \text{CONT} \\
  \end{array}
  \begin{array}{c}
  \text{list of phoneme strings} \\
  \text{head} \\
  \text{list} \\
  \text{mrs} \\
  \end{array}
  \]

- \(\rightarrow\) it is now possible to share syntactic information only

Sharing of Syntactic Information in Coordinations

- symmetric coordination: the CAT value is identical
  \[
  \begin{array}{c}
  \text{PHON} \\
  \text{CAT} \\
  \text{CONT} \\
  \end{array}
  \begin{array}{c}
  \text{list of phoneme strings} \\
  \text{list} \\
  \text{mrs} \\
  \end{array}
  \]

- Examples:
  (3) a. [the man and the woman]
  b. He [knows and likes] this record.
  c. He is [stupid and arrogant].

The Semantic Contribution of Nominal Objects

- semantic index + restrictions
  \[
  \begin{array}{c}
  \text{PHON} \\
  \text{CAT} \\
  \text{IND} \\
  \text{CONT} \\
  \text{REL} \\
  \end{array}
  \begin{array}{c}
  \text{(Buch)} \\
  \text{noun} \\
  \text{PER 3} \\
  \text{arg0} \\
  \text{mrs} \\
  \end{array}
  \]

- Person, number, and gender are relevant for reference/coreference:
  (4) Die Frau \(i\) kauft ein Buch \(j\). Sie \(i\) liest es \(j\).
  the woman buys a book she reads it
The Semantic Contribution of Verbs and Linking

• Linking of valency information and semantic contribution

\[ \text{gibt (gives, finite Form)}: \]

• The referential indices of the NPs are identified with the semantic roles.

Semantics Principle (Part)

In headed structures the semantic index of the mother is identical to the semantic index of the head daughter.

The \text{RELS} list of the mother is the concatenation of the \text{RELS} lists of the daughters.

Demo: Berligram

(5) Jeder Sohn eines Beamten rennt.

every son of a state.employee runs
Types: A Non-Linguistic Example for Multiple Inheritance

- electronic device
- printing device
- scanning device
- printer
- copy machine
- scanner
- laser printer
- negative scanner

Properties of Type Hierarchies

- Subtypes inherits properties and constraints of their supertypes.
- Generalizations can be captured: General restrictions are represented at types that are high in the hierarchy.
- More special types inherit from their super types.
- We can represent information with no redundancy.

Linguistic Generalizations in the Type System

- Types are organized in a hierarchy.
- The most general type is on top.
- Information about properties of objects of a certain type are specified as constraints on this type.
- Subtypes inherit these properties.
- Example: Entries in an Encyclopedia. Entry refers to more general concepts, no repetition of information that is present at more general concepts.
- The upper part of the hierarchy is relevant for all languages ("universal grammar").
- More specific type can be relevant for certain classes of languages or even single languages only.

Type Hierarchy for sign

- sign
  - word
  - phrase
    - non-headed-phrase
    - headed-phrase
      - head-argument-phrase

All subtypes of headed-phrase inherit restrictions.
Hierarchical Organization of Knowledge

Types

Type Hierarchy for sign

All Constraints for a Local Tree (Head-Argument)

Partial Structure in Feature Structure Representation

Head-Driven Phrase Structure Grammar

Lexical Regularities

Vertical Generalizations: Type Hierarchies

The Lexicon

- lexicalization → enormous reduction of the number of dominance schemata
- but very complex lexical entries
- structuring and classification → capturing of generalizations & avoidance of redundancies
- type hierarchies and lexical rules

The Complexity of a Lexical Entry of a Count Noun

Only a very small part of this is idiosyncratic.
Partitioning of the Information

- a. all nouns
  - CAT[HEAD noun]
  - CONT nom-obj

- b. all referential non-pronominal Ns taking a determiner (in addition to a)
  - CAT[SUBCAT DET]
  - CONT
    - IND [PER 3]
    - RELS [INST psoa, ...]

- c. all feminine nouns (in addition to a)
  - CONT[IND][GEN fem]

The Complexity of a Lexical Entry for a Verb

- help- (Lexical entry (root)):
  - PHON ⟨ help ⟩
  - CAT
    - HEAD verb
    - SUBCAT ⟨ NP[nom], NP[dat] ⟩
  - CONT
    - IND [3]
    - RELS ⟨ ARG0 [3], ARG1 [3], ARG2 [3], helf en ⟩

Part of a Possible Type Hierarchy

- appropriate paths have to be added:
  - [SUBCAT ()] is a shorthand for [CAT[SUBCAT ()]]
- Constraints on types hold for their subtypes as well (inheritance).
- Instances are connected via dashed lines.
Examples for Lexical Items

- **PHON** ⟨Frau⟩
- **CONT|RELS** ⟨frau⟩
- **count-noun-root**

- **PHON** ⟨helf⟩
- **CONT|RELS** ⟨helfen⟩
- **np-np-dat-verb-root**

Horizontal and Vertical Generalizations

- Type hierarchies are used to cross-classify linguistic objects (lexical entries, schemata).
- We express generalizations over classes of linguistic objects.
- We are able to say what certain words have in common:
  - woman and man
  - woman und salt
  - woman und plan
- But there are other regularities:
  - kick and kicked as in was kicked
  - love und loved as in was loved
- We can use a hierarchy to represent the properties of kicked and loved, but this would not capture the fact that kick and kicked are related in the same way as love and loved.
- Remark: There are proposals in the literature to treat passive by inheritance, but this does not work in general (Müller, 2006, 2007).

Lexical Rules

- Instead of inheritance we use lexical rules.
  - Jackendoff (1975), Williams (1981), Bresnan (1982),
    - Shieber, Uszkoreit, Pereira, Robinson and Tyson (1983),
    - Flickinger, Pollard and Wasow (1985), Flickinger (1987),
    - Copestake and Briscoe (1992), Meurers (2000)
- Example passive:
  - A lexical rule relates a stem to the corresponding passive form.
- There are different conceptions of lexical rules:
  - Meta Level Lexical Rules (MLR) vs.
  - Description Level Lexical Rules (DLR)
  - See Meurers, 2000 for a detailed discussion.

Lexical Rule for the Passive

The man beats the dog.

The dog was beaten.

Note: This is simplified, see Müller, 2002 for Haider’s passive analysis in HPSG.
Conventions for the Interpretation of Lexical Rules

- Information that is not mentioned in the output, is carried over from the input.
- Example: Passive preserves meaning.
  The CONT values of input and output are identical. Linking information is preserved:

  Active:
  \[
  \begin{array}{c}
  \text{CAT} \quad \text{SUBCAT} \quad \text{IND} \\
  \text{CONT} \quad \text{RELS} \quad \text{ARG0} \quad \text{ARG1} \quad \text{ARG2} \quad \text{beat}
  \end{array}
  \]

  Passive:
  \[
  \begin{array}{c}
  \text{CAT} \quad \text{SUBCAT} \quad \text{IND} \\
  \text{CONT} \quad \text{RELS} \quad \text{ARG0} \quad \text{ARG1} \quad \text{ARG2} \quad \text{beat}
  \end{array}
  \]

The Lexical Rule for the Passive in a Different Notation

\[
\begin{array}{c}
\text{CAT} \quad \text{SUBCAT} \quad \text{IND} \quad \text{NP[nom]} \quad \text{NP[acc]} \\
\text{CONT} \quad \text{RELS} \quad \text{verb}
\end{array}
\]

\[
\begin{array}{c}
\text{ACC-PASSIVE-LEXICAL-RULE}
\end{array}
\]

- like a unary projection, but restricted to the lexicon
- word \( \triangleright \) acc-passive-lexical-rule
- Since lexical rules are typed, we can capture generalizations over lexical rules.
- This form of lexical rule is fully integrated into the HPSG formalism.

The Lexical Rule for the Passive with Morphology

\[
\begin{array}{c}
\text{PHON} \quad f(\text{[H]}) \\
\text{CAT} \quad \text{SUBCAT} \quad \text{IND} \quad \text{NP[nom]} \quad \text{NP[acc]} \\
\text{LEX-DTR} \quad \text{CAT[SUBCAT} \quad \text{NP[nom]} \quad \text{NP[acc]} \quad \text{stem}
\end{array}
\]

\[
\begin{array}{c}
\text{ACC-PASSIVE-LEXICAL-RULE}
\end{array}
\]

- \( f \) is a function that returns the passive form that corresponds to the PHON value of the LEX-DTR (kick \( \rightarrow \) kicked)
- Alternative: Head Affix Structures
  (similar to binary branching structures in syntax)

Head Affix Structures vs. Lexical Rules

- Lexical Rules
  (Örgun, 1996; Riehemann, 1998; Ackerman and Weibelhuth, 1998; Koenig, 1999; Müller, 2002)
- Head Affix approaches
  (Krieger and Nerbonne, 1993; Krieger, 1994; van Eynde, 1994; Lebeth, 1994)

- The approaches can be translated into each other in many cases (Müller, 2002).
- Sometimes it is regarded as an advantage that lexical rules make the stipulation of hundreds of empty affixes for zero inflection and conversion unnecessary.
- Subtractive morphemes are not needed in an LR-based approach.
- Some languages have affixal material that realizes more than one argument (Crysmann, 2002, Chapter 2.1.1.4 and p. 169–171).
Languages with Free(er) Constituent Order

- We will now look at German, since it is interesting in its reordering possibilities.
- German is an SOV language, however in declarative clauses the verb appears in second position and in matrix interrogative clauses, it appears in first position.
- How do we account for the serialization of arguments?
- How do we account for the verb position?

Relatively Free Constituent Order

- Arguments can be serialized in almost any order:
  
  (7) a. weil der Mann der Frau das Buch gibt
  because the man the woman the book gives
  b. weil der Mann das Buch der Frau gibt
  c. weil das Buch der Mann der Frau gibt
  d. weil das Buch der Frau der Mann gibt
  e. weil der Frau der Mann das Buch gibt
  f. weil der Frau das Buch der Mann gibt
  
  (7b–f) require a different prosody and a more restrictive context than (7a) (Höhle, 1982).

Adjuncts in the Mittelfeld

- In addition to the arguments, adjuncts may be serialized in the Mittelfeld.
- These can be placed at arbitrary positions between the arguments:
  
  (8) a. weil morgen der Mann das Buch der Frau gibt
  because tomorrow the man the woman the book gives
  'because the man gives the book to the woman'
  b. weil der Mann morgen das Buch der Frau gibt
  c. weil der Mann das Buch morgen der Frau gibt
  d. weil der Mann das Buch der Frau morgen gibt

Binary Branching Structures

- Sentences like (9) are unproblematic:
  
  (9) weil [der Mann [das Buch [der Frau gibt]]]
  because the man the book the woman gives

- The integration of adjuncts is straightforward as well:
  
  (10) a. weil [morgen [der Mann [das Buch [der Frau gibt]]]]
  b. weil [der Mann [morgen [das Buch [der Frau gibt]]]]
  c. weil [der Mann [das Buch [morgen [der Frau gibt]]]]
  d. weil [der Mann [das Buch [der Frau [morgen gibt]]]]
Permutation of Arguments in the Mittelfeld

- Permutation of arguments is not explained yet.
- Thus far, we have combined the head with the last element in the subcat list.

\[
\text{head-argument-phrase} \to
\begin{bmatrix}
\text{CAT|SUBCAT } A \\
\text{HEAD-DTR|CAT|SUBCAT } A \oplus \langle 1 \rangle \\
\text{NON-HEAD-DTRS } \langle 1 \rangle
\end{bmatrix}
\]

- Generalization of the Head-Argument-Schema:
  We allow to take arguments from the middle of the list.

\[
\begin{bmatrix}
\text{CAT|SUBCAT } A \oplus B \\
\text{HEAD-DTR|CAT|SUBCAT } A \oplus \langle 1 \rangle \oplus B \\
\text{NON-HEAD-DTRS } \langle 1 \rangle
\end{bmatrix}
\]

The Head-Argument-Schema

- old:

\[
\text{head-argument-phrase} \to
\begin{bmatrix}
\text{CAT|SUBCAT } A \\
\text{HEAD-DTR|CAT|SUBCAT } A \oplus \langle 1 \rangle \\
\text{NON-HEAD-DTRS } \langle 1 \rangle
\end{bmatrix}
\]

- new:

\[
\begin{bmatrix}
\text{CAT|SUBCAT } A \oplus B \\
\text{HEAD-DTR|CAT|SUBCAT } A \oplus \langle 1 \rangle \oplus B \\
\text{NON-HEAD-DTRS } \langle 1 \rangle
\end{bmatrix}
\]

- Note: If we want binary branching for English: \( A = \langle \rangle \)

Example: Normal Order

(11) a. weil jeder das Buch kennt
   because everybody the book knows
   \[ V[\text{SUBCAT } \langle \rangle] \]
   \[ \begin{bmatrix} \text{NP[nom]} & V[\text{SUBCAT } \langle 1 \rangle] \end{bmatrix} \]
   \[ \begin{bmatrix} \text{NP[acc]} & V[\text{SUBCAT } \langle 1 \rangle] \end{bmatrix} \]
   jeder das Buch kennt

Example: Reordering

\[ V[\text{SUBCAT } \langle \rangle] \]
\[ \begin{bmatrix} \text{NP[acc]} & V[\text{SUBCAT } \langle 1 \rangle] \end{bmatrix} \]
\[ \begin{bmatrix} \text{NP[nom]} & V[\text{SUBCAT } \langle 1 \rangle] \end{bmatrix} \]
\[ \text{das Buch} \quad \text{jeder} \quad \text{kennt} \]

The difference is the order in which the elements in SUBCAT get saturated.

See Gunji, 1986 for similar suggestions for Japanese.
See Fanselow, 2001 for an equivalent suggestion in the Minimalist Program.
Constituent Order
Permutation of Constituents in the Mittelfeld
Arguments
Demo: Grammar 9
(12) a. daß der Mann der Frau das Buch gibt
   that the man nom the woman dat the book acc gives
b. daß der Mann das Buch der Frau gibt
   that the man nom the book acc the woman dat gives
c. daß der Mann der Frau das Buch morgen gibt
   that the man nom the woman dat the book acc tomorrow gives
d. daß der Mann der Frau morgen das Buch gibt
   that the man nom the woman dat tomorrow the book acc gives
e. daß er oft nicht lacht
   that he often not laughs
f. daß er nicht oft lacht
   that he not often laughs

Head-Driven Phrase Structure Grammar
Constituent Order
Verb Placement
Demo: Grammar 9
(13) Gibt der Mann der Frau das Buch.
gives the man nom the woman dat the book acc

Vorfeldbesetzung in German is a Nonlocal Dependency
• One constituent (adjunct, subject or complement) can be placed in the Vorfeld (Erdmann, 1886; Paul, 1919) → V2 language
• Fronting is a non-local dependency:
(14) a. [Um zwei Millionen Mark]; soll er versucht haben,
   [eine Versicherung zu betrügen].
1
b. „Wer, glaubt er, daß er ist?“ erregte sich ein Politiker vom Nil.
2
c. Wen; glaubst du, daß ich gesehen habe.
3
  'Who do you believe that I saw?'

1taz, 04.05.2001, p. 20.
2Spiegel, 8/1999, p. 18.
3Scherpenisse, 1986, p. 84.
Overview Fronting

• As in the example of head movement: Trace at "canonical" position
• percolation of information in the tree
• constituent movement is not local, verb movement is two different features for modelling (SLASH vs. DSL)

Properties of the Analysis
• percolation of non-local information
• structure sharing
• Information is simultaneously present at every node in the extraction path.
• Nodes in the middle of an unbounded dependency may access this information. (Bouma et al., 2001: Irish, Chamorro, Palauan, Icelandic, Kikuyu, Ewe, Thompson Salish, Moore, French, Spanish, and Yiddish)

Data Structure: Differentiation into Local and Nonlocal Information

• Differenciation between information that is locally relevant (LOCAL) and information that plays a role in nonlocal dependencies (NONLOCAL)

Nonlocal Dependencies

Data Structure for Nonlocal Information

• NONLOC value has internal structure:
  
  \[
  \begin{array}{l}
  \text{QUE } \text{list of npros} \\
  \text{REL } \text{list of indices} \\
  \text{SLASH } \text{list of local structures} \\
  \end{array}
  \]

  \[
  \begin{array}{l}
  \text{nonloc} \\
  \end{array}
  \]

• QUE: list of indices of question words (interrogative clauses)
• REL: list of indices of relative pronouns (relative clauses)
• SLASH: list of local objects (Vorfeldbesetzung, relative clauses)
• We focus on SLASH and ignore the others.
A Trace for the Accusative Object of *kennen* ('know')

- The trace does not contribute a phonology.
- The trace has the local properties that *kennen* requires.
- These are also represented under slash.

The Head Filler Schema (Simplified)

- The head daughter is a finite clause with a verb in final position (*initial+*) and an element in SLASH.
- LOCAL value of the non-head daughter is identical to the element in SLASH.
- nothing can be extracted from the non-head daughter.

The Extraction Trace

- The trace used as argument was specific, but we can generalize over all traces.
- We do not have to specify the LOCAL value of the trace since the verb specifies the LOCAL value of its arguments.
Why Prefer the Constraint-Based Analysis to the Movement-Based Alternative?

- Detailed, empirically adequate accounts
- Precisely formalized and theoretically motivated
- Psycholinguistic concerns
- Cross-linguistic sensitivity to extraction paths
- Lexical exceptions: obligatory gap selection
- Partial identity between filler and gap
- Coordinate Structure Constraint and Across-the-Board Exceptions

Cross-Linguistic Data: Extraction Path Sensitivity

- Irish complementizer selection (McCloskey 1978, 1989)
- French 'stylistic' inversion (Kayne and Pollock 1978)
- Spanish 'stylistic' inversion (Torrego 1984)
- Kikuyu downstep suppression (Clements 1984, Zaenen 1983)
- Chamorro verb agreement (Chung 1982, 1995)
- Yiddish inversion (Diesing 1990)
- Icelandic expletives (Zaenen 1983)
- Adyghe 'wh-agreement' (Polinsky 2007)

Irish

(15) a. Shíl mé goN mbeadh sé ann.
    'I thought he would be there.'

b. Duirt mé gurL shíl mé goN mbeadh sé ann.
    'I said that he would be there.'

c. an fear aL shíl mé aL bheadh ann
   the man COMP thought I COMP would be there
   'the man that I thought would be there'

d. an fear aL duirt mé aL shíl mé aL bheadh ann
   the man COMP said I COMP would be there
   'the man that I said I thought would be there'

e. an fear aL shíl goN mbeadh sé ann
   [the man], COMP thought I COMP would be he, there
   '[the man], that thought he would be there'

f. an fear aL duirt sé aL shíl goN mbeadh sé ann
   the man COMP said he COMP thought I COMP would be he there
   'the man that he said thought he would be there'
Filler-Gap Mismatches

(16) a. You can rely on Dominique’s help.
   b. Dominique’s help, you can rely on __.
   c. * You can rely on that they will help you.
   d. That they will help you, you can rely on __.

How can movement turn an NP into a CP?
Certain Filler-Gap constructions involve constraints of partial identity, not total identity.

Coordinate Structure Constraint and ATB Exceptions – I

(17) a. * Which dignitaries do you think
    [[Sandy photographed the castle] and [Chris visited __]]?
   b. * Which dignitaries do you think
    [[Sandy photographed __] and [Chris visited the castle]]?
   c. Which dignitaries do you think
    [[Sandy photographed __] and [Chris visited __]]?

The Movement-Based Analysis

(18) a. [S you think [S’ [S Kim should help who; i]]]
   b. [S you think [S’ who; [S Kim should help __; i]]?]
   c. who; [S you think [S’ e; [S Kim should help __; i]]?]
   d. who; [S do you think [S’ e; [S Kim should help __; i]]?]

Across-the-Board Movement?

(19) who; [S do you think
    [S’ e; [[S Kim likes ___; i] and [S Pat hates ___; i]]]]?

There is no uniform movement algorithm that allows across-the-board movement. (Gazdar, Pullum, Sag and Wasow, 1982)
Across-the-Board Constraint

SLASH is among the features whose values are identified across conjuncts:

```
S[SLASH \[1\] \(NP\)]
```

- NP
- VP[SLASH \[1\]]
- you
- V
- think
- S[SLASH \[1\]]
- C
- S[SLASH \[1\]]
- Kim likes
- and
- Pat hates

HPSG – The Frankenstein Theory

- Bob Carpenter (Mineur, 1995): HPSG is a Frankenstein Theory. Sewed together from various other theories. Influences:
  - GPSG (no surprise, authors overlap): nonlocal dependencies, ID/LP format
  - Categorial Grammar (valence, functor/argument relationships)
  - GB (parts of X Theory, parts of structural aspects)

HPSG vs. Transformational Grammar

- A lot of insights are taken over from GB analyses of the 80ies.
  - Categories are feature sets
  - \(\lambda\) Theory (not all aspects)
  - verb position in German (Grewendorf, 1988)
- Differences:
  - No transformations
  - No explicit structuring of the category feature sets
  - Lexical integrity (Bresnan and Mchombo, 1995)
    - no IP node for German, no affixes in syntax → no movement of affixes
  - lexical analysis of passive (as in LFG, see Bresnan, 1982)
  - Scrambling is not movement (Gunji, 1986, see also Fanselow, 1993, 2001)
  - extraction is percolation of information
Evidence from a single language and UG

- What does it mean for other languages that a rule/morpheme is present in one particular language?
- Possible answer:
  If we have a certain structure in language X, it must be present in all languages.
- Example:
  - Basque: Tree positions for object agreement (AgrO, AgrIO)
  - Japanese: Tree position for topic marker
  - German and Dutch neither have object agreement nor topic morphemes.
- Conclusion:
  If such inferences regarding properties of particular languages, one has to assume (very specific!) innate linguistic knowledge.

German is English/Romance (SVO, Laenzlinger following Kayne)

- All languages are SVO underlyingly.
- The object is moved out of the VP.
- The subject is fronted.
- The empty VP is fronted.
- There are further empty heads (Cinque, 1999).
- Innateness has to be assumed.

English, German, . . . are Hungarian

- Hornstein, Nunes and Grohmann (2005, p. 124): agreement head for the checking of case features
- Preposition is moved there.
- DP is put into the specifier position of this head.
- Evidence for this: Agreement in Hungarian postpositional phrases
- English is like Hungarian, but the movement is invisible.
German is German, . . . Hungarian is Hungarian

- A PP is a P together with an NP (or DP).
- No movement instead of two movements.
- Structure has five nodes less.
- Truly minimal!
- Question: What constitutes an explanation? Where and how is complexity of language represented?

Sociological Differences

- The way arguments work differs dramatically.
- Avoid empty elements!
  This should be a strategy for every linguistic theory (Occam’s Razor!)

HPSG vs. OT

- OT is not a (complete) linguistic theory.
- You need a linguistic theory and on top of it you can do OT.
- Often OT papers do not give their assumptions about the underlying grammar.
- Factorial typology is attractive, but requires the assumption of domain specific innate knowledge about language.
- OT is often misunderstood to provide a way to deal with gradedness. Gradedness can be and has been introduced into HPSG implementations (as in OT-LFG).

HPSG vs. LFG

- HPSG uses typed feature structures, LFG does not.
- Generalizations can be expressed in type hierarchies. LFG uses macro hierarchies for this (Dalrymple, Kaplan and King, 2004).
- There are subtle differences between types and macros (for instance type inference).
Two Levels of Description: c-structure and f-structure

- LFG has two levels of description: c-structure and f-structure. They have different status.
  - c-structure rules are phrase structure rules with atomic node labels. Possibly extended by features.
  - In HPSG phrases are of the same kind as lexical items and lexical rules. This makes it possible to capture generalizations regarding such objects. Examples:
    - Complementizer (word) and initial verb (LR).
    - Adjective (word) and relative clause (phrase)
  - Crosslinguistically oriented work in LFG usually focuses on f-structures. c-structures are often not made explicit.
  - In implementations they vary widely.

LFG’s f-structure and HPSG’s projected Argument Structure

- Wambaya: The traditional NP can be realized discontinuously. Adjectives and nouns agree in case.
- Nordlinger (1998) suggested LFG analysis in which constituents refer to the f-structure for the enforcement of agreement.
- Bender (2008): This can be modeled in HPSG if a non-cancellation approach to valence is assumed.
- Non-cancellation was first suggested in GB: Higginbotham (1985, p. 560).
- Introduced and motivated for HPSG: Meurers, 1999, Przepiórkowski, 1999, Müller, 2008, Chapter 17.4

The Biggest Problem with LFG (not so serious slide)

- The biggest problem with LFG is:
  - They sold the theory to Microsoft. (http://powerset.com/)
  - They sold the theory to the German Telekom: http://www.wipo.int/pctdb/en/wo.jsp?WO=2004003888&IA=WO2004003888&DISPLAY=DESC
  - So HPSG is the only free linguistic theory.

HPSG vs. CG

- Many insights are taken over from CG.
  - Binary branching functor argument structures.
  - (some HPSG grammars use flat structures)
  - Argument composition for predicate complexes (Geach, 1970).
  - CG has problems with relative clauses and pied piping (Pollard, 1988).
  - Additional rules are needed in CG, some are already there (topicalization).
  - It does not make sense for all structures to assume a head (functor).
Comparison
HPSG vs. CxG
Constructionist Aspects

HPSG vs. CxG: Constructionist Aspects

HPSG and CxG are close friends:

- Many of the insights of CxG regarding idiosyncrasies and similar points are taken over into other frameworks.
  - HPSG: Sag, 1997
  - LFG: Asudeh, Dalrymple and Toivonen, 2008
  - Simpler Syntax: Culicover and Jackendoff, 2005

- So all of these frameworks can be regarded as constructional approaches to language (Goldberg).

Formalization and Argument Structure Constructions

However:

- Most work in CxG lacks precision.
- Attempts to formalize the analyses (Kay, 2002) failed (Müller, 2006).
- Formal variants are either directly encoded in Lisp (Fluid CxG) or use the formal apparatus of HPSG (Bergen and Chang, 2005; Sag, 2007).

- Phrasal analyses of argument structure constructions have problems:
  - Müller, 2006, In Preparationa
- Language acquisition facts are not a proof that we need phrasal analyses:
  - Müller, In Preparationa

Empty Elements

- CxG dogma: No empty elements
  - Michaelis and Ruppenhofer, 2001, p. 49–50; Goldberg, 2003, p. 219;
  - Goldberg, 2006, p. 10; Culicover and Jackendoff, 2005

- Approaches with empty elements can be translated into ones without empty elements.
- The ban of empty elements makes capturing of generalizations impossible:
  - Müller, To appeara, Bender, 2002, Sag, Wasow and Bender, 2003, p. 464

- Sociological remark:
  - Most Construction Grammar papers are not about grammar.

Conclusion

- All we have is:
  - features and values
  - complex and atomic values
  - atomic values are types
  - complex values are a certain bundling of features (which makes predictions)
  - identity of values
  - Linguistic objects are of a certain type.

- Therefore they have to satisfy constraints associated with the respective type in order to be licensed by the grammar.

- HPSG can model constructionist aspects of language (CxG insights).
- HPSG is model theoretic/constraint based.
- Since HPSG is precisely formulated, computer implementations are possible, but don’t worry . . .
  - Even if you hate computers, you can work in this framework.
  - It is the right way to go!
Constituent Order: Binary vs. Flat Structures

- We used binary branching structures in Class 1.
  \[ \text{head-argument-phrase} \rightarrow \]
  \[
  \begin{cases}
  \text{CAT|SUBCAT A} \\
  \text{HEAD-DTR|CAT|SUBCAT A} \oplus (B) \\
  \text{NON-HEAD-DTR} (B)
  \end{cases}
  \]

  We will argue for binary branching structures for German shortly.

- However, binary branching is not the only option.
  For languages like English a flat VP is assumed.

- The subject is represented separately (as the value of the feature SPECIFIER).
  The other arguments are represented under COMPS.

- Elements in COMPS are combined with their head in one go.

The English Clause

\[ V[\text{spr}() , \text{comps}()] \]

The following head argument schema licenses VPs, that is, projections of a head that include the head and all its arguments except the specifier.

\[ \text{head-complement-phrase} \rightarrow \]

\[
\begin{cases}
\text{CAT|COMPS ()} \\
\text{HEAD-DTR|CAT|COMPS A} \\
\text{NON-HEAD-DTR} (A)
\end{cases}
\]

Argument-Structure/Valency Mappings: English

- A list valued feature ARGUMENT-STRUCTURE is used for the representation of arguments independent of their function as subject or complement.

- English: The subject is VP-external, both for finite and nonfinite verbs.

- All arguments but the subject are mapped from ARG-ST to COMPS:
  \[ \text{gives:} \]
  \[
  \begin{cases}
  \text{spr}() \\
  \text{comps} (A) \\
  \text{arg-st} \quad \text{NP[nom]} \oplus \text{NP[acc]}, \text{NP[acc]}
  \end{cases}
  \]
  Linking is done with reference to ARG-ST.
Argument-Structure/Valency Mappings: German

- German: no distinction between subject and other arguments for finite verbs.
  (In GB terms: The subject is VP-internal.

- All arguments are mapped from ARG-ST to COMPS:
  gibt (gives, finite Form):
  \[
  \text{SPR} \quad \langle \rangle \\
  \text{COMPS} \quad \text{A} \quad \langle NP[\text{nom}], NP[\text{acc}], NP[\text{dat}] \rangle \\
  \text{ARG-ST} \quad \text{A} \quad \langle NP[\text{nom}], NP[\text{acc}], NP[\text{dat}] \rangle
  \]

References
