Phrases and Sentences

1. Language models
2. Chunking
3. Structural descriptions
4. Parsing with phrase structure grammars
5. Probabilistic parsers
6. Parsing with dependency grammars
7. Principles and Parameters
8. Unification-based grammars
9. Semantics construction

Phrases and Sentences

Unification-based Grammars

- Feature structures
- Rules with complex categories
- Subcategorization
- Movement
- Constraint-based models

Unification-based grammars

- feature structures
- rules with complex categories
- subcategorization
- movement

Feature structures

- feature structures describe linguistic objects (lexical items or phrases) as sets of attribute value pairs
- complex categories: name of the category may be part of the feature structure

Haus: $\begin{bmatrix} \text{cat} & \text{N} \\ \text{case} & \text{nom} \\ \text{num} & \text{sg} \\ \text{gen} & \text{neutr} \end{bmatrix}$ *house:* $\begin{bmatrix} \text{cat} & \text{N} \\ \text{num} & \text{sg} \end{bmatrix}$

- a feature structure is a functional mapping from a finite set of attributes to the set of possible values
  - unique names for attributes / unique value assignment
  - number of attributes is finite but arbitrary
  - feature structure can be extended by additional features

Feature structures

- partial descriptions: underspecified feature structures

Frauen: $\begin{bmatrix} \text{cat} & \text{N} \\ \text{num} & \text{pl} \\ \text{gen} & \text{fem} \end{bmatrix}$

women: $\begin{bmatrix} \text{cat} & \text{N} \\ \text{num} & \text{pl} \end{bmatrix}$

fish: $\begin{bmatrix} \text{cat} & \text{N} \end{bmatrix}$

Feature structures

- subsumtion:

  A feature structure M_1 subsumes a feature structure M_2 iff every attribute-value pair from M_1 is also contained in M_2.

  $\rightarrow$ not all pairs from M_2 need also be in M_1

- constraint-based notation (Shieber 1986): $M_1 \sqsubseteq M_2$
  - M_2 contains a superset of the constraints contained in M_1
  - M_2 is an extension of M_1 (Pollard und Sag 1987)
  - M_1 is less informative than M_2 (Shieber 1986, Pollard und Sag 1987)

  but:

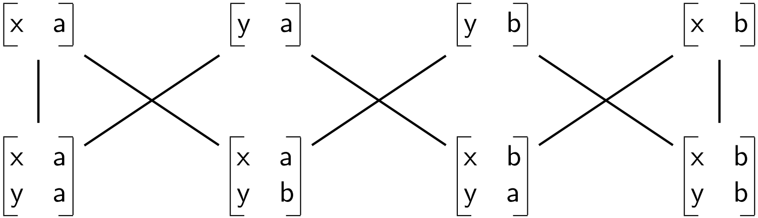
  - M_1 is more general than M_2

- alternative notation:

  instance-based (Pollard und Sag 1987): $M_1 \succeq M_2$

Feature structures

- subsumtion hierarchy



Feature structures

- formal properties of subsumtion
  - reflexive: $\forall \mathsf{M}_i . \mathsf{M}_i \sqsubseteq \mathsf{M}_i$
  - transitive: $\forall \mathsf{M}_i \forall \mathsf{M}_j \forall \mathsf{M}_k . \mathsf{M}_i \sqsubseteq \mathsf{M}_j \wedge \mathsf{M}_j \sqsubseteq \mathsf{M}_k \rightarrow \mathsf{M}_i \sqsubseteq \mathsf{M}_k$
  - antisymmetrical: $\forall \mathsf{M}_i \forall \mathsf{M}_j . \mathsf{M}_i \sqsubseteq \mathsf{M}_j \wedge \mathsf{M}_j \sqsubseteq \mathsf{M}_i \rightarrow \mathsf{M}_i = \mathsf{M}_j$
- subsumtion relation defines a partial order
- not all feature structures need to be in a subsumtion relation

Feature structures

- unification I (subsumtion-based)

  If M_1, M_2 and M_3 are feature structures, then M_3 is the unification of M_1 and M_2

  $$M_3 = M_1 \sqcup M_2$$

  iff

  - M_3 is subsumed by M_1 and M_2 and
  - M_3 subsumes all other feature structures, that are also subsumed by M_1 and M_2.

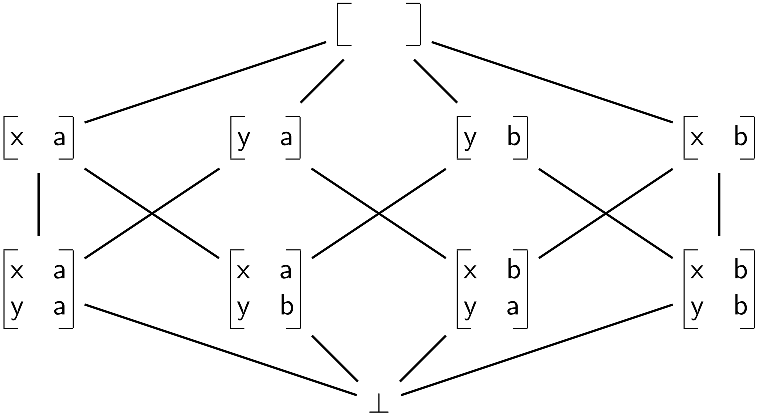
- result of a unification (M_3) is the most general feature structure which is subsumed by M_1 and M_2

Feature structures

- not all feature structures are in a subsumtion relation
  $\rightarrow$ unification may fail
- completing the subsumtion hierarchy to a lattice
  - bottom ($\bot$): inconsistent (overspecified) feature structure
  - top ($\top$): totally underspecified feature structure
    corresponds to an unnamed variable ([ ])

Feature structures

- subsumtion lattice



Feature structures

- unification II (based on the propositional content) (Pollard und Sag 1987)

  The unification of two feature structures M_1 und M_2 is the conjunction of all propositions from the feature structures M_1 and M_2.

- unification combines two aspects:
  1. test of compatibility
  2. accumulation of information

- result of a unification combines two aspects
  1. Boolean value whether the unification was successful
  2. union of the compatible information from both feature structures

Feature structures

- formal properties of the unification
  - idempotent: $\mathsf{M} \sqcup \mathsf{M} = \mathsf{M}$
  - commutative: $\mathsf{M}_i \sqcup \mathsf{M}_j = \mathsf{M}_j \sqcup \mathsf{M}_i$
  - associative: $(\mathsf{M}_i \sqcup \mathsf{M}_j) \sqcup \mathsf{M}_k = \mathsf{M}_i \sqcup (\mathsf{M}_j \sqcup \mathsf{M}_k)$
  - neutral element: $\top \sqcup \mathsf{M} = \mathsf{M}$
  - zero element: $\bot \sqcup \mathsf{M} = \bot$

- unification and subsumtion can be mutally defined from each other

  $\mathsf{M}_i \sqsubseteq \mathsf{M}_j \leftrightarrow \mathsf{M}_i \sqcup \mathsf{M}_j = \mathsf{M}_j$

Feature structures

- recursive feature structures: conditions are not to be defined for individual features but complete feature collections (data abstraction)
- value of an attribute is again a feature structure

she: $\begin{bmatrix} \text{cat} & \text{Pro} \\ \text{bar} & 0 \\ \text{agr} & \begin{bmatrix} \text{pers} & \text{3rd} \\ \text{num} & \text{sg} \\ \text{gen} & \text{fem} \\ \text{case} & \text{nom} \end{bmatrix} \end{bmatrix}$ *us*: $\begin{bmatrix} \text{cat} & \text{Pro} \\ \text{bar} & 0 \\ \text{agr} & \begin{bmatrix} \text{pers} & \text{1st} \\ \text{num} & \text{pl} \\ \text{case} & \text{acc} \end{bmatrix} \end{bmatrix}$

Feature structures

- access to the values through paths

$$\langle \text{ cat } \rangle = \text{Pro}$$
$$\langle \text{ bar } \rangle = 0$$
$$\langle \text{ agr num } \rangle = \text{sg}$$
$$\langle \text{ agr gen } \rangle = \text{fem}$$
$$\langle \text{ agr } \rangle = \begin{bmatrix} \text{num} & \text{sg} \\ \text{gen} & \text{fem} \end{bmatrix}$$

Feature structures

- unification III (constructive algorithm)

  Two feature structures M_1 and M_2 unify, iff for every common feature of both structures

  - in case of atomic values both value assignments are identical or
  - in case of complex values both values unify.

  If successful unification produces as a result the set of all complete paths from M_1 *and* M_2 with their corresponding values. If unification fails the result will be $\perp$.

Feature structures

- recursive data structures can be used
  - lists
  - trees

$$(\text{A B C}) \quad \Longrightarrow \quad \begin{bmatrix} \text{first} & \text{A} \\ \text{rest} & \begin{bmatrix} \text{first} & \text{B} \\ \text{rest} & \begin{bmatrix} \text{first} & \text{C} \\ \text{rest} & \text{nil} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Feature structures

- example: subcategorisation list

$$
(\text{NP[dat] NP[akk]}) \quad \Longrightarrow \quad \begin{bmatrix} \text{first} & \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 2 \\ \text{case} & \text{dat} \end{bmatrix} \\ \text{rest} & \begin{bmatrix} \text{first} & \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 2 \\ \text{case} & \text{acc} \end{bmatrix} \\ \text{rest} & \text{nil} \end{bmatrix} \end{bmatrix}
$$

- two lists unify iff
  - they have the same length and
  - their elements unify pairwise.

Feature structures

- information in a feature structure is conjunctively combined
- feature structures may also contain disjunctions

$$
\begin{bmatrix} \text{agr} & \begin{bmatrix} \text{pers} & \text{2nd} \\ \text{num} & \left\{ \textit{sg} \;\; \textit{pl} \right\} \\ \text{case} & \left\{ \textit{nom} \;\; \textit{acc} \right\} \end{bmatrix} \end{bmatrix}
$$

$$
\begin{bmatrix} \text{agr} & \left\{ \begin{bmatrix} \text{cas} & \text{nom} \\ \text{gen} & \text{masc} \\ \text{num} & \text{sg} \end{bmatrix} \begin{bmatrix} \text{cas} & \text{gen} \\ \text{gen} & \text{fem} \\ \text{num} & \text{sg} \end{bmatrix} \begin{bmatrix} \text{cas} & \text{dat} \\ \text{gen} & \text{fem} \\ \text{num} & \text{sg} \end{bmatrix} \begin{bmatrix} \text{cas} & \text{gen} \\ \text{num} & \text{pl} \end{bmatrix} \right\} \end{bmatrix}
$$

Rules with complex categories

- categories with complexity level information

$$\begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 2 \end{bmatrix} \rightarrow \begin{bmatrix} \text{cat} & \text{D} \end{bmatrix} \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 1 \end{bmatrix}$$

- modelling of government

$$\begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 1 \end{bmatrix} \rightarrow \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 0 \end{bmatrix} \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 2 \\ \text{cas} & \text{gen} \end{bmatrix}$$

Rules with complex categories

- representing the rule structure as a feature structure

  example: binary branching rule: $X0 \rightarrow X1\ X2$

$$\begin{bmatrix} \text{X0} & \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 2 \end{bmatrix} \\ \text{X1} & \begin{bmatrix} \text{cat} & \text{D} \\ \text{bar} & 0 \end{bmatrix} \\ \text{X2} & \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 1 \end{bmatrix} \end{bmatrix}$$

Rules with complex categories

- representation of feature structures as path equations

$$
\begin{bmatrix}
\text{X0} & \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 2 \end{bmatrix} \\
\text{X1} & \begin{bmatrix} \text{cat} & \text{D} \\ \text{bar} & 0 \end{bmatrix} \\
\text{X2} & \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 1 \end{bmatrix}
\end{bmatrix}
\Longrightarrow
\begin{array}{l}
\langle \text{ XO cat } \rangle = \text{ N} \\
\langle \text{ XO bar } \rangle = \text{ 2} \\
\langle \text{ X1 cat } \rangle = \text{ D} \\
\langle \text{ X1 bar } \rangle = \text{ 0} \\
\langle \text{ X2 cat } \rangle = \text{ N} \\
\langle \text{ X2 bar } \rangle = \text{ 1}
\end{array}
$$

- features may corefer (coreference, reentrancy, structure sharing)

Rules with complex categories

- applications of coreference:
  - agreement: $\langle$ X1 agr $\rangle = \langle$ X2 agr $\rangle$
  - projection: $\langle$ X0 agr $\rangle = \langle$ X2 agr $\rangle$

Rules with complex categories

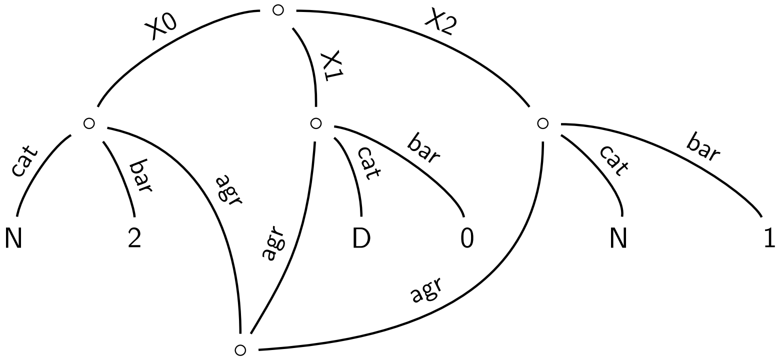
- representation in feature matricees by means of coreference marker or path equations

$$\begin{bmatrix} \text{X0} & \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 2 \\ \text{agr} & \boxed{1} \end{bmatrix} \\ \text{X1} & \begin{bmatrix} \text{cat} & \text{D} \\ \text{bar} & 0 \\ \text{agr} & \boxed{1} \end{bmatrix} \\ \text{X2} & \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 1 \\ \text{agr} & \boxed{1} \end{bmatrix} \end{bmatrix} \qquad \begin{bmatrix} \text{X0} & \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 2 \\ \text{agr} & \end{bmatrix} \\ \text{X1} & \begin{bmatrix} \text{cat} & \text{D} \\ \text{bar} & 0 \\ \text{agr} & = \langle \text{ X0 agr } \rangle \end{bmatrix} \\ \text{X2} & \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 1 \\ \text{agr} & = \langle \text{ X0 agr } \rangle \end{bmatrix} \end{bmatrix}$$

- coreference corresponds to a named variable

Rules with complex categories

- feature structures with coreference correspond to a directed acyclic graph



Rules with complex categories

- generalised adjunct rule for prepositional phrases

$$\begin{bmatrix} \mathrm{X0} & \begin{bmatrix} \mathrm{cat} & \boxed{1} \\ \mathrm{bar} & 1 \end{bmatrix} \\ \mathrm{X1} & \begin{bmatrix} \mathrm{cat} & \boxed{1} \\ \mathrm{bar} & 1 \end{bmatrix} \\ \mathrm{X2} & \begin{bmatrix} \mathrm{cat} & \mathrm{P} \\ \mathrm{bar} & 2 \end{bmatrix} \end{bmatrix}$$

Rules with complex categories

- consequences of coreference on the information content:
  - structural equality (type identity): $\begin{bmatrix} \text{x} & [\,] \\ \text{y} & [\,] \end{bmatrix}$
  - referential identity (token identity): $\begin{bmatrix} \text{x} & \boxed{1}\,[\,] \\ \text{y} & \boxed{1} \end{bmatrix}$
  - a coreference is an additional constraint
  - equality is more general than identity: $\begin{bmatrix} \text{x} & [\,] \\ \text{y} & [\,] \end{bmatrix} \sqsubseteq \begin{bmatrix} \text{x} & \boxed{1}\,[\,] \\ \text{y} & \boxed{1} \end{bmatrix}$
- definition of unification is not affected by the introduction of coreference

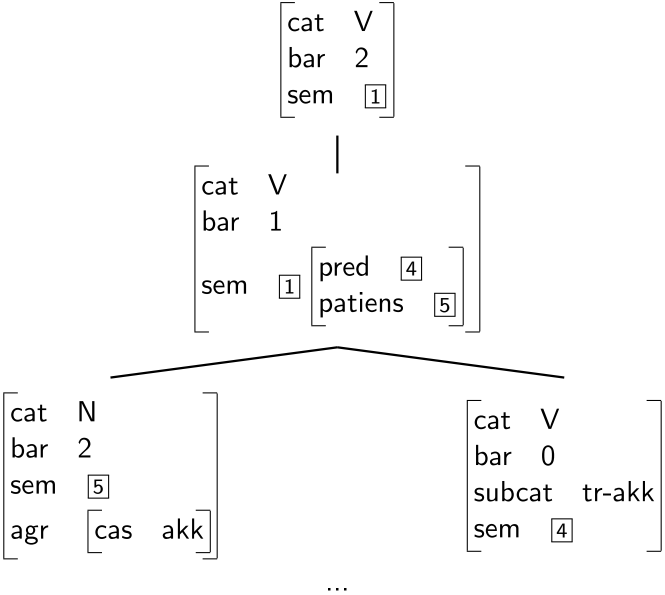
Rules with complex categories

- construction of arbitrary structural descriptions
  e.g. logical form

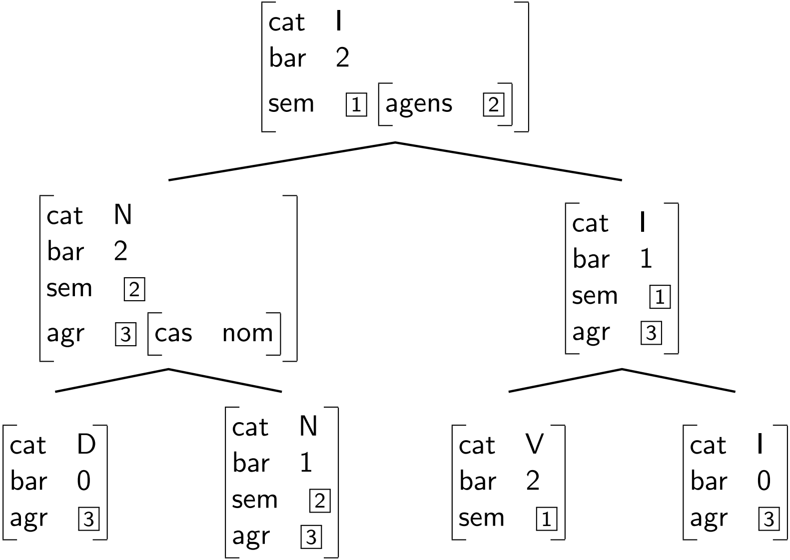
$$\begin{bmatrix} \text{cat} & \text{I} \\ \text{bar} & 2 \\ \text{sem} & \boxed{1}\begin{bmatrix} \text{agens} & \boxed{2} \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 2 \\ \text{sem} & \boxed{2} \\ \text{agr} & \boxed{3}\begin{bmatrix} \text{cas} & \text{nom} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \text{cat} & \text{I} \\ \text{bar} & 1 \\ \text{sem} & \boxed{1} \\ \text{agr} & \boxed{3} \end{bmatrix}$$

$$\begin{bmatrix} \text{cat} & \text{V} \\ \text{bar} & 1 \\ \text{sem} & \begin{bmatrix} \text{pred} & \boxed{1} \\ \text{patiens} & \boxed{2} \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 2 \\ \text{sem} & \boxed{2} \\ \text{agr} & \begin{bmatrix} \text{cas} & \text{akk} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \text{cat} & \text{V} \\ \text{bar} & 0 \\ \text{subcat} & \text{tr-akk} \\ \text{sem} & \boxed{1} \end{bmatrix}$$

Rules with complex categories



Rules with complex categories



…

Rules with complex categories

- construction of left recursive structures with right recursive rules
- left recursive rules (DCG-notation)

```
np(np(Snp,Spp)) --> np(Snp), pp(Spp).
np(np(Sd,Sn)) --> d(Sd), n(Sn).
```

- right recursive rules

```
np(np(Sd,Sn)) --> d(Sd), n(Sn).
np(Spps) --> d(Sd), n(Sn), pps(np(Sd,Sn),Spps).

pps(Snp,np(Snp,Spp)) --> pp(Spp).
pps(Snp,Spps) --> pp(Spp), pps(np(Snp,Spp),Spps).
```

Rules with complex categories

- example: *the house behind the street with the red roof*

```
?- np(S,[t,h,bts,wtrr],[ ]).
   np(Spps1) --> d(Sd), n(Sn), pps(np(Sd,Sn),Spps1).        S=Spps1
   . . .
   ?- pps(np(d(t),n(h)),Spps1,[bts,wtrr],Z1).
      pps(Snp2,Spps2) --> pp(Spp), pps(np(Snp,Spp),Spps2). Spps1=Spps2
      . . .
      ?- pps(np(np(d(t),n(h)),pp(bts)),Spps2,[wtrr],Z2)
         pps(Snp,np(Snp,Spp)) --> pp(Spp).

Snp = np(np(d([t]),n([h])),pp([bts])),
Spps2 = np(np(np(d([t]),n([h])),pp([bts])),pp([wtrr])
```

Rules with complex categories

- parsing with complex categories
  - test for identity has to be replaced by unifiability
  - but: unification is destructive
    - information is added to rules or lexical entries
    - feature structures need to be copied prior to unification

Subcategorization

- modelling of valence requirements as a list

$$\textit{geben:}\quad \begin{bmatrix} \text{cat} & \text{V} \\ \text{bar} & 0 \\ \text{subcat} & \begin{bmatrix} \text{first} & \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 2 \\ \text{agr}|\text{cas} & \text{akk} \end{bmatrix} \\ \text{rest} & \begin{bmatrix} \text{first} & \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 2 \\ \text{agr}|\text{cas} & \text{dat} \end{bmatrix} \\ \text{rest} & \text{nil} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Subcategorisation

- processing of the information by means of suitable rules

$$\begin{bmatrix} \text{cat} & \text{V} \\ \text{bar} & 0 \\ \text{subcat} & \boxed{1} \end{bmatrix} \rightarrow \boxed{2}\begin{bmatrix} \; \end{bmatrix} \begin{bmatrix} \text{cat} & \text{V} \\ \text{bar} & 0 \\ \text{subcat} & \begin{bmatrix} \text{first} & \boxed{2} \\ \text{rest} & \boxed{1} \end{bmatrix} \end{bmatrix} \qquad \text{rule 1}$$

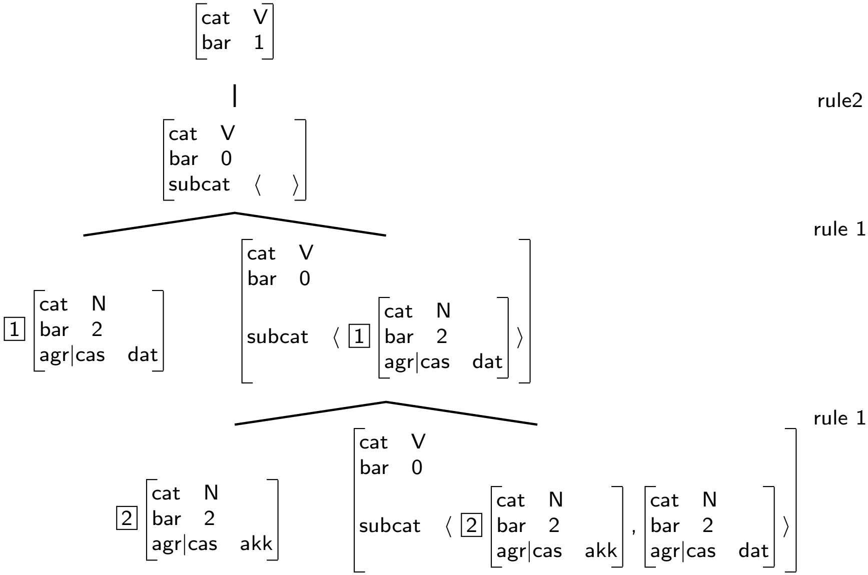
$$\begin{bmatrix} \text{cat} & \text{V} \\ \text{bar} & 1 \end{bmatrix} \rightarrow \begin{bmatrix} \text{cat} & \text{V} \\ \text{bar} & 0 \\ \text{subcat} & \text{nil} \end{bmatrix} \qquad \text{rule 2}$$

Subcategorisation

- list notation

$$\textit{geben:}\quad \begin{bmatrix} \text{cat} & \text{V} \\ \text{bar} & 0 \\ \text{subcat} & \left\langle \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 2 \\ \text{agr}|\text{cas} & \text{akk} \end{bmatrix}, \begin{bmatrix} \text{cat} & \text{N} \\ \text{bar} & 2 \\ \text{agr}|\text{cas} & \text{dat} \end{bmatrix} \right\rangle \end{bmatrix}$$

Subcategorisation



Movement

- movement operations are unidirectional and procedural
- goal: declarative integration into feature structures
- slash operator

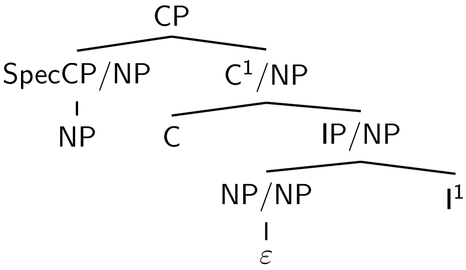
S/NP	sentence without a noun phrase
VP/V	verb phrase without a verb
S/NP/NP	
...	

  - first used in categorial grammar (Bar-Hillel 1963)
  - also order sensitive variant: S\NP/NP

Movement

- topicalization

CP $\rightarrow$ SpecCP/NP C^1/NP	
SpecCP/NP $\rightarrow$ NP	slash introduction
C^1/NP $\rightarrow$ C IP/NP	slash transition
IP/NP $\rightarrow$ NP/NP I^1	slash transition
NP/NP $\rightarrow$ ε	slash elimination



Movement

- encoding in feature structures: slash feature
  - moved constituents are connected to their trace by means of coreference
  - computation of the logical form is invariant against movement operations

Constraint-based models

- head-driven phrase-structure grammar (HPSG, Pollard and Sag 1987, 1994)
- inspired by the principles & parameter model of Chomsky (1981)
- constraints: implications over feature structures:
  if the premise can be unified with a feature structure unify the consequence with that structure.

$$\begin{bmatrix} \textit{type}_1 \\ \end{bmatrix} \rightarrow \begin{bmatrix} \text{X1}|\ldots|\ \text{XN} & \boxed{1} \\ \text{Y1}|\ldots|\text{YM} & \boxed{1} \end{bmatrix}$$

Constraint-based models

- feature structures need to be typed

Haus:
$$\begin{bmatrix} \textit{nomen} & \\ \text{cat} & \text{N} \\ \text{agr} & \begin{bmatrix} \textit{agr} & \\ \text{case} & \text{nom} \\ \text{num} & \text{sg} \\ \text{gen} & \text{neutr} \end{bmatrix} \end{bmatrix}$$

- extention of unification and subsumtion to typed feature structures
  - subsumtion:

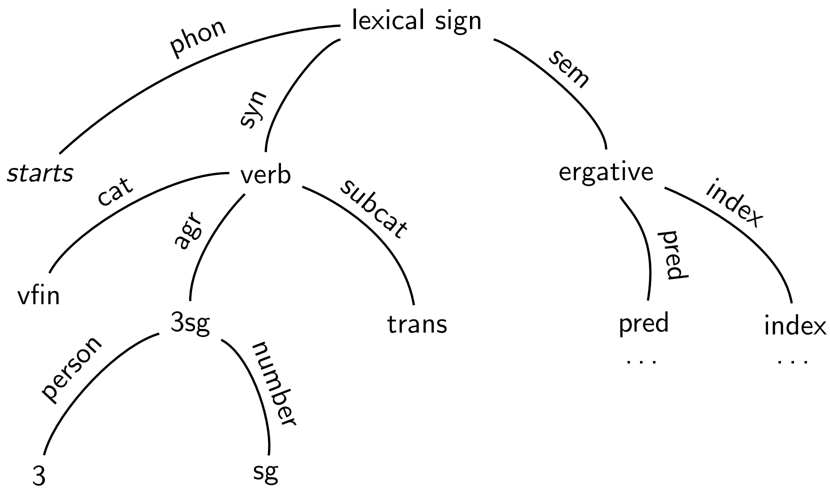
    $\text{M}_i^m \sqsubseteq \text{M}_j^n$ gdw. $\text{M}_i \sqsubseteq \text{M}_j$ und $m = n$

  - unification:

    $\text{M}_i^m \sqcup \text{M}_j^n = \text{M}_k^o$ gdw. $\text{M}_k = \text{M}_i \sqcup \text{M}_j$ und $m = n = o$

Constraint-based models

- graphical interpretation: types as node annotations



Constraint-based models

- types are organized in a type hierarchy:
  - partial order for types:

    sub(*verb*,*finite*)
    sub(*verb*,*finite*)
    ...
  - hierarchical abstraction
- subsumtion for types:

$$m \sqsubseteq n \quad \text{iff} \quad \begin{cases} \text{sub}(m, n) \\ \text{sub}(m, x) \wedge \text{sub}(x, n) \end{cases}$$

- unification for types:

$$m \sqcup n = o \quad \text{iff} \quad \begin{array}{l} m \sqsubseteq o \wedge n \sqsubseteq o \quad \text{and} \\ \neg\exists x . m \sqsubseteq x \wedge n \sqsubseteq x \wedge x \sqsubseteq o \end{array}$$

Constraint-based models

- subsumtion for typed feature structures:

$$\mathrm{M}_i^m \sqsubseteq \mathrm{M}_j^n \quad \text{iff} \quad \begin{array}{l} \mathrm{M}_i \sqsubseteq \mathrm{M}_j \quad \text{and} \\ \mathrm{m} \sqsubseteq \mathrm{n} \end{array}$$

- unification for typed feature structures:

$$\mathrm{M}_i^m \sqcup \mathrm{M}_j^n = \mathrm{M}_k^o \quad \text{iff} \quad \begin{array}{l} \mathrm{M}_k = \mathrm{M}_i \sqcup \mathrm{M}_j \quad \text{and} \\ o = m \sqcup n \end{array}$$

Constraint-based models

- HPSG: lexical signs

- *word*
  - PHON
  - SYNSEM
    - *synsem*
      - LOC
        - *local*
          - CAT
            - *cat*
              - HEAD
              - SUBCAT
          - CONT
            - *npro/ppro*
              - INDEX
              - RESTR
          - CONX
            - BACKGR { [*psoa*], . . . }
      - NONLOC

Constraint-based models

- HPSG: phrasal signs
  - signs of type *phrase*
    additional features: Daughters, (Quantifier-Store)
  - most important special case:
    head-comp-struc

Constraint-based models

- DAUGHTERS (DTRS)
  - constituent structure of a phrase
  - HEAD-DTR (*phrase*)
  - COMP-DTRS (list of elementes of type *phrase*)

$$\begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \langle \textit{Kim, walks} \rangle \\ \text{SYNSEM} & \text{S}[\textit{fin}] \\ \text{DTRS} & \begin{bmatrix} \textit{head-comp-struc} & \\ \text{HEAD-DTR} & \begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \langle \textit{walks} \rangle \\ \text{SYNSEM} & \text{VP}[\textit{fin}] \end{bmatrix} \\ \text{COMP-DTRS} & \left\langle \begin{bmatrix} \textit{phrase} & \\ \text{PHON} & \langle \textit{Kim} \rangle \\ \text{SYNSEM} & \text{NP}[\textit{nom}] \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

Constraint-based models

- head-feature principle
  - projection of head features to the phrase level
  - the HEAD-feature of a head structure corefers with the HEAD-feature of its head daughter.

$$\begin{bmatrix}\text{DTRS} & \begin{bmatrix}\textit{head-struc}\\ \\ \end{bmatrix}\end{bmatrix} \quad \rightarrow$$

$$\begin{bmatrix}\text{SYNSEM}|\text{LOC}|\text{CAT}|\text{HEAD} & \boxed{1}\\ \text{DTRS}|\text{HEAD-DTR}|\text{SYNSEM}|\text{LOC}|\text{CAT}|\text{HEAD} & \boxed{1}\end{bmatrix}$$