

# 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

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

# 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

<i>Haus:</i>	$\left[ \begin{array}{ll} \text{cat} & \text{N} \\ \text{case} & \text{nom} \\ \text{num} & \text{sg} \\ \text{gen} & \text{neutr} \end{array} \right]$	<i>house:</i>	$\left[ \begin{array}{ll} \text{cat} & \text{N} \\ \text{num} & \text{sg} \end{array} \right]$
--------------	---	---------------	--

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

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

## 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  $M_2$ .

→ 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)

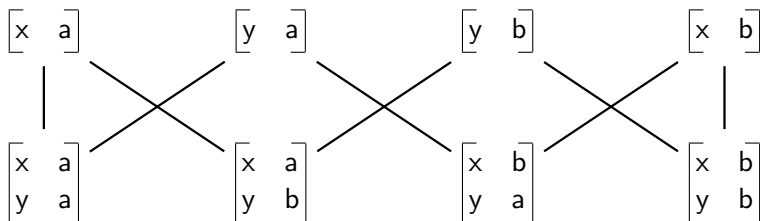
but:

- $M_1$  is more general than  $M_2$
- alternative notation:

instance-based (POLLARD UND SAG 1987):  $M_1 \succeq M_2$

# Feature structures

- subsumption hierarchy





# Feature structures

- formal properties of subsumtion
  - reflexive:  $\forall M_i. M_i \sqsubseteq M_i$
  - transitive:  $\forall M_i \forall M_j \forall M_k. M_i \sqsubseteq M_j \wedge M_j \sqsubseteq M_k \rightarrow M_i \sqsubseteq M_k$
  - antisymmetrical:  $\forall M_i \forall M_j. M_i \sqsubseteq M_j \wedge M_j \sqsubseteq M_i \rightarrow M_i = M_j$
- subsumtion relation defines a partial order
- not all feature structures need to be in a subsumtion relation

# Feature structures

- unification I (subsumption-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

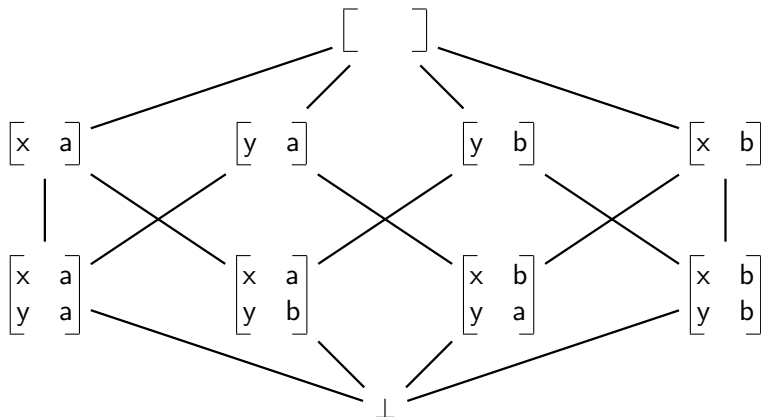
- $M_3$  is subsumed by  $M_1$  and  $M_2$  and
  - $M_3$  subsumes all other feature structures, that are also subsumed by  $M_1$  and  $M_2$ .
- result of a unification ( $M_3$ ) is the most general feature structure which is subsumed by  $M_1$  and  $M_2$

# Feature structures

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

# Feature structures

- subsumption 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:  $M \sqcup M = M$
  - commutative:  $M_i \sqcup M_j = M_j \sqcup M_i$
  - associative:  $(M_i \sqcup M_j) \sqcup M_k = M_i \sqcup (M_j \sqcup M_k)$
  - neutral element:  $\top \sqcup M = M$
  - zero element:  $\perp \sqcup M = \perp$
- unification and subsumption can be mutually defined from each other
  - $M_i \sqsubseteq M_j \leftrightarrow M_i \sqcup M_j = M_j$

# 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*: 
$$\left[ \begin{array}{l} \text{cat} \text{ Pro} \\ \text{bar} \text{ 0} \\ \text{agr} \left[ \begin{array}{l} \text{pers} \text{ 3rd} \\ \text{num} \text{ sg} \\ \text{gen} \text{ fem} \\ \text{case} \text{ nom} \end{array} \right] \end{array} \right]$$

*us*: 
$$\left[ \begin{array}{l} \text{cat} \text{ Pro} \\ \text{bar} \text{ 0} \\ \text{agr} \left[ \begin{array}{l} \text{pers} \text{ 1st} \\ \text{num} \text{ pl} \\ \text{case} \text{ acc} \end{array} \right] \end{array} \right]$$

# 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)

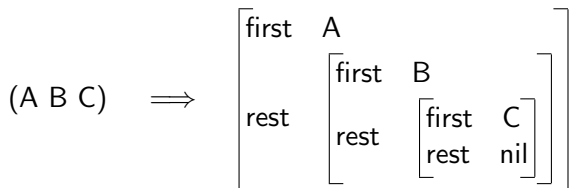
Two feature structures  $M_1$  and  $M_2$  unify, iff for every common feature of both structures

- 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  $\perp$ .

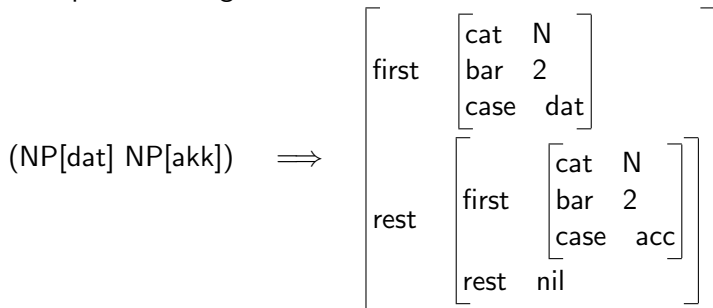
# Feature structures

- recursive data structures can be used
  - lists
  - trees



# Feature structures

- example: subcategorisation list



- 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

$$\left[ \text{agr} \left[ \begin{array}{ll} \text{pers} & 2\text{nd} \\ \text{num} & \{ sg \ pl \} \\ \text{case} & \{ nom \ acc \} \end{array} \right] \right]$$

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

# 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:  $X_0 \rightarrow X_1 X_2$

$X_0$	$\begin{bmatrix} \text{cat} & N \\ \text{bar} & 2 \end{bmatrix}$
$X_1$	$\begin{bmatrix} \text{cat} & D \\ \text{bar} & 0 \end{bmatrix}$
$X_2$	$\begin{bmatrix} \text{cat} & N \\ \text{bar} & 1 \end{bmatrix}$

## Rules with complex categories

- representation of feature structures as path equations

$$\left[ \begin{array}{l} X0 \\ X1 \\ X2 \end{array} \left[ \begin{array}{l} \text{cat} \\ \text{bar} \end{array} \right] \left[ \begin{array}{l} N \\ 2 \end{array} \right] \right] \Rightarrow \begin{array}{l} \langle X0 \text{ cat} \rangle = N \\ \langle X0 \text{ bar} \rangle = 2 \\ \langle X1 \text{ cat} \rangle = D \\ \langle X1 \text{ bar} \rangle = 0 \\ \langle X2 \text{ cat} \rangle = N \\ \langle X2 \text{ bar} \rangle = 1 \end{array}$$

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

# Rules with complex categories

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



## Rules with complex categories

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

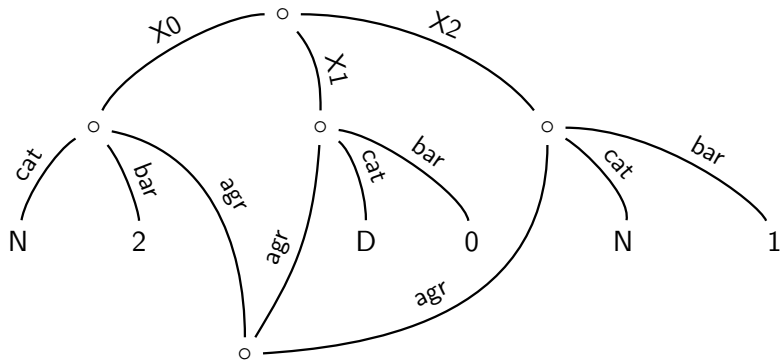
$$\left[ \begin{array}{l} X0 \\ X1 \\ X2 \end{array} \left[ \begin{array}{ll} \text{cat} & N \\ \text{bar} & 2 \\ \text{agr} & \boxed{1} \end{array} \right] \right]$$
$$\left[ \begin{array}{l} X1 \\ X2 \end{array} \left[ \begin{array}{ll} \text{cat} & D \\ \text{bar} & 0 \\ \text{agr} & \boxed{1} \end{array} \right] \right]$$
$$\left[ \begin{array}{l} X2 \end{array} \left[ \begin{array}{ll} \text{cat} & N \\ \text{bar} & 1 \\ \text{agr} & \boxed{1} \end{array} \right] \right]$$

$$\left[ \begin{array}{l} X0 \\ X1 \\ X2 \end{array} \left[ \begin{array}{ll} \text{cat} & N \\ \text{bar} & 2 \\ \text{agr} & \end{array} \right] \right]$$
$$\left[ \begin{array}{l} X1 \\ X2 \end{array} \left[ \begin{array}{ll} \text{cat} & D \\ \text{bar} & 0 \\ \text{agr} & = \langle X0 \text{ agr} \rangle \end{array} \right] \right]$$
$$\left[ \begin{array}{l} X2 \end{array} \left[ \begin{array}{ll} \text{cat} & N \\ \text{bar} & 1 \\ \text{agr} & = \langle X0 \text{ agr} \rangle \end{array} \right] \right]$$

- 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

X0	cat	1
	bar	1
X1	cat	1
	bar	1
X2	cat	P
	bar	2

## Rules with complex categories

- consequences of coreference on the information content:

- structural equality (type identity):  $\begin{bmatrix} x & [ ] \\ y & [ ] \end{bmatrix}$

- referential identity (token identity):  $\begin{bmatrix} x & \boxed{1} & [ ] \\ y & \boxed{1} & [ ] \end{bmatrix}$

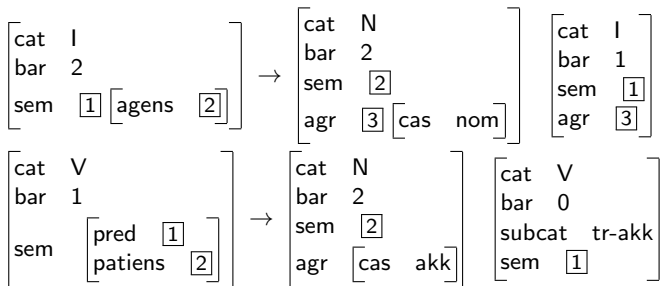
- a coreference is an additional constraint

- equality is more general than identity:  $\begin{bmatrix} x & [ ] \\ y & [ ] \end{bmatrix} \sqsubseteq \begin{bmatrix} x & \boxed{1} & [ ] \\ y & \boxed{1} & [ ] \end{bmatrix}$

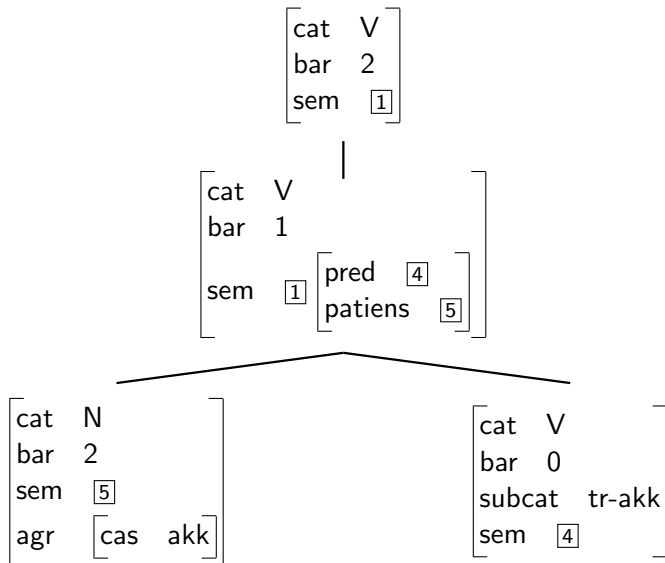
- definition of unification is not affected by the introduction of coreference

## Rules with complex categories

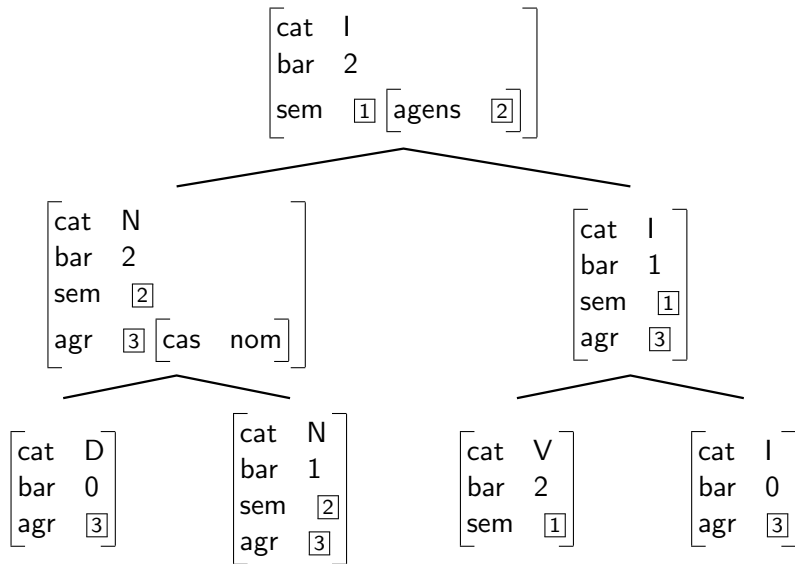
- construction of arbitrary structural descriptions  
e.g. logical form



## 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)

$\text{np}(\text{np}(\text{Snp}, \text{Spp})) \text{ --> } \text{np}(\text{Snp}), \text{pp}(\text{Spp}).$

$\text{np}(\text{np}(\text{Sd}, \text{Sn})) \text{ --> } \text{d}(\text{Sd}), \text{n}(\text{Sn}).$

- right recursive rules

$\text{np}(\text{np}(\text{Sd}, \text{Sn})) \text{ --> } \text{d}(\text{Sd}), \text{n}(\text{Sn}).$

$\text{np}(\text{Spps}) \text{ --> } \text{d}(\text{Sd}), \text{n}(\text{Sn}), \text{pps}(\text{np}(\text{Sd}, \text{Sn}), \text{Spps}).$

$\text{pps}(\text{Snp}, \text{np}(\text{Snp}, \text{Spp})) \text{ --> } \text{pp}(\text{Spp}).$

$\text{pps}(\text{Snp}, \text{Spps}) \text{ --> } \text{pp}(\text{Spp}), \text{pps}(\text{np}(\text{Snp}, \text{Spp}), \text{Spps}).$



## 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])),
```

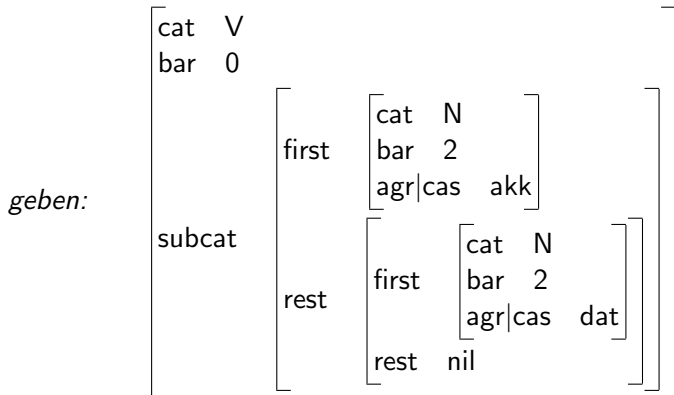
```
Spps2 = np(np(np(d([t]),n([h])),pp([bts])),pp([wtrr]))
```

# 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



# Subcategorisation

- processing of the information by means of suitable rules

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

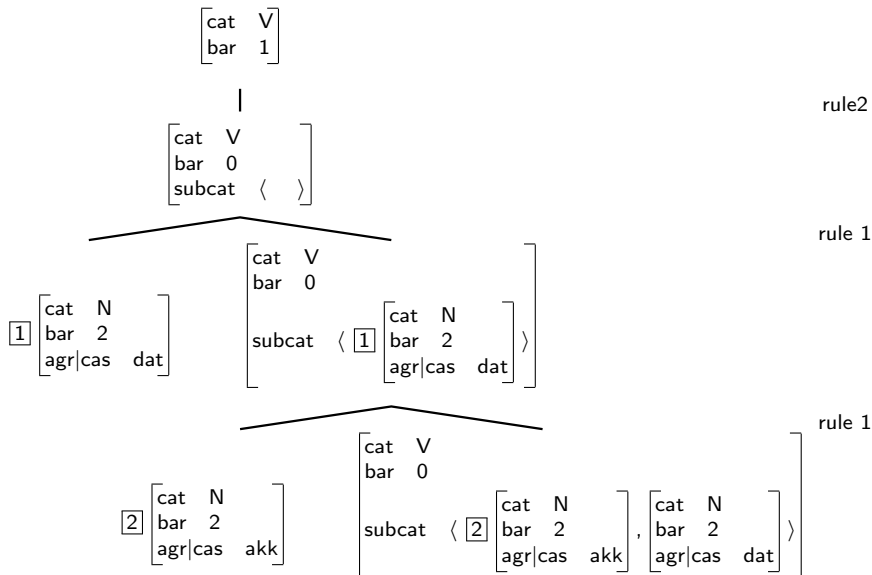
$$\begin{bmatrix} \text{cat} & V \\ \text{bar} & 1 \end{bmatrix} \rightarrow \begin{bmatrix} \text{cat} & V \\ \text{bar} & 0 \\ \text{subcat} & \text{nil} \end{bmatrix} \quad \text{rule 2}$$

# Subcategorisation

- list notation

*geben*: 
$$\left[ \begin{array}{l} \text{cat } V \\ \text{bar } 0 \\ \text{subcat } \left\langle \left[ \begin{array}{l} \text{cat } N \\ \text{bar } 2 \\ \text{agr|cas } \text{akk} \end{array} \right], \left[ \begin{array}{l} \text{cat } N \\ \text{bar } 2 \\ \text{agr|cas } \text{dat} \end{array} \right] \right\rangle \end{array} \right]$$

# 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:  $S \backslash NP / NP$

# Movement

- topicalization

$CP \rightarrow \text{SpecCP/NP} \quad C^1/\text{NP}$

$\text{SpecCP/NP} \rightarrow \text{NP}$

$C^1/\text{NP} \rightarrow C \quad \text{IP/NP}$

$\text{IP/NP} \rightarrow \text{NP/NP} \quad I^1$

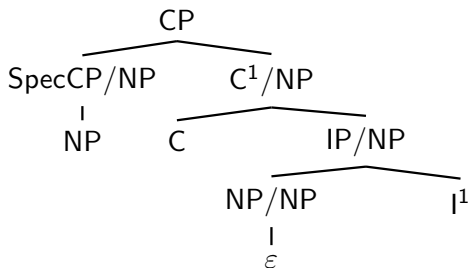
$\text{NP/NP} \rightarrow \varepsilon$

slash introduction

slash transition

slash transition

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

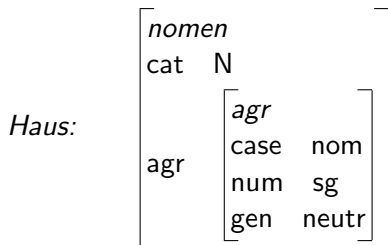
- head-driven phrase-structure grammar (HPSG, POLLARD AND SAG 1987, 1994)
- inspired by the principles & parameter model of Chomsky (1981)
- constraints: implications over feature structures:  
if the premise can be unified with a feature structure unify the consequence with that structure.

$$\boxed{type_1} \rightarrow \boxed{\begin{array}{l} X1 | \dots | XN \quad \boxed{1} \\ Y1 | \dots | YM \quad \boxed{1} \end{array}}$$

- can be used to model principles of universal grammar

# Constraint-based models

- feature structures need to be typed



- extension of unification and subsumtion to typed feature structures

- subsumtion:

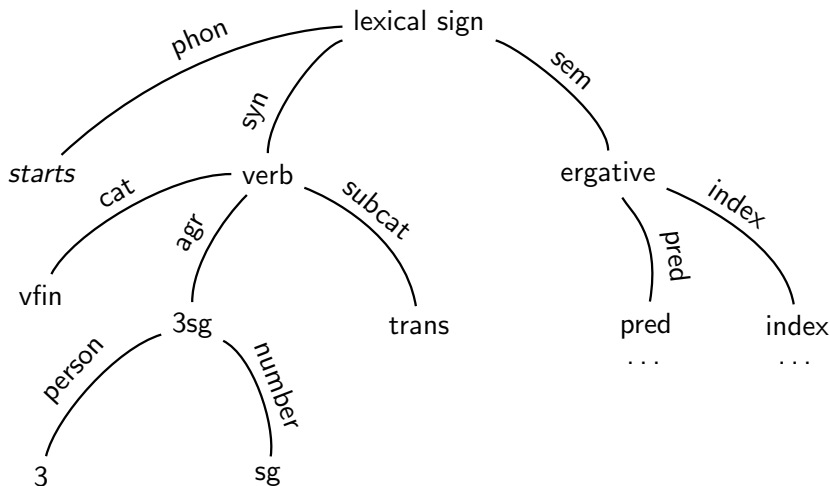
$$M_i^m \sqsubseteq M_j^n \text{ gdw. } M_i \sqsubseteq M_j \text{ und } m = n$$

- unification:

$$M_i^m \sqcup M_j^n = M_k^o \text{ gdw. } M_k = M_i \sqcup M_j \text{ und } m = n = o$$

## Constraint-based models

- graphical interpretation: types as node annotations



## Constraint-based models

- types are organized in a type hierarchy:

- partial order for types:

$\text{sub}(\text{verb}, \text{finite})$

$\text{sub}(\text{verb}, \text{infinite})$

...

- hierarchical abstraction

- subsumption 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:

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

# Constraint-based models

- subsumption for typed feature structures:

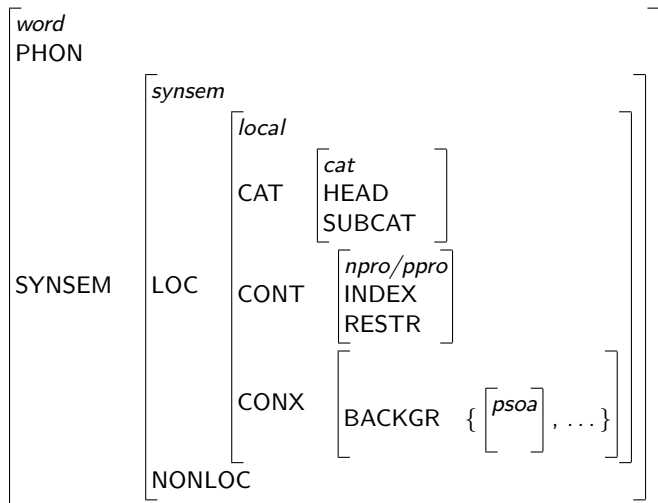
$$M_i^m \sqsubseteq M_j^n \quad \text{iff} \quad \begin{array}{l} M_i \sqsubseteq M_j \quad \text{and} \\ m \sqsubseteq n \end{array}$$

- unification for typed feature structures:

$$M_i^m \sqcup M_j^n = M_k^o \quad \text{iff} \quad \begin{array}{l} M_k = M_i \sqcup M_j \quad \text{and} \\ o = m \sqcup n \end{array}$$

# Constraint-based models

- HPSG: lexical signs



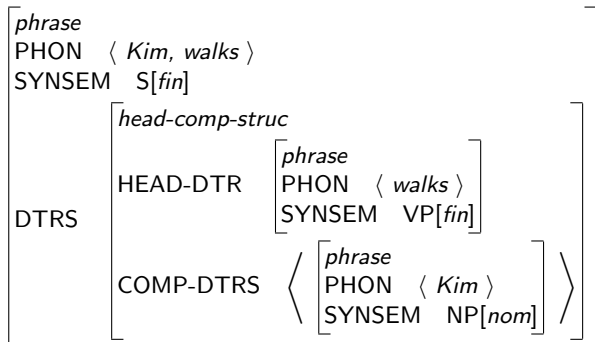
# Constraint-based models

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



# Constraint-based models

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

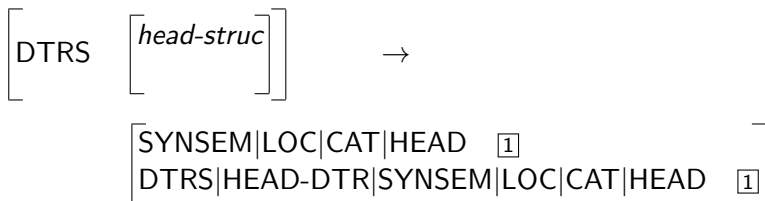


# Principles and Parameters

- universal grammar
  - sign hierarchy: universally available types (with type definitions)
  - dominance schemata:  
limited inventory of universally applicable phrase types
    - head-complement structures, head-adjunct structures, ...
  - universal constraints
    - head feature principle, subcategorisation principle, ...
- language specific grammar
  - lexicon (possibly supplemented by lexical rules)
  - specialisations of the sign hierarchy
  - additional or specialised dominance schemata

# 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.



## Constraint-based models

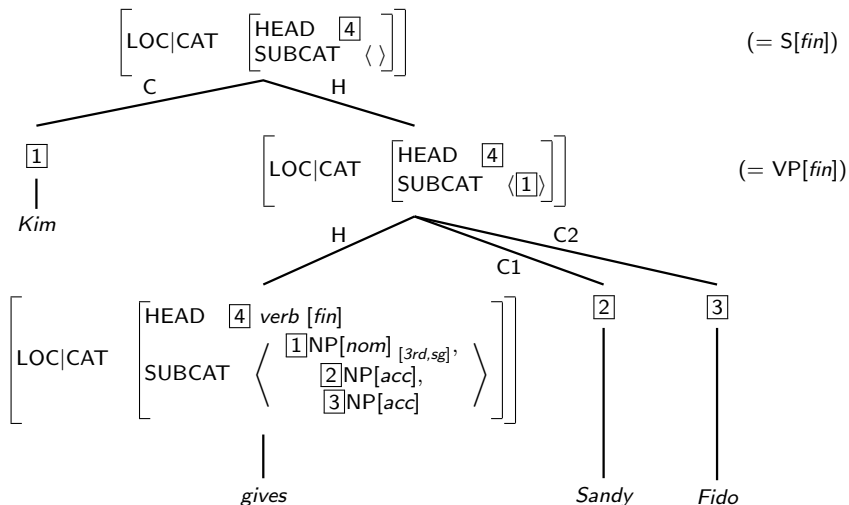
- subcategorisation principle:

In a head-complement-phrase the SUBCAT-value of the head daughter is equal to the combination of the SUBCAT-list of the phrase with the SYNSEM-values of the complement daughters (arranged according to increasing obliqueness).

$$\left[ \text{DTRS} \left[ \textit{head-compl-struct} \right] \right] \rightarrow \left[ \begin{array}{l} \text{SYNSEM|LOC|CAT|SUBCAT} \quad \boxed{1} \\ \text{DTRS} \left[ \begin{array}{l} \text{HEAD-DTR|SYNSEM|LOC|CAT|SUBCAT} \quad \text{append}(\boxed{1}, \boxed{2}) \\ \text{COMP-DTRS} \quad \boxed{2} \end{array} \right] \end{array} \right]$$

# Constraint-based models

- subcategorization principle:



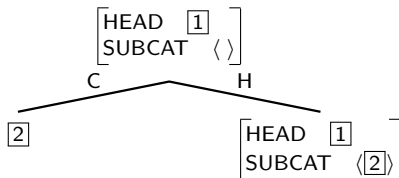
# Dominance Schemata

- disjunctively specified principle: every phrase instantiates one of a finite set of structural patterns

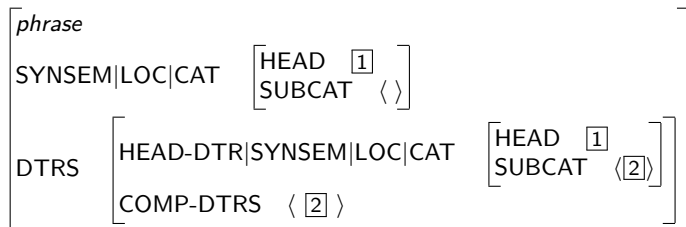
$$\boxed{\textit{phrase}} \rightarrow \text{schema}_1 \vee \dots \vee \text{schema}_n$$

- Schema 1:

a saturated phrase ( $\boxed{\text{SUBCAT } \langle \rangle}$ ) with a DTRS value of type *head-comp-structure* where the value of feature HEAD-DTRS is a phrasal sign and the value of COMP-DTRS is a list of length one.



# Dominance Schemata

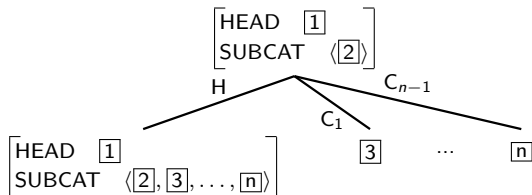


- immediate consequence of the subcategorisation and head feature principles
- licenses phrases like
$$\begin{array}{l} S \rightarrow NP VP \\ NP \rightarrow \text{Det } N^1 \end{array}$$

# Dominance Schemata

- Schema 2:

an almost saturated phrase with a single subcat element remaining, a DTRS value of type *head-comp-struct* and a lexical sign as head daughter



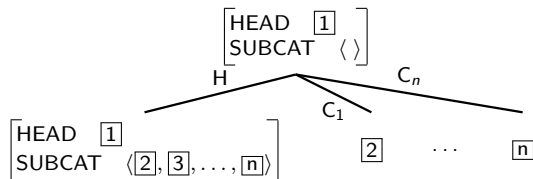
- licenses verb phrases including all complements of the head
- complexity levels of the  $\bar{X}$  theory are replaced by the distinctions between
  - lexical / phrasal signs
  - saturated / unsaturated phrases



# Dominance Schemata

- Schema 3:

a saturated phrase ( $\llbracket \text{SUBCAT } \langle \rangle \rrbracket$ ) with a DTRS value of type *head-comp-structure* and a lexical head



- licenses “scrambling” structures: almost free phrase order (including the subject) e.g. German, Japanese
- Schema 4: head-marker structures (*that John left*)
- Schema 5: head-adjunct structures (e.g. adjective modifiers)
  - idea: adjuncts select their head

## Constraint-based models

- more constraints for constructing a semantic description (predicate-argument structure, quantor handling, ...)
- advantages of principle-based modelling:
  - modularization: general requirements (e.g. agreement, construction of a semantic representation) are implemented once and not repeatedly in various rules
  - object-oriented approach: heavy use of inheritance
  - context-free backbone of the grammar is removed almost completely; only very few general structural schemata remain (head-complement structure, head-adjunct structure, coordinated structure, ...)
  - integrated treatment of semantics in a general form