Linearization Grammars

Detmar Meurers
Ohio State University
dm@ling.osu.edu

Stefan Müller
Universität Bremen
Stefan.Mueller@cl.uni-bremen.de

ESSLLI 2003, Wien

Course WWW: http://ling.osu.edu/~dm/03/esslli/ or http://www.cl.uni-bremen.de/~stefan/Lehre/ESSLLI2003/

General Remarks

- Who are you?
  - Name
  - Affiliation
  - Country

General Remarks

- Who are you?
  - Name
  - Affiliation
  - Country
- The reader contains some important articles. The reader is not on the CD Rom. It will not be availible in the net.

General Remarks

- Who are you?
  - Name
  - Affiliation
  - Country

- The reader contains some important articles. The reader is not on the CD Rom. It will not be availible in the net. In addition to the reader there are handouts (= copies of the slides).

General Remarks

- Who are you?
  - Name
  - Affiliation
  - Country
- The reader contains some important articles. The reader is not on the CD Rom. It will not be availible in the net. In addition to the reader there are handouts (= copies of the slides).
- Bring the handouts and write down additional notes.

General Remarks

- Who are you?
  - Name
  - Affiliation
  - Country
- The reader contains some important articles. The reader is not on the CD Rom. It will not be availible in the net. In addition to the reader there are handouts (= copies of the slides).
- Bring the handouts and write down additional notes.
- Ask questions!!!
  - During the lectures and during the breaks.
  - Do not hesitate to write us if you encounter problems after the course.

Outline

- Phrase Structure
- A Topological Model for German
- From Phrase Structure to GPSG and HPSG
- German Word Order Phenomena and their Analyses in Standard-HPSG/GPSG
- Discontinuous Constituents
- Parsing
- Syntactically Annotated Corpora and Discontinuity
- Summary

Phrase Structure (I)

- Syntactic constituency is a central notion in generative linguistics.

Phrase Structure (I)

- Syntactic constituency is a central notion in generative linguistics.
- Tree structures are used to represent syntactic constituency.

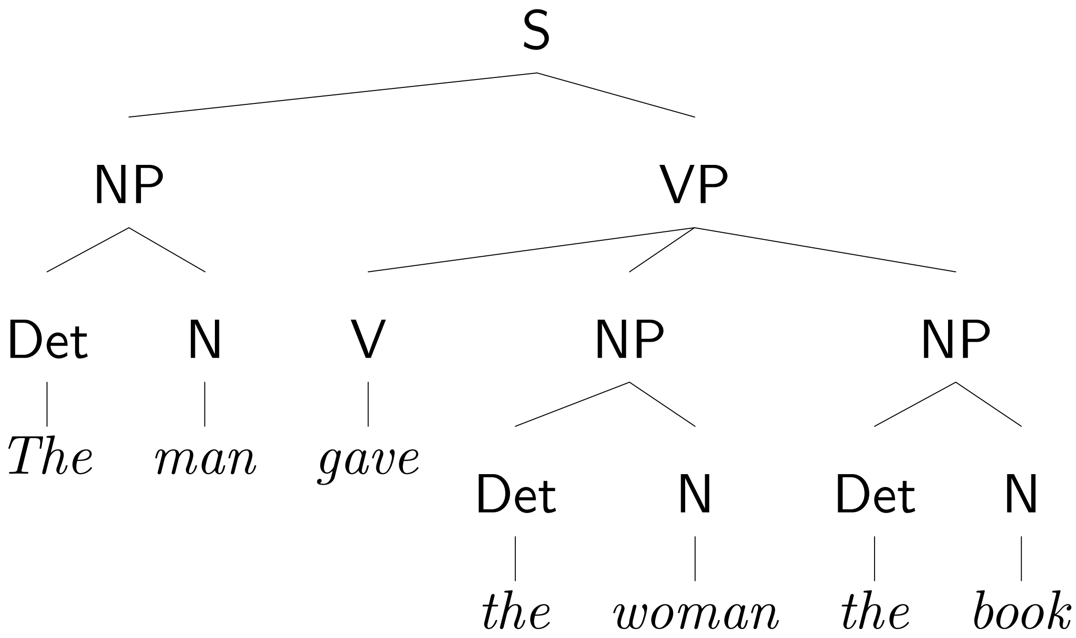
Phrase Structure (I)

- Syntactic constituency is a central notion in generative linguistics.
- Tree structures are used to represent syntactic constituency.
- Phrase structure rules license
  - immediate dominance and
  - immediate precedence

  in local trees.

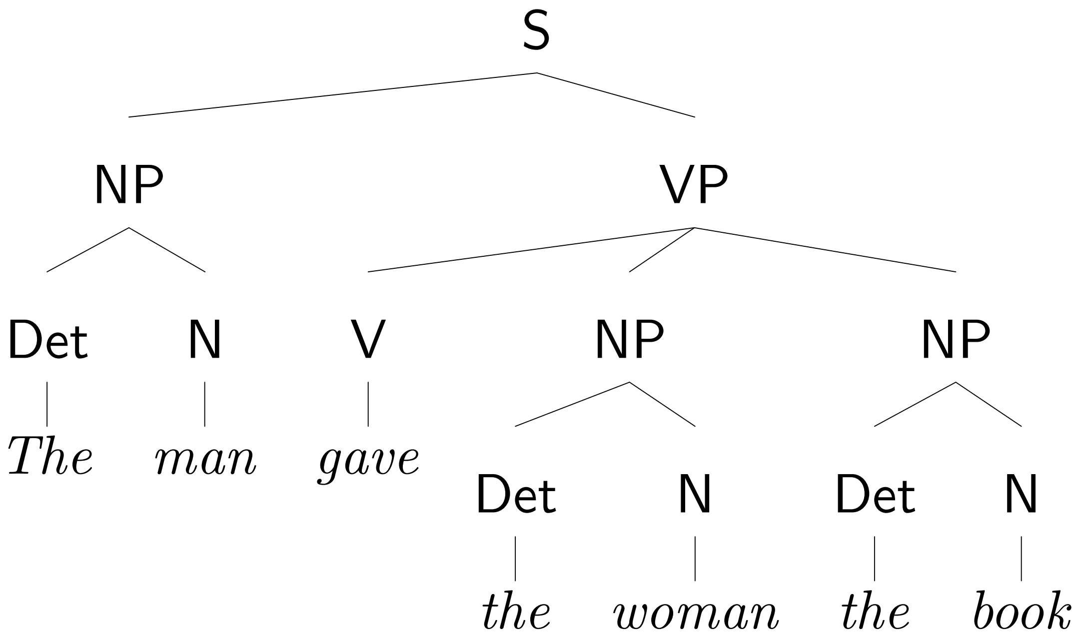
Phrase Structure (II)

S $\rightarrow$ NP, VP
NP $\rightarrow$ Det, N
VP $\rightarrow$ V, NP, NP



Phrase Structure (II)

S → NP, VP
NP → Det, N
VP → V, NP, NP



Phrase structure grammars are well-suited for languages with rigid constituent order like English (with additional mechanisms for long-distance dependencies, etc.).

Topology of the German Clause

- Verb Final (VF)

(1) Peter hat erzählt,

daß	er	das Eis		gegessen	hat.
that	he	the	ice cream	eaten	has

Topology of the German Clause

- Verb Final (VF)

(1) Peter hat erzählt,

daß er das Eis gegessen hat.
that he the ice cream eaten has

- Verb Initial (V1) Position

(2) Hat Peter das Eis gegessen?
has Peter the ice cream eaten

Topology of the German Clause

- Verb Final (VF)

(1) Peter hat erzählt,

daß er das Eis gegessen hat.
that he the ice cream eaten has

- Verb Initial (V1) Position

(2) Hat Peter das Eis gegessen?
has Peter the ice cream eaten

- Verb Second (V2) Position

(3) Peter hat das Eis gegessen.
Peter has the ice cream eaten

The Topological Fields Model (cf., eg., Höhle, 1986)

VL:	KOORD	K_L	Comp		Mittelfeld	Verbkompl	Nachfeld
V1:	KOORD	K_L	V_{fin}		Mittelfeld	Verbkompl	Nachfeld
V2:	PARORD KOORD	K_L	Vorfeld	V_{fin}	Mittelfeld	Verbkompl	Nachfeld

The Topological Fields Model (cf., eg., Höhle, 1986)

VL:	KOORD	K_L	Comp		Mittelfeld	Verbkompl	Nachfeld
V1:	KOORD	K_L	V_{fin}		Mittelfeld	Verbkompl	Nachfeld
V2:	PARORD KOORD	K_L	Vorfeld	V_{fin}	Mittelfeld	Verbkompl	Nachfeld

Generalizations:

The Topological Fields Model (cf., eg., Höhle, 1986)

VL:	KOORD	K_L		Comp	Mittelfeld	Verbkompl	Nachfeld
V1:	KOORD	K_L		V_{fin}	Mittelfeld	Verbkompl	Nachfeld
V2:	PARORD KOORD	K_L	Vorfeld	V_{fin}	Mittelfeld	Verbkompl	Nachfeld

Generalizations:

- complementizer and finite verb are distributed complementarily.

The Topological Fields Model (cf., eg., Höhle, 1986)

VL:	KOORD	K_L		Comp	Mittelfeld	Verbkompl	Nachfeld
V1:	KOORD	K_L		V_{fin}	Mittelfeld	Verbkompl	Nachfeld
V2:	PARORD KOORD	K_L	Vorfeld	V_{fin}	Mittelfeld	Verbkompl	Nachfeld

Generalizations:

- complementizer and finite verb are distributed complementarily.
- Comp/V_{fin} is sometimes referred to as the left sentence bracket.

The Topological Fields Model (cf., eg., Höhle, 1986)

VL:	KOORD	K_L		Comp	Mittelfeld	Verbkompl	Nachfeld
V1:	KOORD	K_L		V_{fin}	Mittelfeld	Verbkompl	Nachfeld
V2:	PARORD KOORD	K_L	Vorfeld	V_{fin}	Mittelfeld	Verbkompl	Nachfeld

Generalizations:

- complementizer and finite verb are distributed complementarily.
- Comp/V_{fin} is sometimes referred to as the left sentence bracket.
- Vorfeld contains exactly one constituent.

The Topological Fields Model (cf., eg., Höhle, 1986)

VL:	KOORD	K_L		Comp	Mittelfeld	Verbkompl	Nachfeld
V1:	KOORD	K_L		V_{fin}	Mittelfeld	Verbkompl	Nachfeld
V2:	PARORD KOORD	K_L	Vorfeld	V_{fin}	Mittelfeld	Verbkompl	Nachfeld

Generalizations:

- complementizer and finite verb are distributed complementarily.
- Comp/V_{fin} is sometimes referred to as the left sentence bracket.
- Vorfeld contains exactly one constituent.
- Mittelfeld is freely ordered sequence of zero or more constituents.

The Topological Fields Model (cf., eg., Höhle, 1986)

VL:	KOORD	K_L		Comp	Mittelfeld	Verbkompl	Nachfeld
V1:	KOORD	K_L		V_{fin}	Mittelfeld	Verbkompl	Nachfeld
V2:	PARORD KOORD	K_L	Vorfeld	V_{fin}	Mittelfeld	Verbkompl	Nachfeld

Generalizations:

- complementizer and finite verb are distributed complementarily.
- Comp/V_{fin} is sometimes referred to as the left sentence bracket.
- Vorfeld contains exactly one constituent.
- Mittelfeld is freely ordered sequence of zero or more constituents.
- Verbal complex, also referred to as right sentence bracket, is a strictly ordered sequence of verbs (and verbal particles).

The Topological Fields Model (cf., eg., Höhle, 1986)

VL:	KOORD	K_L		Comp	Mittelfeld	Verbkompl	Nachfeld
V1:	KOORD	K_L		V_{fin}	Mittelfeld	Verbkompl	Nachfeld
V2:	PARORD KOORD	K_L	Vorfeld	V_{fin}	Mittelfeld	Verbkompl	Nachfeld

Generalizations:

- complementizer and finite verb are distributed complementarily.
- Comp/V_{fin} is sometimes referred to as the left sentence bracket.
- Vorfeld contains exactly one constituent.
- Mittelfeld is freely ordered sequence of zero or more constituents.
- Verbal complex, also referred to as right sentence bracket, is a strictly ordered sequence of verbs (and verbal particles).
- Nachfeld is freely ordered sequence of zero or more constituents.

The Topological Field Model (II)

Not all fields need to be occupied:

(4) Der Mann gibt der Frau das Buch, die er kennt.

Der Mann	gibt	der Frau das Buch,	die er kennt.
VF	V_{fin}	MF	NF

The Topological Field Model (II)

Not all fields need to be occupied:

(4) $\underbrace{\text{Der Mann}}_{\text{VF}}\ \underbrace{\text{gibt}}_{\text{V}_{fin}}\ \underbrace{\text{der Frau das Buch,}}_{\text{MF}}\ \underbrace{\text{die er kennt}}_{\text{NF}}.$

The status test can be used to distinguish Mittelfeld and Nachfeld:

The Topological Field Model (II)

Not all fields need to be occupied:

(4) $\underbrace{\text{Der Mann}}_{\text{VF}}$ $\underbrace{\text{gibt}}_{\text{V}_{fin}}$ $\underbrace{\text{der Frau das Buch,}}_{\text{MF}}$ $\underbrace{\text{die er kennt}}_{\text{NF}}$.

The status test can be used to distinguish Mittelfeld and Nachfeld:

(5) Der Mann hat der Frau das Buch gegeben, die er kennt.
the man has the woman the book given who he knows
'The man gave the book to the woman who he knows.'

The Topological Field Model (II)

Not all fields need to be occupied:

(4) Der Mann gibt der Frau das Buch, die er kennt.

VF	V_{fin}	MF	NF
Der Mann	gibt	der Frau das Buch,	die er kennt.

The status test can be used to distinguish Mittelfeld and Nachfeld:

(5) Der Mann hat der Frau das Buch gegeben, die er kennt.
the man has the woman the book given who he knows
'The man gave the book to the woman who he knows.'

(6) * Der Mann hat der Frau das Buch, die er kennt, gegeben.
the man has the woman the book who he knows given

The Topological Field Model: A Complex Example (I)

A sentence can be embedded inside the Vorfeld, Mittelfeld, or Nachfeld (Reis, 1980, p. 82).

(7) Die Pflanzen, die in ihrem Zimmer standen, waren schon so lange nicht mehr
the plants which in her room stood were already so long no more
gegossen worden, daß es einem bei ihrem Anblick ganz schwer ums Herz
watered been that it one at their viewing wholly heavy around the heart
wurde.
became
‘The plants in her room had long not been watered so that it was a sad sight.’

The Topological Field Model: A Complex Example (II)

Vorfeld	V$_{fin}$	Mittelfeld	Verbkompl	NF
die Pflanzen S$_2$	waren	schon so lange nicht mehr	gegossen worden	S$_3$

The Topological Field Model: A Complex Example (II)

Vorfeld	V_{fin}	Mittelfeld	Verbkompl	NF
die Pflanzen S_2	waren	schon so lange nicht mehr	gegossen worden	S_3

S_2:

Comp	Mittelfeld	Verbkompl
die	in ihrem Zimmer	standen

The Topological Field Model: A Complex Example (II)

Vorfeld	V_{fin}	Mittelfeld	Verbkompl	NF
die Pflanzen S_2	waren	schon so lange nicht mehr	gegossen worden	S_3

S_2:

Comp	Mittelfeld	Verbkompl
die	in ihrem Zimmer	standen

S_3:

Comp	Mittelfeld	Verbkompl
daß	es einem bei ihrem Anblick ganz schwer ums Herz	wurde

More Examples

Vorfeld	left br.	Mittelfeld	right br.	Nachfeld
Karl	schläft.			
Karl	hat		geschlafen.	
Karl	erkennt	Maria.		
Karl	färbt	den Mantel	um	den Maria kennt.
Karl	hat	Maria	erkannt.	
Karl	hat	Maria als sie aus dem Zug stieg sofort	erkannt.	
Karl	hat	Maria sofort	erkannt	als sie aus dem Zug stieg.
Karl	hat	Maria zu erkennen	behauptet.	
Karl	hat		behauptet	Maria zu erkennen.
	Schläft	Karl?		
	Schlaf!			
	Iß	jetzt dein Eis	auf!	
	Hat	er doch das ganze Eis alleine	gegessen.	
	weil	er das ganze Eis alleine	gegessen hat	ohne mit der Wimper zu zucken.
	weil	er das ganze Eis alleine	essen können will	ohne gestört zu werden.
	wer	das ganze Eis alleine	gegessen hat.	

Outline

- Phrase Structure
- A Topological Model for German
- From Phrase Structure to GPSG and HPSG
  - Licensing Free Constituent Order with Phrase Structure
  - Immediate Dominance and Linear Precedence (GPSG)
  - HPSG: Valence, Obliqueness, and Constituent Order
- German Word Order Phenomena and their Analyses in Standard-HPSG/GPSG
- Discontinuous Constituents
- Parsing
- Syntactically Annotated Corpora and Discontinuity
- Summary

Licensing Free Constituent Order with Phrase Structure

In German all six permutations of the arguments in the Mittelfeld are possible:

Licensing Free Constituent Order with Phrase Structure

In German all six permutations of the arguments in the Mittelfeld are possible:

Gab der Mann der Frau das Buch?

Gab der Mann das Buch der Frau?

Gab das Buch der Mann der Frau?

Gab das Buch der Frau der Mann?

Gab der Frau der Mann das Buch?

Gab der Frau das Buch der Mann?

Licensing Free Constituent Order with Phrase Structure

In German all six permutations of the arguments in the Mittelfeld are possible:

Gab der Mann der Frau das Buch?
Gab der Mann das Buch der Frau?
Gab das Buch der Mann der Frau?
Gab das Buch der Frau der Mann?
Gab der Frau der Mann das Buch?
Gab der Frau das Buch der Mann?

$\text{S} \rightarrow \text{V}, \text{NP}_{\text{NOM}}, \text{NP}_{\text{ACC}}, \text{NP}_{\text{DAT}}$

Licensing Free Constituent Order with Phrase Structure

In German all six permutations of the arguments in the Mittelfeld are possible:

Gab der Mann der Frau das Buch?
Gab der Mann das Buch der Frau?
Gab das Buch der Mann der Frau?
Gab das Buch der Frau der Mann?
Gab der Frau der Mann das Buch?
Gab der Frau das Buch der Mann?

$S \rightarrow V, NP_{\mathrm{NOM}}, NP_{\mathrm{ACC}}, NP_{\mathrm{DAT}}$
$S \rightarrow V, NP_{\mathrm{NOM}}, NP_{\mathrm{DAT}}, NP_{\mathrm{ACC}}$

Licensing Free Constituent Order with Phrase Structure

In German all six permutations of the arguments in the Mittelfeld are possible:

Gab der Mann der Frau das Buch?
Gab der Mann das Buch der Frau?
Gab das Buch der Mann der Frau?
Gab das Buch der Frau der Mann?
Gab der Frau der Mann das Buch?
Gab der Frau das Buch der Mann?

$S \rightarrow V, NP_{NOM}, NP_{ACC}, NP_{DAT}$
$S \rightarrow V, NP_{NOM}, NP_{DAT}, NP_{ACC}$
$S \rightarrow V, NP_{ACC}, NP_{NOM}, NP_{DAT}$

Licensing Free Constituent Order with Phrase Structure

In German all six permutations of the arguments in the Mittelfeld are possible:

Gab der Mann der Frau das Buch?
Gab der Mann das Buch der Frau?
Gab das Buch der Mann der Frau?
Gab das Buch der Frau der Mann?
Gab der Frau der Mann das Buch?
Gab der Frau das Buch der Mann?

$S \rightarrow V, NP_{NOM}, NP_{ACC}, NP_{DAT}$
$S \rightarrow V, NP_{NOM}, NP_{DAT}, NP_{ACC}$
$S \rightarrow V, NP_{ACC}, NP_{NOM}, NP_{DAT}$
$S \rightarrow V, NP_{ACC}, NP_{DAT}, NP_{NOM}$

Licensing Free Constituent Order with Phrase Structure

In German all six permutations of the arguments in the Mittelfeld are possible:

Gab der Mann der Frau das Buch?
Gab der Mann das Buch der Frau?
Gab das Buch der Mann der Frau?
Gab das Buch der Frau der Mann?
Gab der Frau der Mann das Buch?
Gab der Frau das Buch der Mann?

S $\rightarrow$ V, NP_{NOM}, NP_{ACC}, NP_{DAT}
S $\rightarrow$ V, NP_{NOM}, NP_{DAT}, NP_{ACC}
S $\rightarrow$ V, NP_{ACC}, NP_{NOM}, NP_{DAT}
S $\rightarrow$ V, NP_{ACC}, NP_{DAT}, NP_{NOM}
S $\rightarrow$ V, NP_{DAT}, NP_{NOM}, NP_{ACC}

Licensing Free Constituent Order with Phrase Structure

In German all six permutations of the arguments in the Mittelfeld are possible:

Gab der Mann der Frau das Buch?	S $\rightarrow$ V, NP_{NOM}, NP_{ACC}, NP_{DAT}
Gab der Mann das Buch der Frau?	S $\rightarrow$ V, NP_{NOM}, NP_{DAT}, NP_{ACC}
Gab das Buch der Mann der Frau?	S $\rightarrow$ V, NP_{ACC}, NP_{NOM}, NP_{DAT}
Gab das Buch der Frau der Mann?	S $\rightarrow$ V, NP_{ACC}, NP_{DAT}, NP_{NOM}
Gab der Frau der Mann das Buch?	S $\rightarrow$ V, NP_{DAT}, NP_{NOM}, NP_{ACC}
Gab der Frau das Buch der Mann?	S $\rightarrow$ V, NP_{DAT}, NP_{ACC}, NP_{NOM}

Licensing Free Constituent Order with Phrase Structure

In German all six permutations of the arguments in the Mittelfeld are possible:

Gab der Mann der Frau das Buch?	S $\rightarrow$ V, NP_{NOM}, NP_{ACC}, NP_{DAT}
Gab der Mann das Buch der Frau?	S $\rightarrow$ V, NP_{NOM}, NP_{DAT}, NP_{ACC}
Gab das Buch der Mann der Frau?	S $\rightarrow$ V, NP_{ACC}, NP_{NOM}, NP_{DAT}
Gab das Buch der Frau der Mann?	S $\rightarrow$ V, NP_{ACC}, NP_{DAT}, NP_{NOM}
Gab der Frau der Mann das Buch?	S $\rightarrow$ V, NP_{DAT}, NP_{NOM}, NP_{ACC}
Gab der Frau das Buch der Mann?	S $\rightarrow$ V, NP_{DAT}, NP_{ACC}, NP_{NOM}

Note that each of these rules expresses the same dominance relation.

Separating Linear Precedence

- a missing generalization about:

  $S \rightarrow V, NP_{NOM}, NP_{ACC}, NP_{DAT}$
  $S \rightarrow V, NP_{NOM}, NP_{DAT}, NP_{ACC}$
  $S \rightarrow V, NP_{ACC}, NP_{NOM}, NP_{DAT}$
  $S \rightarrow V, NP_{ACC}, NP_{DAT}, NP_{NOM}$
  $S \rightarrow V, NP_{DAT}, NP_{NOM}, NP_{ACC}$
  $S \rightarrow V, NP_{DAT}, NP_{ACC}, NP_{NOM}$

Separating Linear Precedence

- a missing generalization about:

  $S \rightarrow V, NP_{NOM}, NP_{ACC}, NP_{DAT}$
  $S \rightarrow V, NP_{NOM}, NP_{DAT}, NP_{ACC}$
  $S \rightarrow V, NP_{ACC}, NP_{NOM}, NP_{DAT}$
  $S \rightarrow V, NP_{ACC}, NP_{DAT}, NP_{NOM}$
  $S \rightarrow V, NP_{DAT}, NP_{NOM}, NP_{ACC}$
  $S \rightarrow V, NP_{DAT}, NP_{ACC}, NP_{NOM}$

- separation of immediate dominance and linear precedence (IDLP, Gazdar, Klein, Pullum and Sag, 1985)
  - no order of the daughters in a rule
  - LP constraints on local trees, i.e., trees of depth one

Separating Linear Precedence

- a missing generalization about:

  $S \rightarrow V, NP_{NOM}, NP_{ACC}, NP_{DAT}$
  $S \rightarrow V, NP_{NOM}, NP_{DAT}, NP_{ACC}$
  $S \rightarrow V, NP_{ACC}, NP_{NOM}, NP_{DAT}$
  $S \rightarrow V, NP_{ACC}, NP_{DAT}, NP_{NOM}$
  $S \rightarrow V, NP_{DAT}, NP_{NOM}, NP_{ACC}$
  $S \rightarrow V, NP_{DAT}, NP_{ACC}, NP_{NOM}$

- separation of immediate dominance and linear precedence
  (IDLP, Gazdar, Klein, Pullum and Sag, 1985)

  - no order of the daughters in a rule
  - LP constraints on local trees, i.e., trees of depth one

- instead of six just one rule & no order restriction for the right hand side

  $S \rightarrow V\ NP_{NOM}\ NP_{ACC}\ NP_{DAT}$

Formulating Linearization Restrictions Again

An immediate dominance rule like “S $\rightarrow$ V NP$_{\mathrm{NOM}}$ NP$_{\mathrm{ACC}}$ NP$_{\mathrm{DAT}}$” alone incorrectly permits orders where the verb appears in the middle of the NPs:

(8) * Der Mann der Frau gibt ein Buch.
the man the woman gives a book

Formulating Linearization Restrictions Again

An immediate dominance rule like "S $\rightarrow$ V NP$_{\mathrm{NOM}}$ NP$_{\mathrm{ACC}}$ NP$_{\mathrm{DAT}}$" alone incorrectly permits orders where the verb appears in the middle of the NPs:

(8) * Der Mann der Frau gibt ein Buch.
the man the woman gives a book

- We have to restrict the position of the verb.

Formulating Linearization Restrictions Again

An immediate dominance rule like "S → V NP_{NOM} NP_{ACC} NP_{DAT}" alone incorrectly permits orders where the verb appears in the middle of the NPs:

(8)	*	Der	Mann	der	Frau	gibt	ein	Buch.
		the	man	the	woman	gives	a	book

- We have to restrict the position of the verb.
- (simplified) Linear Precedence (LP) Statements:

  $\text{V}[\textsc{initial}+] < \text{X}$
  $\text{X} < \text{V}[\textsc{initial}-]$

Valence in GPSG/HPSG

- GPSG: lexical entries have numbers to select specific phrase structure rules.
- HPSG: lexical entries contain list with descriptions of arguments

$$\textit{gibt}\ (\textit{gives})\text{: } \textsc{subcat}\ \langle\ \mathrm{NP}[\textit{nom}],\ \mathrm{NP}[\textit{acc}],\ \mathrm{NP}[\textit{dat}]\ \rangle$$

- Arguments are ordered according to an Obliqueness-Hierarchy (Keenan and Comrie, 1977; Pullum, 1977).

The Obliqueness Hierarchy

Keenan and Comrie, 1977; Pullum, 1977; Grewendorf, 1985, 1988; Pollard and Sag, 1987

SUBJECT => DIRECT OBJECT => INDIRECT OBJECT => OBLIQUES => GENITIVES => OBJECTS OF COMPARISON

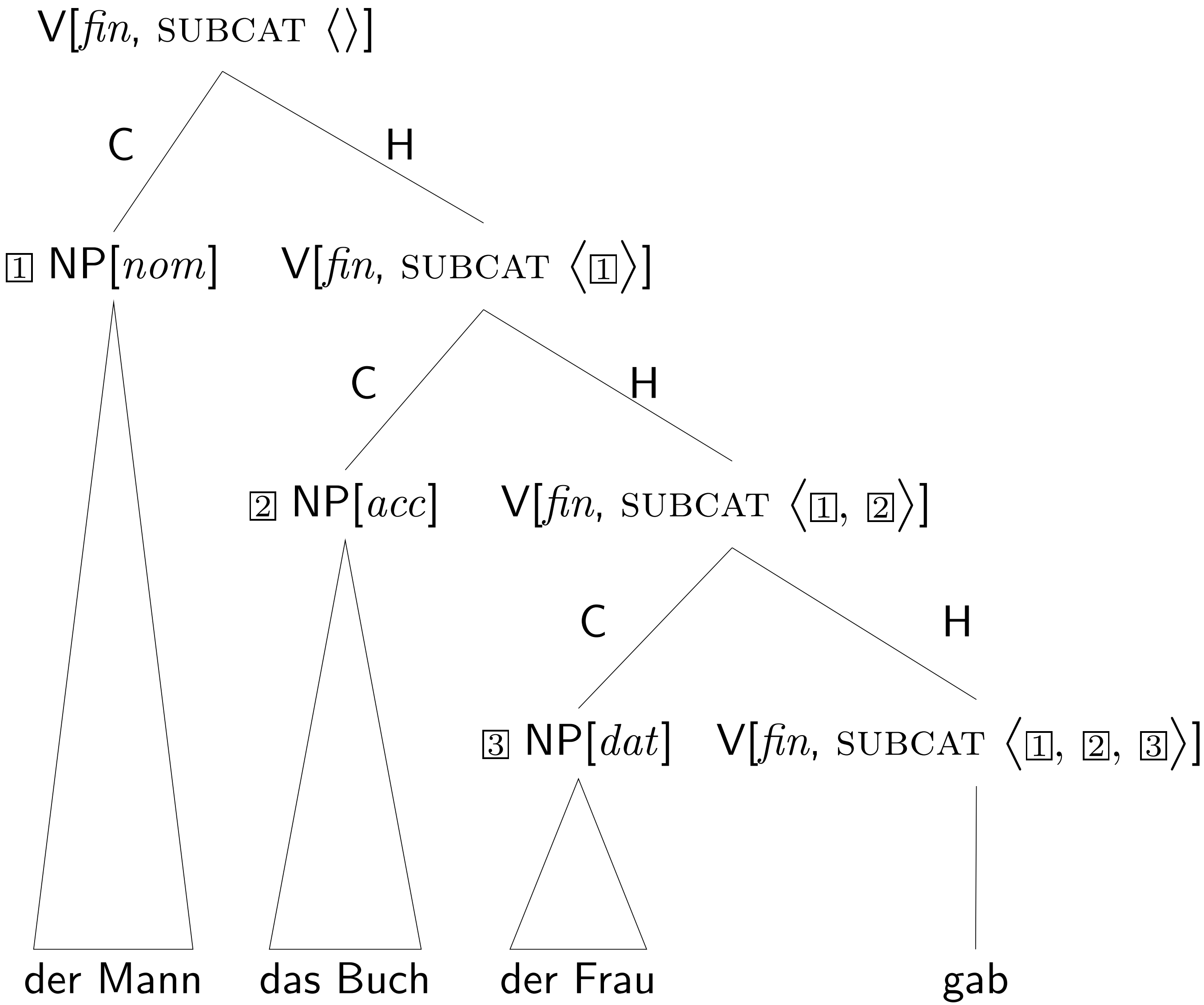
- syntactic activeness of grammatical functions
- higher elements participate easier in syntactic constructions
  - ellipsis (Klein, 1985)
  - topic drop in German (*Vorfeldellipse*) (Fries, 1988),
  - non-matching free relative clauses (Bausewein, 1990; Pittner, 1995; Müller, 1999b),
  - passive (Keenan and Comrie, 1977)
  - depictive secondary predicates (Müller, 2001, 2002)
  - Binding Theory (Grewendorf, 1985; Pollard and Sag, 1994)

Example for a Lexical Entry

gibt ('gives'):

$$\begin{bmatrix} \text{CAT} & \begin{bmatrix} \text{HEAD} & \begin{bmatrix} \text{VFORM} & \textit{fin} \\ \textit{verb} & \end{bmatrix} \\ \text{SUBCAT} & \left\langle \text{NP}[\textit{nom}]_{\boxed{1}},\ \text{NP}[\textit{acc}]_{\boxed{2}},\ \text{NP}[\textit{dat}]_{\boxed{3}} \right\rangle \end{bmatrix} \\ \text{CONT} & \begin{bmatrix} \text{AGENT} & \boxed{1} \\ \text{THEME} & \boxed{2} \\ \text{GOAL} & \boxed{3} \\ \textit{geben} & \end{bmatrix} \\ \textit{loc} & \end{bmatrix}$$

Example: Binary Branching Structures



Constituent Ordering in HPSG

- There is no surface order encoded in the dominance schemata:
  Schema 1 (Head Complement Schema (binary branching))

$$\begin{bmatrix} \text{SYNSEM}|\text{CAT}|\text{SUBCAT}\ \boxed{1} \\ \text{HEAD-DTR} & \begin{bmatrix} \text{SYNSEM}|\text{CAT}|\text{SUBCAT}\ \boxed{1} \oplus \left\langle \boxed{2} \right\rangle \\ \textit{sign} \end{bmatrix} \\ \text{NON-HEAD-DTRS} & \left\langle \boxed{2} \right\rangle \\ \textit{head-complement-structure} \end{bmatrix}$$

- corresponds to head first or complement first serialization:
  [SUBCAT $\boxed{1}$] $\rightarrow$ H[SUBCAT $\boxed{1} \oplus \langle\ \boxed{2}\ \rangle$], $\boxed{2}$
  [SUBCAT $\boxed{1}$] $\rightarrow$ $\boxed{2}$, H[SUBCAT $\boxed{1} \oplus \langle\ \boxed{2}\ \rangle$]

The Constituent Order Principle

- A relational constraint computes the PHON value of the mother:

  Constituent Order Principle adapted from Pollard and Sag, 1987:

$$headed\text{-}structure \rightarrow \begin{bmatrix} \text{PHON} & order\text{-}constituents(\boxed{1}, \boxed{2}) \\ \text{HEAD-DTR} & \boxed{1} \\ \text{NON-HEAD-DTRS} & \boxed{2} \end{bmatrix}$$

- What can be encoded in the *order-constituents* relation?
  - In principle: any relation (incl. shuffling, dropping, adding phonologies)
  - In traditional HPSG: concatenation

A Simple Case: Binary Branching Structures and Concatenation

- head first:

$$\begin{bmatrix} \textsc{phon} & \boxed{1} \oplus \boxed{2} \\ \textsc{head-dtr} & \begin{bmatrix} \textsc{phon} & \boxed{1} \end{bmatrix} \\ \textsc{non-head-dtrs} & \left\langle \begin{bmatrix} \textsc{phon} & \boxed{2} \end{bmatrix} \right\rangle \\ \textit{headed-structure} & \end{bmatrix}$$

example:

$$\begin{bmatrix} \textsc{phon} & \langle\ \textit{loves},\ \textit{Karl}\ \rangle \\ \textsc{head-dtr} & \begin{bmatrix} \textsc{phon} & \langle\ \textit{loves}\ \rangle \end{bmatrix} \\ \textsc{non-head-dtrs} & \left\langle \begin{bmatrix} \textsc{phon} & \langle\ \textit{Karl}\ \rangle \end{bmatrix} \right\rangle \\ \textit{headed-structure} & \end{bmatrix}$$

A Simple Case: Binary Branching Structures and Concatenation

- head first:

$$\begin{bmatrix} \text{PHON} & \boxed{1} \oplus \boxed{2} \\ \text{HEAD-DTR} & \begin{bmatrix} \text{PHON } \boxed{1} \end{bmatrix} \\ \text{NON-HEAD-DTRS} & \left\langle \begin{bmatrix} \text{PHON } \boxed{2} \end{bmatrix} \right\rangle \\ \textit{headed-structure} & \end{bmatrix}$$

example:

$$\begin{bmatrix} \text{PHON} & \langle \textit{ loves, Karl } \rangle \\ \text{HEAD-DTR} & \begin{bmatrix} \text{PHON } \langle \textit{ loves } \rangle \end{bmatrix} \\ \text{NON-HEAD-DTRS} & \left\langle \begin{bmatrix} \text{PHON } \langle \textit{ Karl } \rangle \end{bmatrix} \right\rangle \\ \textit{headed-structure} & \end{bmatrix}$$

- head last:

$$\begin{bmatrix} \text{PHON} & \boxed{2} \oplus \boxed{1} \\ \text{HEAD-DTR} & \begin{bmatrix} \text{PHON } \boxed{1} \end{bmatrix} \\ \text{NON-HEAD-DTRS} & \left\langle \begin{bmatrix} \text{PHON } \boxed{2} \end{bmatrix} \right\rangle \\ \textit{headed-structure} & \end{bmatrix}$$

example:

$$\begin{bmatrix} \text{PHON} & \langle \textit{ Karl, sleeps } \rangle \\ \text{HEAD-DTR} & \begin{bmatrix} \text{PHON } \langle \textit{ sleeps } \rangle \end{bmatrix} \\ \text{NON-HEAD-DTRS} & \left\langle \begin{bmatrix} \text{PHON } \langle \textit{ Karl } \rangle \end{bmatrix} \right\rangle \\ \textit{headed-structure} & \end{bmatrix}$$

Linearization Rules in HPSG

- reference to feature values: $[\text{HEAD } \textit{prep}] > [\text{HEAD } \textit{noun}]$
  orders all prepositions to the left of nominal constituents

(9) a. in the bathroom
    b. * the bathroom in

- reference to immediate dominance schema embedding daughters: FILLER $<$ HEAD

Linearization Rules in HPSG

- reference to feature values: $[\textsc{head}\ \mathit{prep}] > [\textsc{head}\ \mathit{noun}]$
  orders all prepositions to the left of nominal constituents

(9) a. in the bathroom
    b. * the bathroom in

- reference to immediate dominance schema embedding daughters: FILLER < HEAD
- reference to both: HEAD[$\textsc{initial}+$] < COMP
  orders all head daughters with the value + for the feature INITIAL to the left of their complements

Linearization Rules in HPSG

- reference to feature values: $[\text{HEAD } \mathit{prep}] > [\text{HEAD } \mathit{noun}]$
  orders all prepositions to the left of nominal constituents

  (9) a. in the bathroom
  b. * the bathroom in

- reference to immediate dominance schema embedding daughters: FILLER $<$ HEAD
- reference to both: HEAD[INITIAL+] $<$ COMP
  orders all head daughters with the value $+$ for the feature INITIAL to the left of their complements
- extension proposed by Uszkoreit (1987): violable, weighted LP rules
  different markedness of orders in (10):

  (10) a. Gab der Mann der Frau das Buch.
  b. Gab der Mann das Buch der Frau.
  . . .

Outline

- Phrase Structure
- A Topological Model for German
- From Phrase Structure to GPSG and HPSG
- German Word Order Phenomena and their Analyses in Standard-HPSG/GPSG
  - Constituent Order in the Mittelfeld
  - The Mittelfeld and the Predicate Complex
  - The Position of the Finite Verb
- Discontinuous Constituents
- Parsing
- Syntactically Annotated Corpora and Discontinuity
- Summary

Relatively Free Constituent Order in the German Clause

How account for the possible orders in main clauses (11)–(16) and in embedded clauses (17)–(22)?

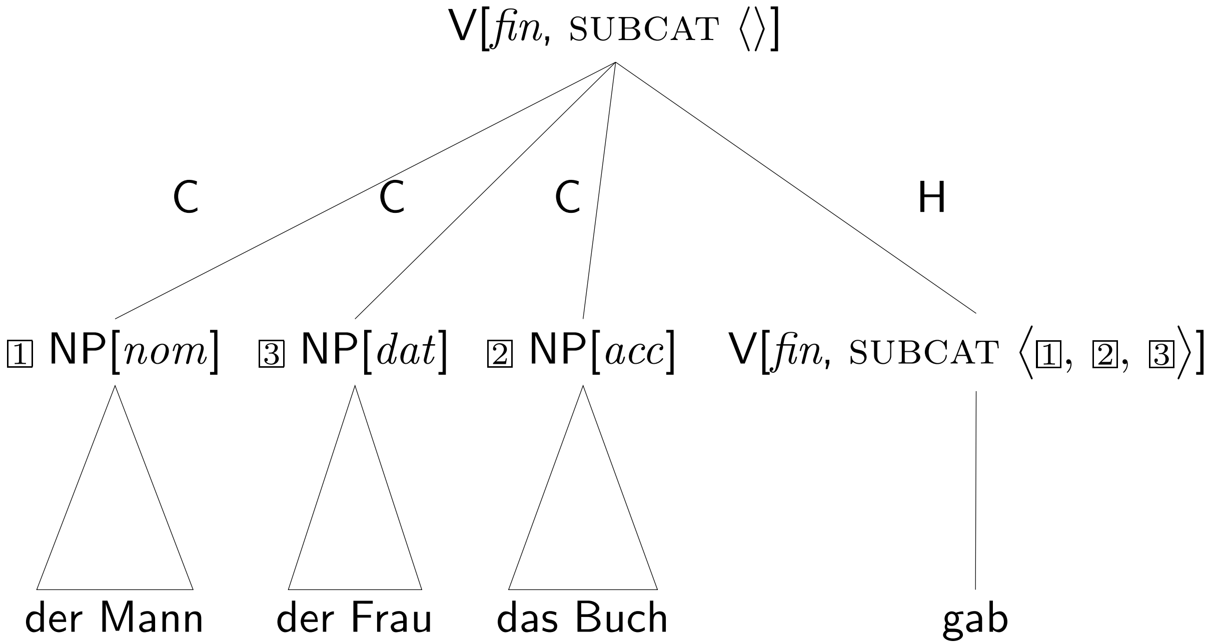
(11) Gab der Mann der Frau das Buch?
(12) Gab der Mann das Buch der Frau?
(13) Gab das Buch der Mann der Frau?
(14) Gab das Buch der Frau der Mann?
(15) Gab der Frau der Mann das Buch?
(16) Gab der Frau das Buch der Mann?

(17) weil der Mann der Frau das Buch gab.
(18) weil der Mann das Buch der Frau gab.
(19) weil das Buch der Mann der Frau gab.
(20) weil das Buch der Frau der Mann gab.
(21) weil der Frau der Mann das Buch gab.
(22) weil der Frau das Buch der Mann gab.

several proposals by Uszkoreit (1987), Pollard (1996),
Reape (1996, 1992, 1994), Kathol (1995, 2000),
Müller (1995, 1999a, 2000a,b, 2002)

Flat Structures

- Pollard (1996) in HPSG (see Uszkoreit, 1987 for GPSG):



- The head contains a description of its arguments ([1], [2], [3]).
- Arguments are daughters of the same node.
- All permutations are possible since elements are sisters in a local tree.

Problems with Flat Structures: Huge Number of Rules

- If one uses a phrase structure based backbone,
  the number of rules is quite big.

- Many different rules are required in a GPSG grammar with flat structures:
  - intransitive verbs like *schlafen* (S $\rightarrow$ V, NP)
  - transitive verbs like *kennen* (S $\rightarrow$ V, NP, NP)
  - ditransitive verbs like *geben* (S $\rightarrow$ V, NP, NP, NP)
  - verbs with four arguments like *kaufen* (S $\rightarrow$ V, NP, NP, PP, PP)
  - verb in initial position: verbal complex at the right periphery of the clause

- Problem also relevant for processing HPSG grammars with a phrase structure backbone.

Problems with Flat Structures: Adjuncts (I)

- Adjuncts can be placed everywhere between the complements:

(23) a. Gab der Mann der Frau das Buch gestern?
b. Gab der Mann der Frau gestern das Buch?
c. Gab der Mann gestern der Frau das Buch?
d. Gab gestern der Mann der Frau das Buch?

- The number of adjuncts is not restricted. → The number of rules is infinite.

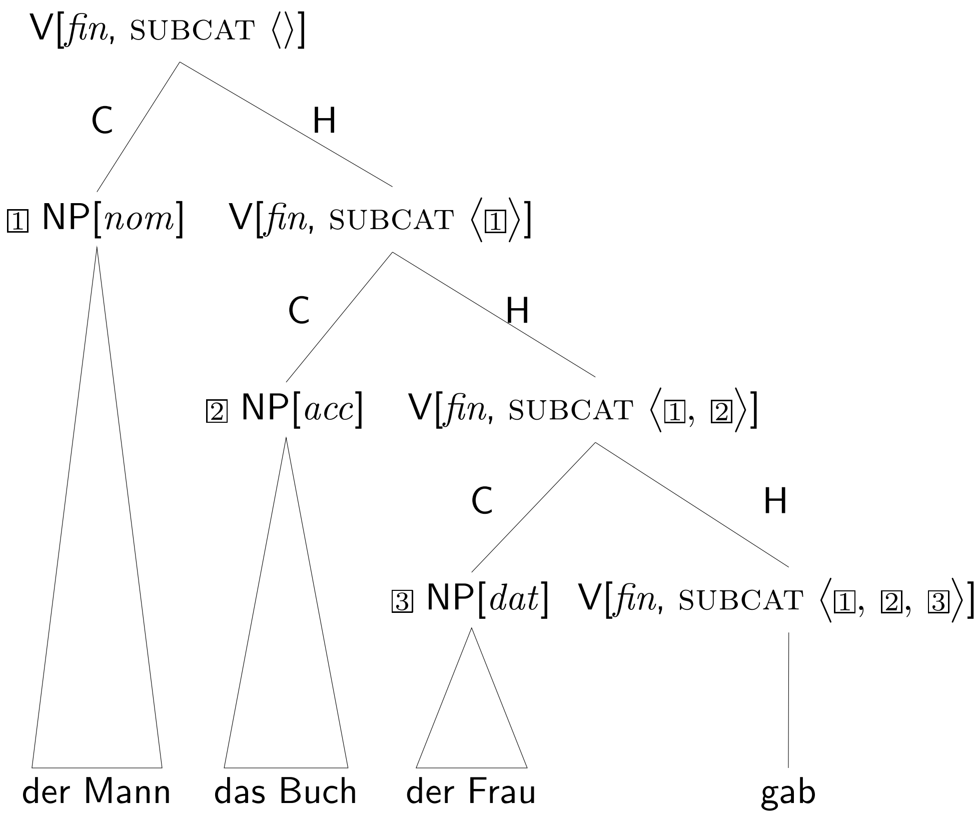
S → V NP NP NP	S → V NP NP NP Adj Adj
S → V NP NP NP Adj	
S → V NP NP Adj NP	. . .
S → V NP Adj NP NP	
S → V Adj NP NP NP	S → V NP NP NP Adj Adj Adj

- Even with ad hoc restrictions huge set of rules.

Problems with Flat Structures: Adjuncts (II)

- Kasper (1994): underspecified number of daughters, adjuncts and complements in the same tree, computation of the meaning by relational constraints (little programs)

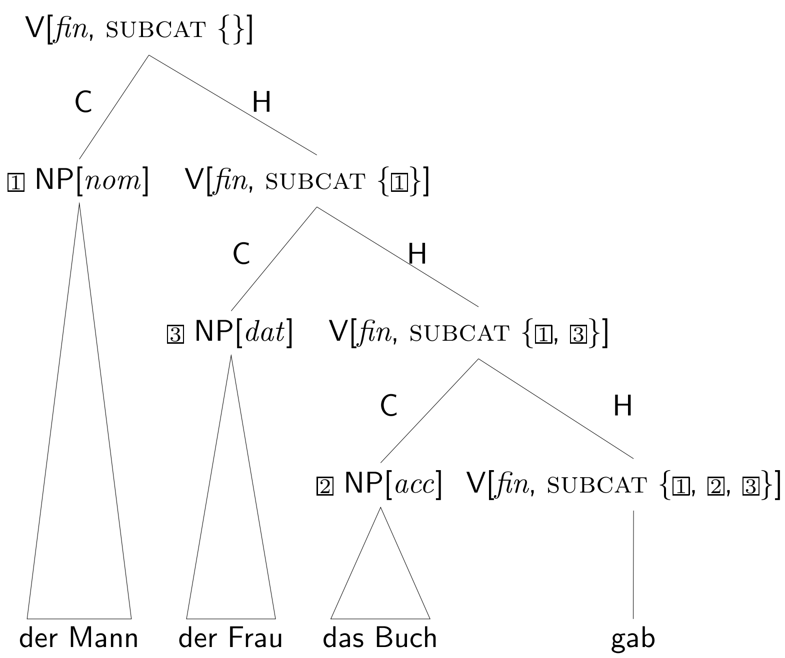
Binary Branching Structures



- trivial to account for the free appearance of adjuncts
- But how do we account for the free ordering of arguments?
  (Usually only the last element from the SUBCAT list can be combined with the head in a head-complement structure.)

A Subcat Set

- Gunji (1986), Hinrichs and Nakazawa (1989), Pollard (1996), Oliva (1992), Engelkamp, Erbach and Uszkoreit (1992), Nerbonne (1994), and Kiss (2001)



- an element of the SUBCAT set is combined with the head
- the only condition is that combined elements are adjacent

Problems of the Subcat Set Approach

- spurious ambiguities if the head is in the middle example: coordinated structures
  Paritong (1992): conjunction takes two arguments

(24) a. [Karl [and Mary]]
b. [[Karl and] Mary]

- spurious ambiguities if nonlocal phenomena are involved:
  Adjacency is not sufficient to determine the position of a trace.

(25) a. Der Frau$_i$ gab der Mann das Buch $\__i$.
the woman$_{dat}$ gave the man$_{nom}$ the book$_{acc}$
'The man gave the woman the book.'
b. Der Frau$_i$ gab der Mann $\__i$ das Buch.
the woman$_{dat}$ gave the man$_{nom}$ the book$_{acc}$
c. Der Frau$_i$ gab $\__i$ der Mann das Buch.
the woman$_{dat}$ gave the man$_{nom}$ the book$_{acc}$

A Subcat List and a Relaxed Subcat Principle

- Baker (1999) and Trale implementation of Müller (To Appear b): relaxation of the SUBCAT principle

- The SUBCAT principle is changed so that any element from the SUBCAT list may be combined with a head, not just the last member of the SUBCAT list.

- the same problems as with the set-based approach

A Lexical Rule

- Uszkoreit (1986): lexical rule that takes a verb and computes lexical items with permuted elements in the SUBCAT list

- at least six lexical items are licensed for a ditransitive verb like *geben*

(26) a. $\langle$ NP[*nom*], NP[*acc*], NP[*dat*] $\rangle$
b. $\langle$ NP[*nom*], NP[*dat*], NP[*acc*] $\rangle$
c. $\langle$ NP[*acc*], NP[*nom*], NP[*dat*] $\rangle$
d. $\langle$ NP[*acc*], NP[*dat*], NP[*nom*] $\rangle$
e. $\langle$ NP[*dat*], NP[*nom*], NP[*acc*] $\rangle$
f. $\langle$ NP[*dat*], NP[*acc*], NP[*nom*] $\rangle$

- Uszkoreit suggested 18 (!) lexical entries.
Instantiation of all features that are relevant for linearization.

- Meurers (1994) and Müller and Kasper (2000) in HPSG implementations.

A Problem for the Lexical Rule Approach

The analysis of sentences like (27) results in spurious ambiguities:

(27) Der Frau gab der Mann das Buch.
the woman$_{dat}$ gave the man$_{nom}$ the book$_{acc}$
'The man gave the woman the book.'

The element in the Vorfeld is extracted. Analysis as a nonlocal dependency.

All three valence patterns in (28) can be used to analyze (27):

(28)
a. $\langle \mathrm{NP}[nom], \mathrm{NP}[acc], \mathrm{NP}[dat] \rangle$
b. $\langle \mathrm{NP}[nom], \mathrm{NP}[dat], \mathrm{NP}[acc] \rangle$
c. $\langle \mathrm{NP}[dat], \mathrm{NP}[nom], \mathrm{NP}[acc] \rangle$

NP[nom] and NP[acc] are in the same order in all lists.
NP[dat] is serialized independently of the other two NPs.

Outline

- Phrase Structure
- A Topological Model for German
- From Phrase Structure to GPSG and HPSG
- German Word Order Phenomena and their Analyses in Standard-HPSG/GPSG
  - Constituent Order in the Mittelfeld
  - The Mittelfeld and the Predicate Complex
  - The Position of the Finite Verb
- Discontinuous Constituents
- Parsing
- Syntactically Annotated Corpora and Discontinuity
- Summary

The Mittelfeld and the Predicate Complex

- Arguments belonging to different heads may be permuted:

weil	es^3	ihm^2	jemand^1	zu lesen^3	versprochen^2	hat^1.	(Haider, 1990)
because	it$_{acc}$	him$_{dat}$	somebody$_{nom}$	to read	promised	has	

  'because somebody promised him to read it.'

- Arguments belonging to different heads may be permuted:

weil	es^3	ihm^2	jemand^1	zu lesen^3	versprochen^2	hat^1.	(Haider, 1990)
because	it$_{acc}$	him$_{dat}$	somebody$_{nom}$	to read	promised	has	

'because somebody promised him to read it.'

The Mittelfeld and the Predicate Complex

- Arguments belonging to different heads may be permuted:

weil	es^3	ihm^2	$jemand^1$	zu $lesen^3$	$versprochen^2$	hat^1.	(Haider, 1990)
because	it_{acc}	him_{dat}	$somebody_{nom}$	to read	promised	has	

'because somebody promised him to read it.'

The Mittelfeld and the Predicate Complex

- Arguments belonging to different heads may be permuted:

weil	es^3	ihm^2	jemand^1	zu lesen^3	versprochen^2	hat^1.	(Haider, 1990)
because	it$_{acc}$	him$_{dat}$	somebody$_{nom}$	to read	promised	has	

'because somebody promised him to read it.'

Data for all six permutations of the arguments can be found, parallel to simplex verbs.

The Mittelfeld and the Predicate Complex

- Arguments belonging to different heads may be permuted:

weil	es^{3}	ihm^{2}	jemand^{1}	zu lesen^{3}	versprochen^{2}	hat^{1}.	(Haider, 1990)
because	it$_{acc}$	him$_{dat}$	somebody$_{nom}$	to read	promised	has	

  ‘because somebody promised him to read it.’

  Data for all six permutations of the arguments can be found, parallel to simplex verbs.

- Adjuncts can appear everywhere between the arguments.

The Mittelfeld and the Predicate Complex

- Arguments belonging to different heads may be permuted:

weil	es^3	ihm^2	jemand^1	zu lesen^3	versprochen^2	hat^1. (Haider, 1990)
because	it$_{acc}$	him$_{dat}$	somebody$_{nom}$	to read	promised	has

  'because somebody promised him to read it.'

  Data for all six permutations of the arguments can be found, parallel to simplex verbs.

- Adjuncts can appear everywhere between the arguments.

- The pattern can be explained by assuming that the predicate complex behaves like a simplex syntactic head.
  The generalizations about the Mittelfeld are the same.

The Analysis of the Predicate Complex (I)

(29) daß er dem Mann [[geholfen haben] wird].
that he the man helped have will
'that he will have helped the man.'

- Hinrichs and Nakazawa (1989) suggest an argument attraction approach:

  Example *haben* (perfect auxiliary):

$$\begin{bmatrix} \text{SUBCAT } \boxed{1} \oplus \left\langle \text{V}\left[\text{SUBCAT } \boxed{1}\right] \right\rangle \\ \textit{cat} \end{bmatrix}$$

The Analysis of the Predicate Complex (I)

(29) daß er dem Mann [[geholfen haben] wird].
that he the man helped have will
'that he will have helped the man.'

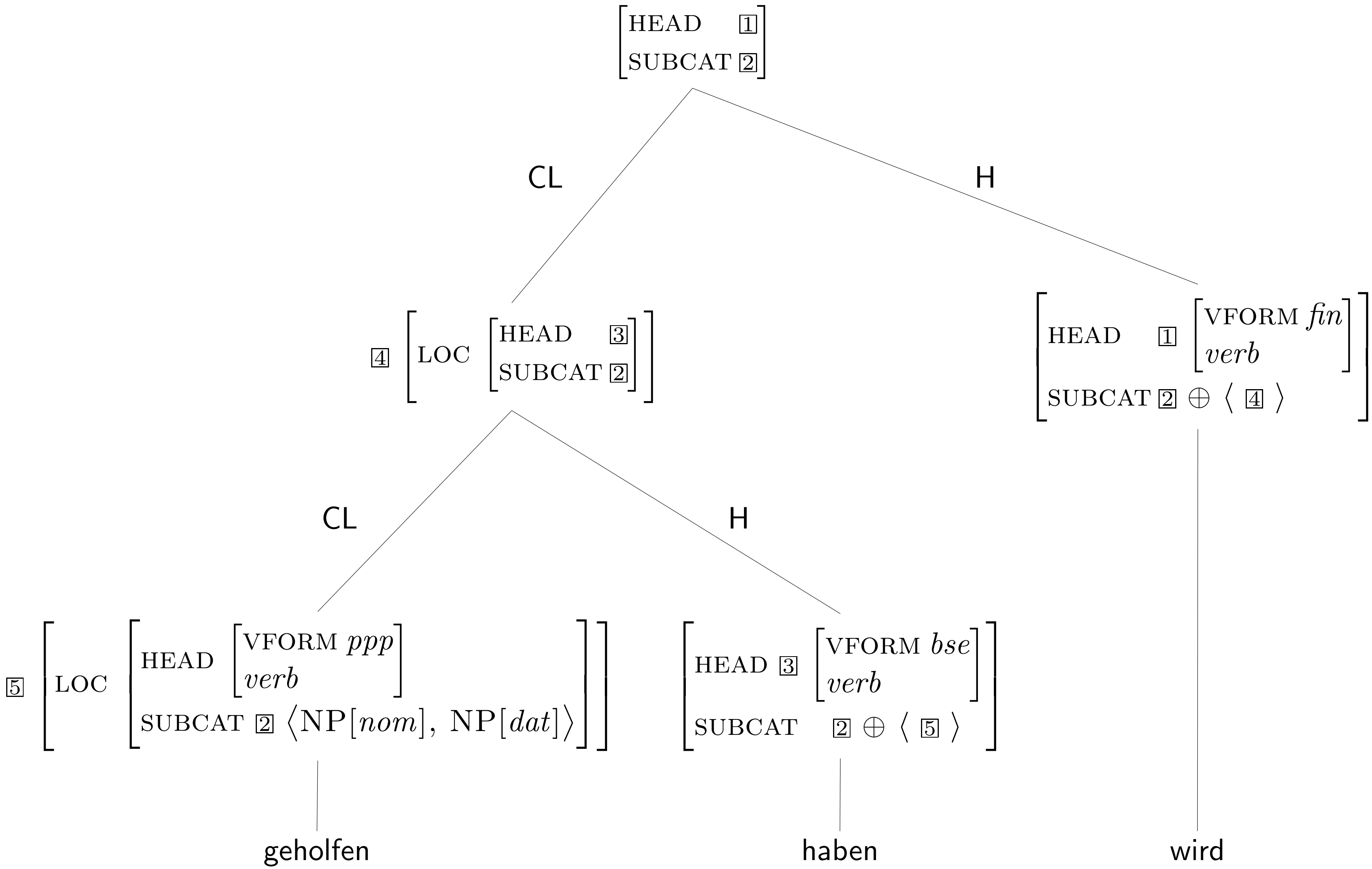
- Hinrichs and Nakazawa (1989) suggest an argument attraction approach:

  Example *haben* (perfect auxiliary):

  $$\begin{bmatrix} \text{SUBCAT } \boxed{1} \oplus \left\langle \text{V}\left[\text{SUBCAT } \boxed{1}\right]\right\rangle \\ \textit{cat} \end{bmatrix}$$

- *geholfen*, *haben*, and *wird* form a complex head

The Analysis of the Predicate Complex (II)



Problems for Processing Sentences with a Predicate Complex (I)

- *haben* is a raising verb. It does not impose constraints on the valence properties of the embedded verbal complex.

(30)

a. weil er getanzt hat. (intransitive)
   because he danced has

b. weil er sie geliebt hat. (transitive)
   because he her loved has

c. weil er es ihr gegeben hat. (ditransitive)
   because he it her given has

Problems for Processing Sentences with a Predicate Complex (I)

- *haben* is a raising verb. It does not impose constraints on the valence properties of the embedded verbal complex.

(30) a. weil er getanzt hat. (intransitive)
because he danced has

b. weil er sie geliebt hat. (transitive)
because he her loved has

c. weil er es ihr gegeben hat. (ditransitive)
because he it her given has

- For all sentences we have a version with the finite verb in initial position:

(31) a. Hat er getanzt?

b. Hat er sie geliebt?

c. Hat er es ihr gegeben?

Problems for Processing Sentences with a Predicate Complex (II)

- The number of arguments of *haben* is unknown until we processed the embedded verbal complex.
- If the verb is in initial position, we do not know anything about the constituents that follow.
- It is not sufficient to know the arguments of the predicate complex in the right sentence bracket since the finite verb also may contribute arguments.
  Example AcI constructions:

(32) a. daß ihn (den Erfolg) uns niemand auskosten ließ. (Höhle)
that it$_{acc}$ the success us$_{acc}$ nobody$_{nom}$ enjoy let
'that nobody let us make the most of it.'

b. Deshalb ließ ihn uns niemand auskosten.
therefor let it us nobody enjoy

Problems for Processing Sentences with a Predicate Complex (II)

- The number of arguments of *haben* is unknown until we processed the embedded verbal complex.
- If the verb is in initial position, we do not know anything about the constituents that follow.
- It is not sufficient to know the arguments of the predicate complex in the right sentence bracket since the finite verb also may contribute arguments.
  Example AcI constructions:

(32) a. daß ihn (den Erfolg) uns niemand auskosten ließ. (Höhle)
that it$_{acc}$ the success us$_{acc}$ nobody$_{nom}$ enjoy let
'that nobody let us make the most of it.'

b. Deshalb ließ ihn uns niemand auskosten.
therefor let it us nobody enjoy

- predicate complex formation may be iterated:

(33) weil Hans Cecilia John das Nilpferd füttern helfen läßt.
because Hans Cecilia John the hippo feed help let
'because Hans lets Cecilia help John feed the hippo.'

→ without ad hoc stipulations a maximal number of arguments of a (complex) head cannot be given
→ independent of parsing direction, we do not know what we do

Problems for Processing Sentences with a Predicate Complex (II)

- The number of arguments of *haben* is unknown until we processed the embedded verbal complex.
- If the verb is in initial position, we do not know anything about the constituents that follow.
- It is not sufficient to know the arguments of the predicate complex in the right sentence bracket since the finite verb also may contribute arguments.
  Example AcI constructions:

(32) a. daß ihn (den Erfolg) uns niemand auskosten ließ. (Höhle)
that it$_{acc}$ the success us$_{acc}$ nobody$_{nom}$ enjoy let
'that nobody let us make the most of it.'

b. Deshalb ließ ihn uns niemand auskosten.
therefor let it us nobody enjoy

- predicate complex formation may be iterated:

(33) weil Hans Cecilia John das Nilpferd füttern helfen läßt.
because Hans Cecilia John the hippo feed help let
'because Hans lets Cecilia help John feed the hippo.'

$\longrightarrow$ without ad hoc stipulations a maximal number of arguments of a (complex) head cannot be given
$\longrightarrow$ independent of parsing direction, we do not know what we do

- Crysmann (2003): The maximal number of arguments that can be added by a verb in initial position is two (AcI construction).

Outline

- Phrase Structure
- A Topological Model for German
- From Phrase Structure to GPSG and HPSG
- German Word Order Phenomena and their Analyses in Standard-HPSG/GPSG
  - Constituent Order in the Mittelfeld
  - The Mittelfeld and the Predicate Complex
  - The Position of the Finite Verb
- Discontinuous Constituents
- Parsing
- Syntactically Annotated Corpora and Discontinuity
- Summary

The Position of the Finite Verb

Two analyses were suggested:

- verb last is the base order, verb first order is derived
  - GB: nowadays standard, goes back to Fourquet (1957), Bierwisch (1963), Reis (1974), Thiersch (1978)

The Position of the Finite Verb

Two analyses were suggested:

- verb last is the base order, verb first order is derived
  - GB: nowadays standard, goes back to Fourquet (1957), Bierwisch (1963), Reis (1974), Thiersch (1978)
  - GPSG: Jacobs (1986)

The Position of the Finite Verb

Two analyses were suggested:

- verb last is the base order, verb first order is derived
  - GB: nowadays standard, goes back to Fourquet (1957), Bierwisch (1963), Reis (1974), Thiersch (1978)
  - GPSG: Jacobs (1986)
  - HPSG: Kiss and Wesche (1991), Oliva (1992), Netter (1992), Kiss (1993), Frank (1994), Kiss (1995), Meurers (2000), Müller and Kasper (2000), Müller (To Appear b)

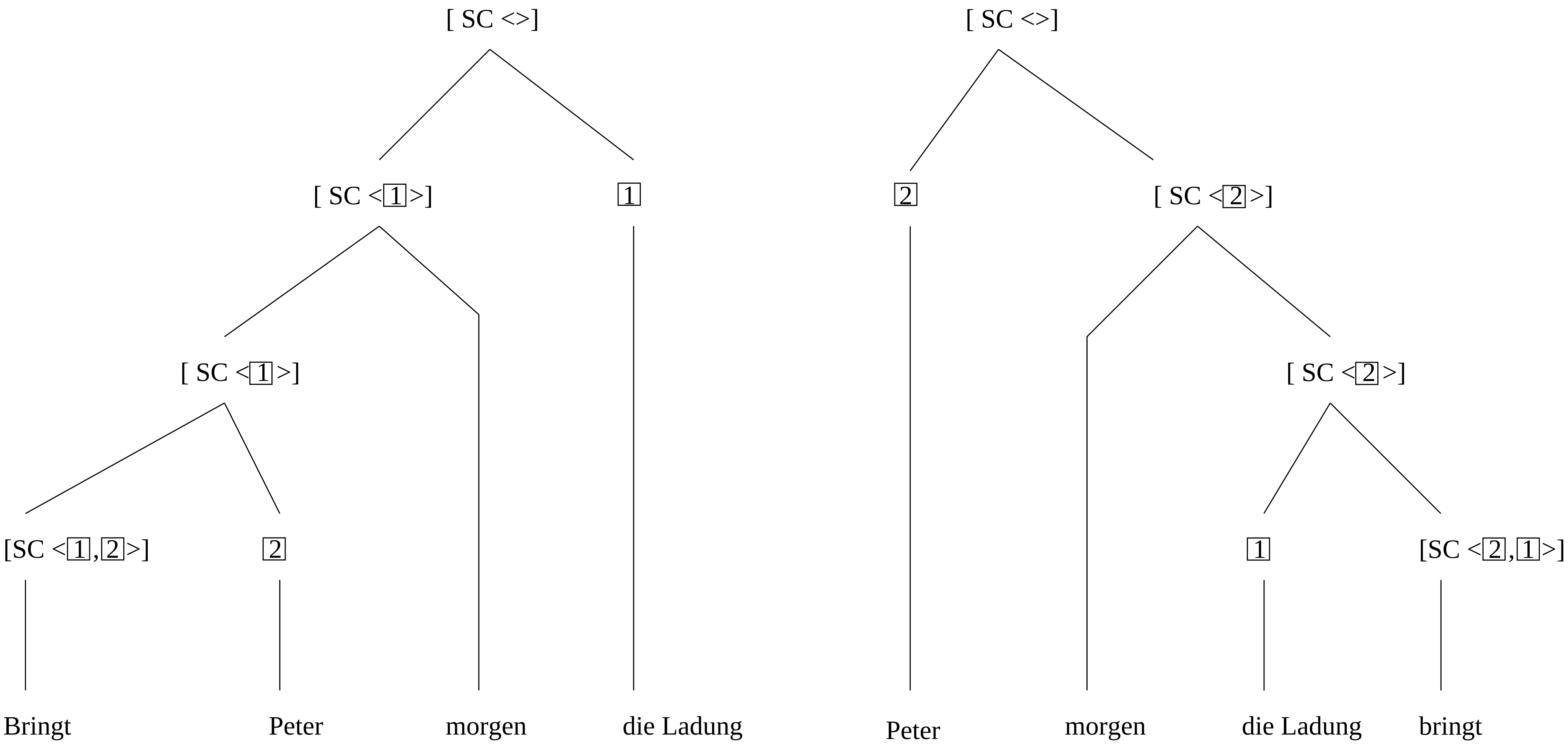
The Position of the Finite Verb

Two analyses were suggested:

- verb last is the base order, verb first order is derived
  - GB: nowadays standard, goes back to Fourquet (1957), Bierwisch (1963), Reis (1974), Thiersch (1978)
  - GPSG: Jacobs (1986)
  - HPSG: Kiss and Wesche (1991), Oliva (1992), Netter (1992), Kiss (1993), Frank (1994), Kiss (1995), Meurers (2000), Müller and Kasper (2000), Müller (To Appear b)

- verb first is one possible position of the verb, just as verb last is Uszkoreit (1987), Pollard (1996)

Verb First and Binary Branching

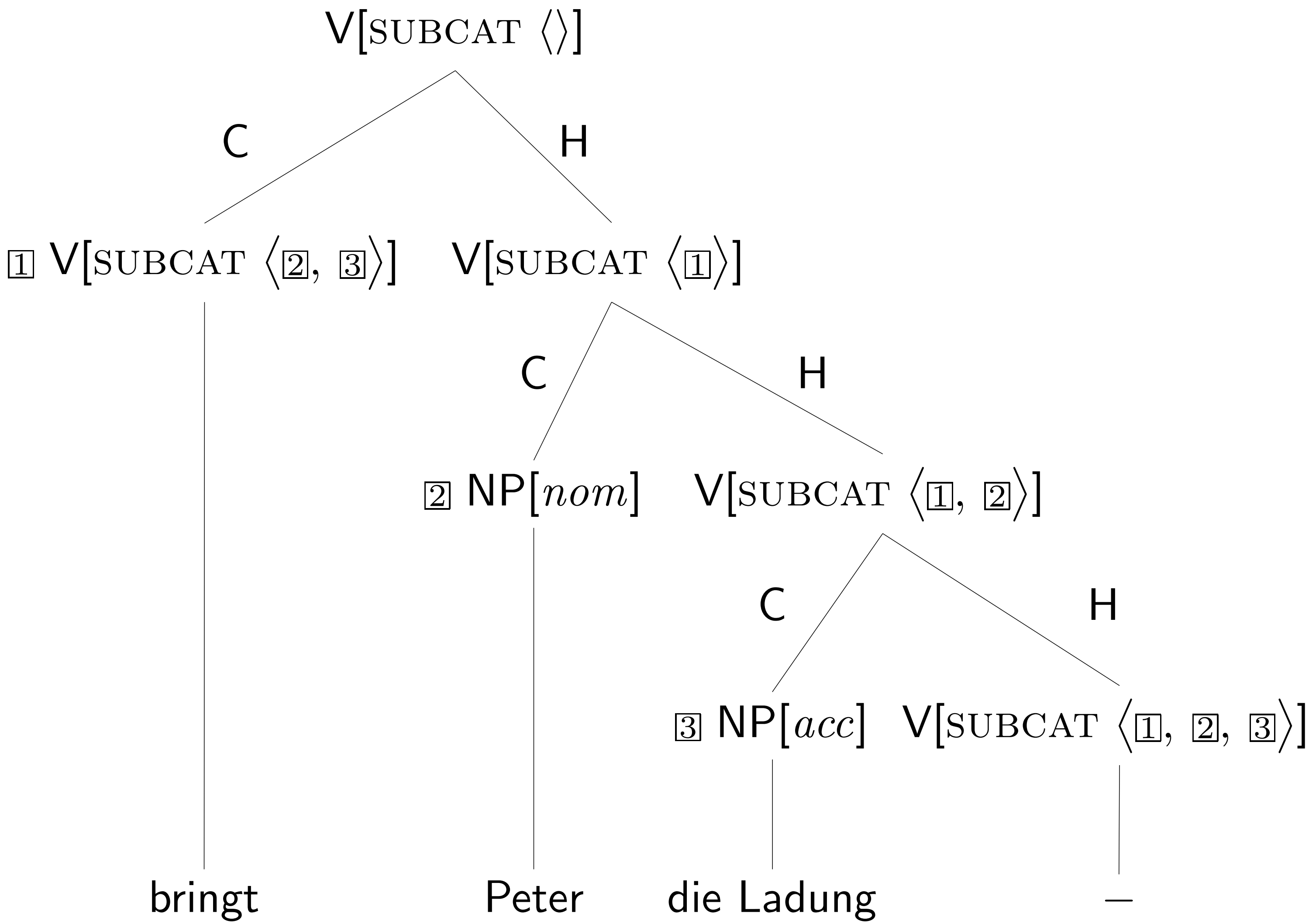


Verb First and Binary Branching

(34) a. weil Peter morgen die Ladung bringt.
b. Bringt Peter morgen die Ladung?

- Netter (1992), Kiss and Wesche (1991), Oliva (1992), Kiss (1995), Frank (1994), Meurers (2000), Müller (To Appear b): empty verbal head in final position

Netter (1992): Analysis with Phonologically Empty Head



Netter's Verbal Trace

- Netter uses argument attraction known from the analysis of the verbal complex:

$$\left[\begin{array}{ll} \text{CAT} & \left[\begin{array}{ll} \text{HEAD} & \textit{verb} \\ \text{SUBCAT} & \left\langle \text{V}\left[\text{SUBCAT } \boxed{1} \qquad\qquad \right] \right\rangle \oplus \boxed{1} \end{array}\right] \\ \\ \textit{loc} & \end{array}\right]$$

- The empty verbal head selects the arguments of the verb in initial position ($\boxed{1}$) + the verb in initial position (V[SUBCAT $\boxed{1}$]).

Netter's Verbal Trace

- Netter uses argument attraction known from the analysis of the verbal complex:

$$\left[\begin{array}{l} \text{CAT} \quad \left[\begin{array}{ll} \text{HEAD} & \textit{verb} \\ \text{SUBCAT} & \left\langle \text{V}[\text{SUBCAT } \boxed{1}, \text{ CONT } \boxed{2}] \right\rangle \oplus \boxed{1} \end{array}\right] \\ \text{CONT } \boxed{2} \\ \textit{loc} \end{array}\right]$$

- The empty verbal head selects the arguments of the verb in initial position ($\boxed{1}$) + the verb in initial position (V[SUBCAT $\boxed{1}$, CONT $\boxed{2}$]).

- The trace has the same meaning as the verb in initial position ($\boxed{2}$).

Problem: Underspecification of Valence Information

- Without further restrictions, the empty element is underspecified.
- Structures that are never used in full analyses are licensed locally.

(35) Die Frau gibt dem Mann das Buch.

1. * die
2. * die, Frau
3. [die Frau]
4. * die, Frau, gibt
5. * [die Frau], gibt
6. * [die Frau gibt]
7. * die, Frau, gibt, dem
8. * [die Frau], gibt, dem
9. * [die Frau gibt], dem
10. * die, Frau, gibt, dem, Mann
11. * [die Frau], gibt, dem, Mann
12. * [die Frau gibt], dem, Mann
13. * [die Frau gibt], [dem Mann]
14. * [die Frau], gibt, [dem Mann]
15. * die, Frau, gibt, dem, Mann, das
16. * die, Frau, gibt, dem, Mann, das
17. * [die Frau], gibt, dem, Mann, das
18. * [die Frau gibt], dem, Mann, das
19. * [die Frau gibt], [dem Mann], das
20. * [die Frau], gibt, [dem Mann], das
21. * die, Frau, gibt, dem, Mann, das, Buch
22. * [die Frau], gibt, dem, Mann, das, Buch
23. * [die Frau gibt], dem, Mann, das, Buch
24. * [die Frau gibt], [dem Mann], das, Buch
25. * [die Frau], gibt, [dem Mann], das, Buch
26. * die, Frau, gibt, dem, Mann, [das, Buch]

27 * [die Frau], gibt, dem, Mann, [das, Buch]
28 * [die Frau gibt], dem, Mann, [das, Buch]
29 * [die Frau gibt], [dem Mann], [das, Buch]
30 * [die Frau], gibt, [dem Mann], [das, Buch]
31 * Frau
32 * Frau, gibt
33 * Frau, gibt, dem
34 * Frau, gibt, dem, Mann
35 * Frau, gibt, [dem, Mann]
36 * Frau, gibt, dem, Mann, das
37 * Frau, gibt, [dem, Mann], das
38 * Frau, gibt, dem, Mann, das, Buch
39 * Frau, gibt, [dem, Mann], das, Buch
40 * Frau, gibt, dem, Mann, [das, Buch]
41 * Frau, gibt, [dem, Mann], [das, Buch]
42 * gibt
43 * gibt, dem
44 * gibt, dem, Mann
45 * gibt, [dem, Mann]
46 * gibt, dem, Mann, das
47 * gibt, [dem, Mann], das
48 * gibt, dem, Mann, das, Buch
49 * gibt, [dem, Mann], das, Buch
50 * gibt, dem, Mann, [das, Buch]
51 * gibt, [dem, Mann], [das, Buch]
52 * dem
53 * dem, Mann
54 * [dem, Mann]
55 * dem, Mann, das
56 * [dem, Mann], das
57 * dem, Mann, das, Buch
58 * [dem, Mann], das, Buch
59 * dem, Mann, [das, Buch]
60 [dem, Mann], [das, Buch]
61 * Mann
62 * Mann, das
63 * Mann, das, Buch
64 * Mann, [das Buch]
65 * das
66 * das Buch
67 [das Buch]
68 * Buch

In the Verb*mobil* grammar, additional constraints were used to exclude the unwanted structures.

For example: All non-verbal complements are fully saturated.

Nevertheless the search space is huge.

Projections of the Empty Head

In any case we need a lot of projections for cases like:

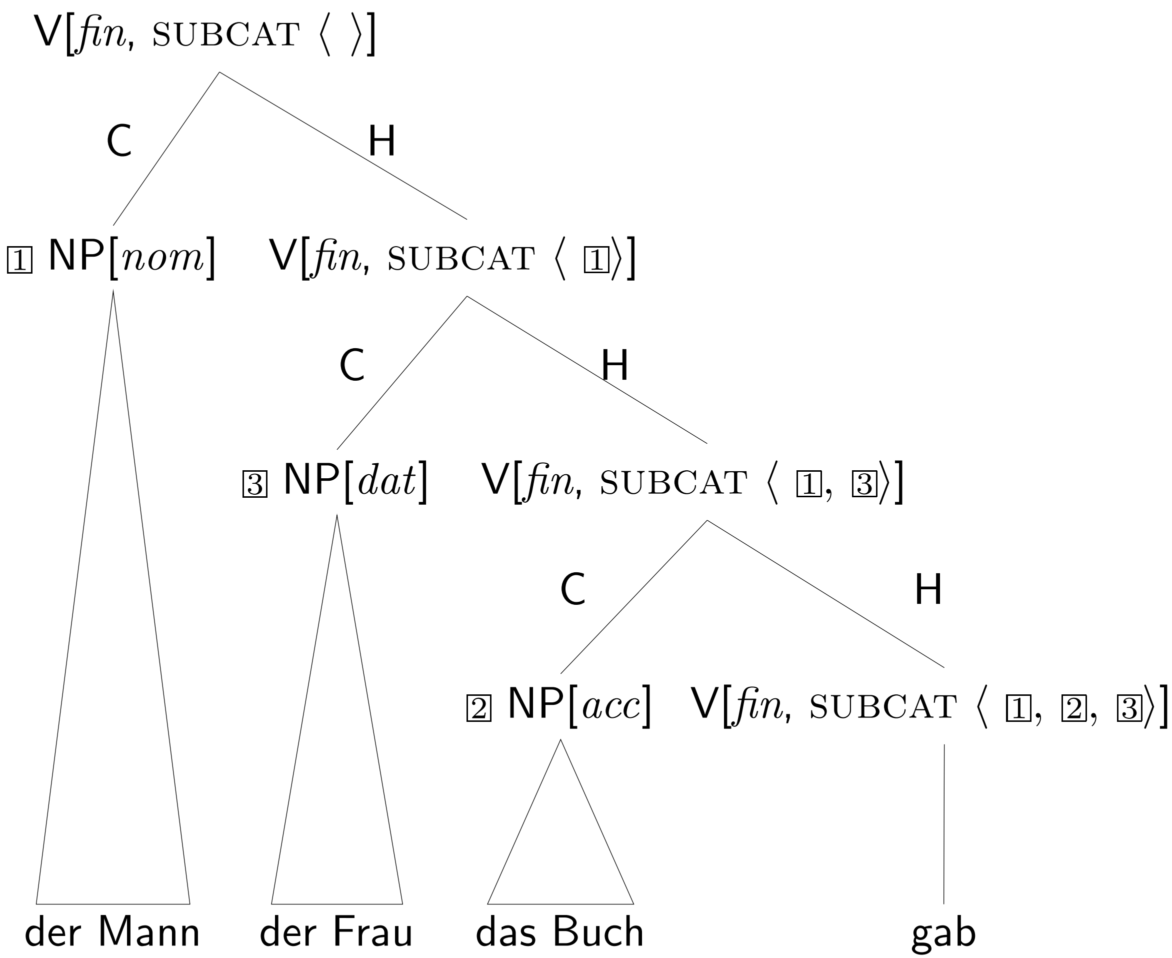
(36) in den nächsten zwei Wochen
in the following two weeks

Various parts of this string can function as full phrases:

(37) a. Er arbeitete zwei Wochen daran.
b. Darüber habe ich Wochen nachgedacht.
c. Ich kenne zwei.
d. Ich kenne den.
e. Ich nehme den nächsten.
f. Ich gehe in den.
g. Ich gehe in den nächsten.
h. Ich arbeite in den nächsten zwei.

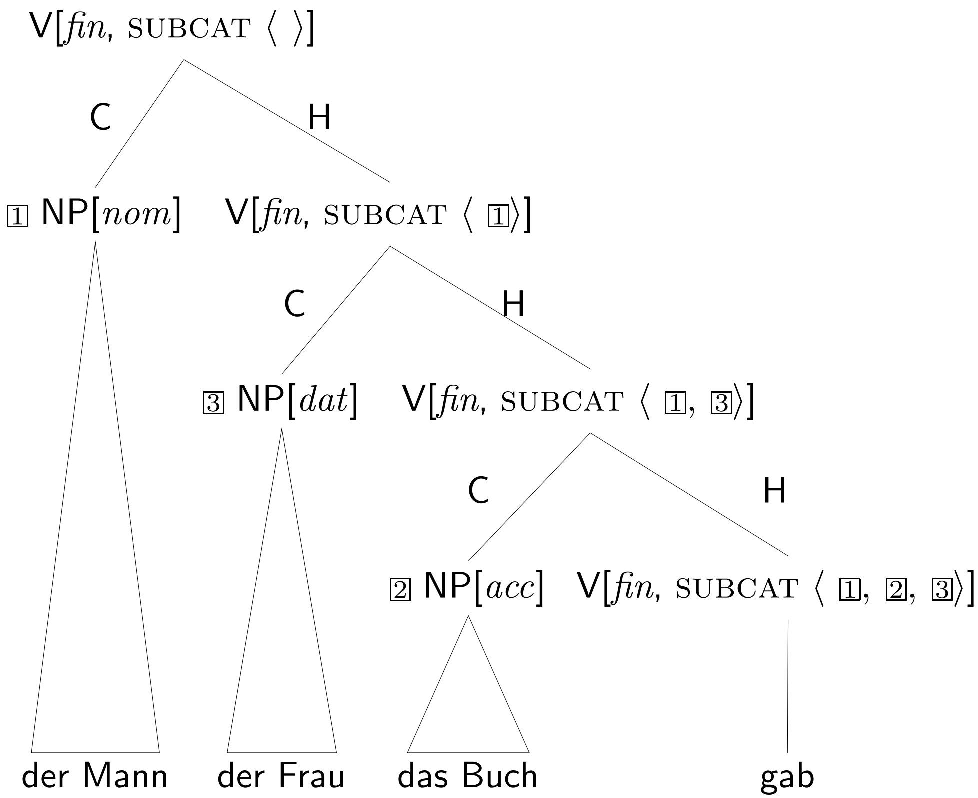
If we do bottom-up parsing, the empty verbal head can be combined with *Wochen* and this projection can be combined with *zwei*, *den nächsten zwei*, or *in den nächsten zwei*. And so on.

Binary Branching and Linearization Rules for the Mittelfeld (I)



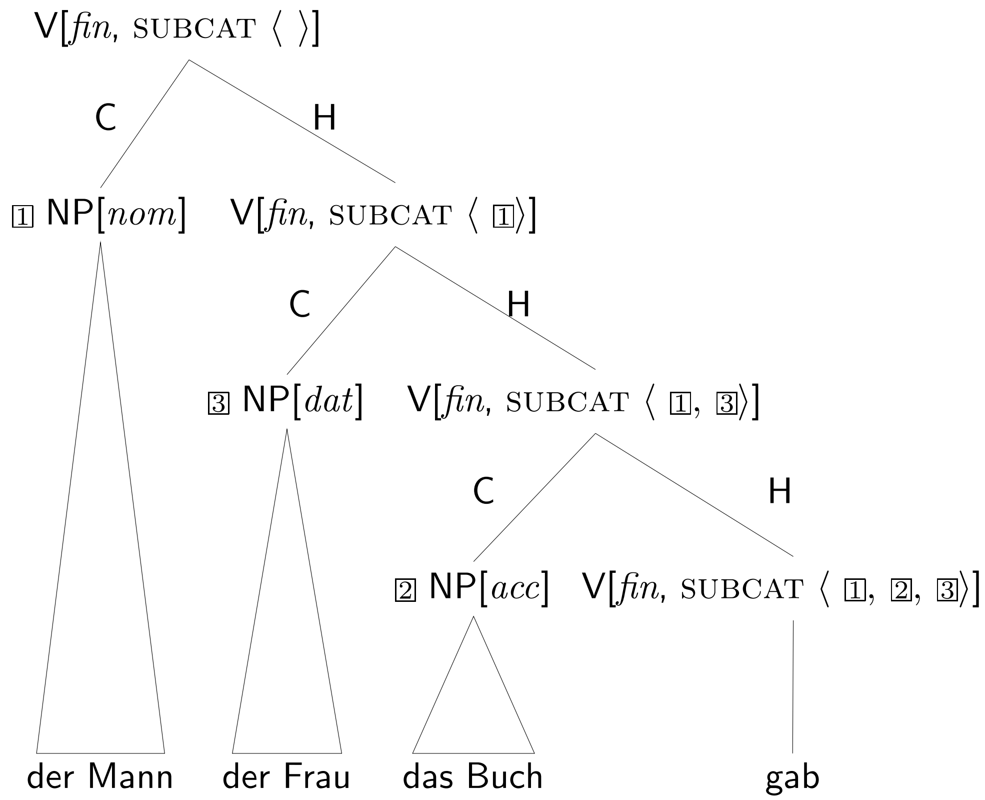
- Note: There is no way to check linearization constraints for the Mittelfeld locally!

Binary Branching and Linearization Rules for the Mittelfeld (II)



Solution one:
All constraints relevant for linearization are encoded in the lexical rules that license different permutations.
This is Uszkoreit's approach. We need at least 18 lexical items for a ditransitive verb.
If we do not assume verb movement as for instance Crysmann (2003),
we need 36 lexical entries for a (finite) ditransitive verb.
For *geben* ('to give') we would get entries for infinitival forms in addition. → The parser would die because of high lexical ambiguity unless there is lazy evaluation of constraints.

Binary Branching and Linearization Rules for the Mittelfeld (III)



Solution two:
We keep a list of elements that are combined with the head and check the LP rules with regard to this list.
This was never done in implemented systems (the Verb*mobil* grammar and successors).
If one introduces such an additional list the performance will drop considerably,
since more space is needed and copying takes longer.

This has to be kept in mind when systems are compared.

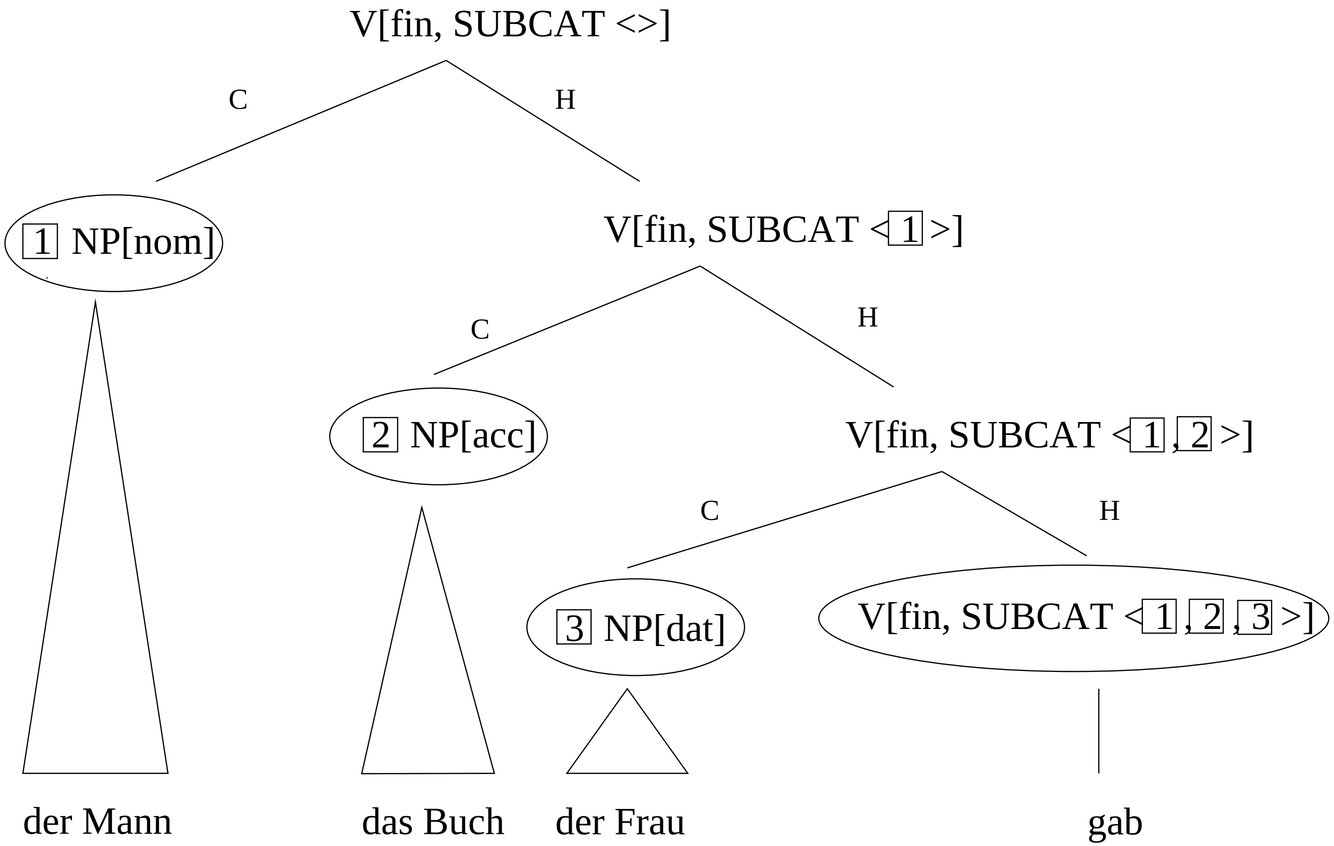
Outline

- Phrase Structure
- A Topological Model for German
- From Phrase Structure to GPSG and HPSG
- German Word Order Phenomena and their Analyses in Standard-HPSG/GPSG
- Discontinuous Constituents
  - Head Projections as Linearization Domains
  - An Extension to Domain Union
- Parsing
- Syntactically Annotated Corpora and Discontinuity
- Summary

Discontinuous Constituents

- extension of the domain in which linearization constraints apply
- computation of phonology values is independent of constituent structure
- German: Reape (1991, 1992, 1994), Pollard, Kasper and Levine (1992, 1994), Kathol and Pollard (1995), Kathol (1995, 2000),
  Müller (1995, 1997, 1999a, 2002), Richter and Sailer (2001)
- Warlpiri: Donohue and Sag (1999)
- Serbo-Croatian: Penn (1999)
- Japanese: Yatabe (2001)
- Dutch: Campbell-Kibler (2001)
- French: Bonami and Godard (To Appear)

Constituent Order Domains and Discontinuous Constituents



- circled nodes get inserted into a list: the linearization domain
- permutation of elements in these domains is restricted only by linearization rules
- linearization domains are head domains
- scrambling is local

Representation of Lexical Heads

$$\begin{bmatrix} \text{PHON} & \boxed{1} \\ \text{SYNSEM} & \boxed{2} \\ \text{DOM} & \left\langle \begin{bmatrix} \text{PHON} & \boxed{1} \\ \text{SYNSEM} & \boxed{2} \\ \text{DOM} & \langle\rangle \\ \multicolumn{2}{l}{\textit{lexical-sign}} \end{bmatrix} \right\rangle \\ \multicolumn{2}{l}{\textit{lexical-sign}} \end{bmatrix}$$

- A lexical head contains a description of itself in its domain.
- Adjunct and complement daughters are inserted into this list and are serialized relative to this element.

Domain Formation

Non-head daughters are inserted into the domain of their head:

$$\textit{headed-structure} \rightarrow \begin{bmatrix} \text{HEAD-DTR}|\text{DOM} & \boxed{1} \\ \text{NON-HEAD-DTRS} & \boxed{2} \\ \text{DOM} & \boxed{1} \bigcirc \boxed{2} \end{bmatrix}$$

The *shuffle* relation holds between three lists A, B, and C, iff C contains all elements of A and B and the order of the elements of A and the order of elements of B is preserved in C.

$$\begin{aligned} \langle a,b \rangle \bigcirc \langle c,d \rangle = & \langle\ a,\ b,\ c,\ d\ \rangle \vee \\ & \langle\ a,\ c,\ b,\ d\ \rangle \vee \\ & \langle\ a,\ c,\ d,\ b\ \rangle \vee \\ & \langle\ c,\ a,\ b,\ d\ \rangle \vee \\ & \langle\ c,\ a,\ d,\ b\ \rangle \vee \\ & \langle\ c,\ d,\ a,\ b\ \rangle \end{aligned}$$

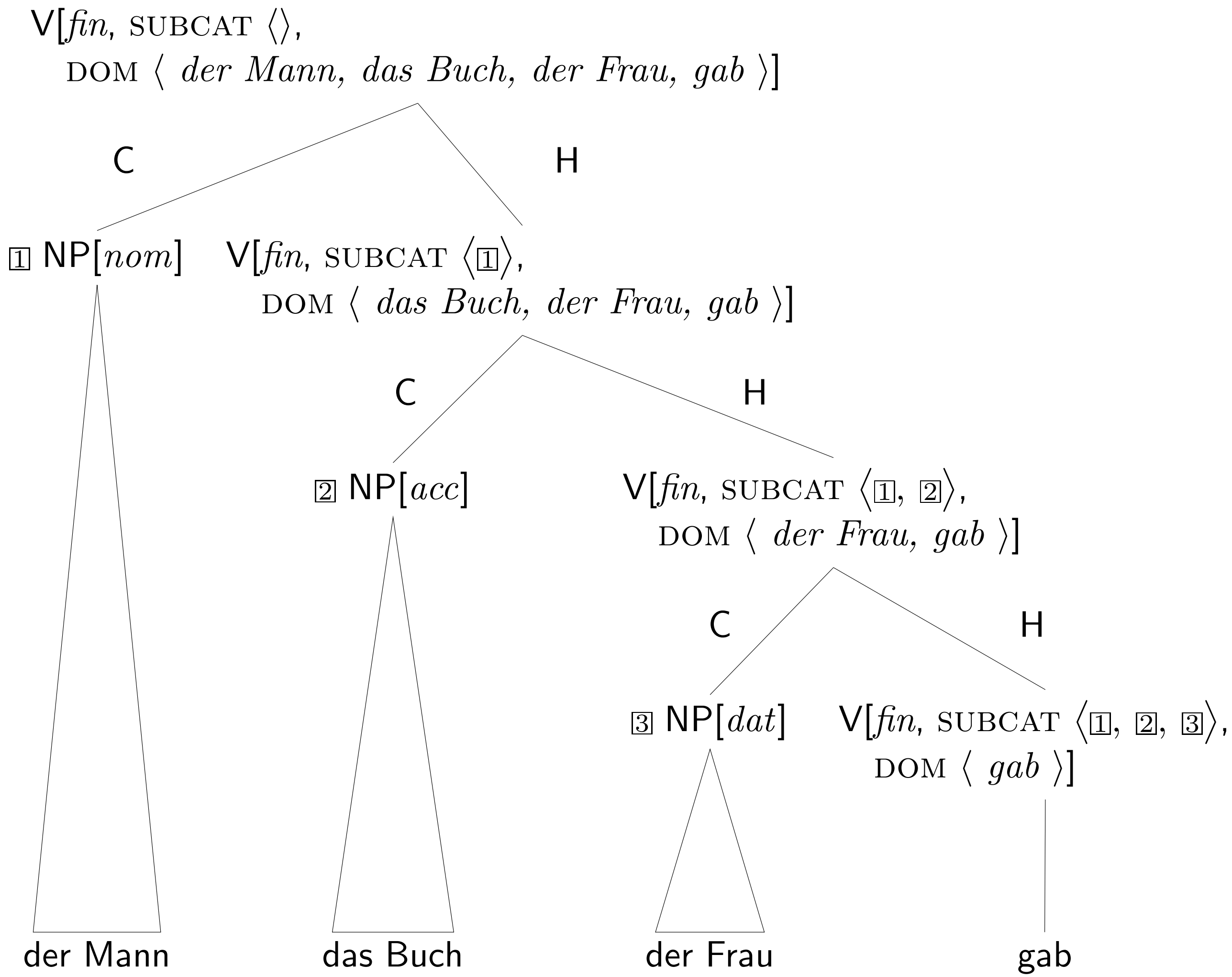
PHON Computation

Elements in DOM are ordered according to their surface order →

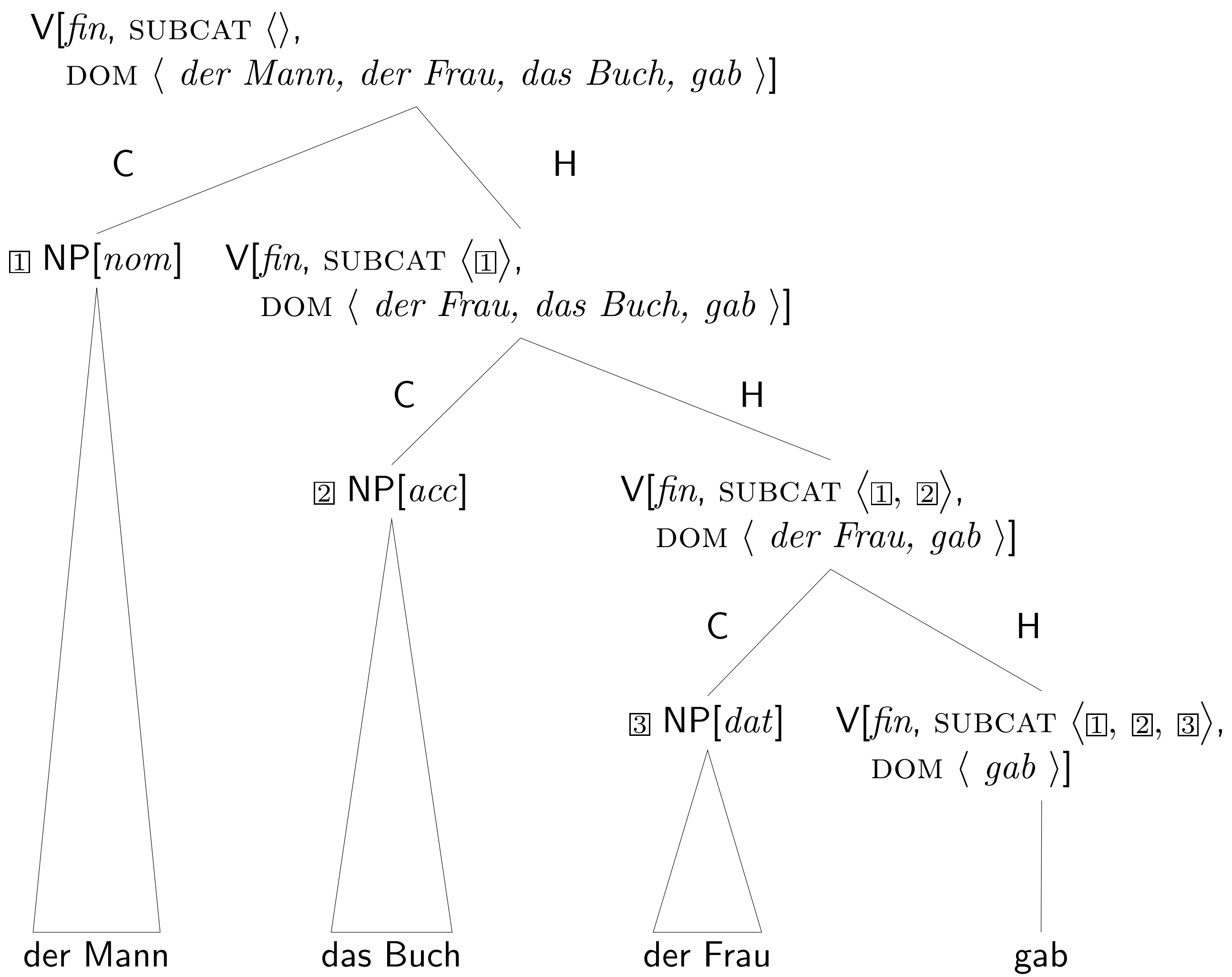
The PHON value of the mother is the concatenation of the PHON values of the domain elements.

$$
\textit{phrasal-sign} \rightarrow \begin{bmatrix} \text{PHON } \boxed{1} \oplus \ \ldots \oplus \boxed{n} \\ \\ \text{DOM } \left\langle \begin{bmatrix} \text{PHON } \boxed{1} \\ \textit{sign} \end{bmatrix}, \ \ldots, \ \begin{bmatrix} \text{PHON } \boxed{n} \\ \textit{sign} \end{bmatrix} \right\rangle \end{bmatrix}
$$

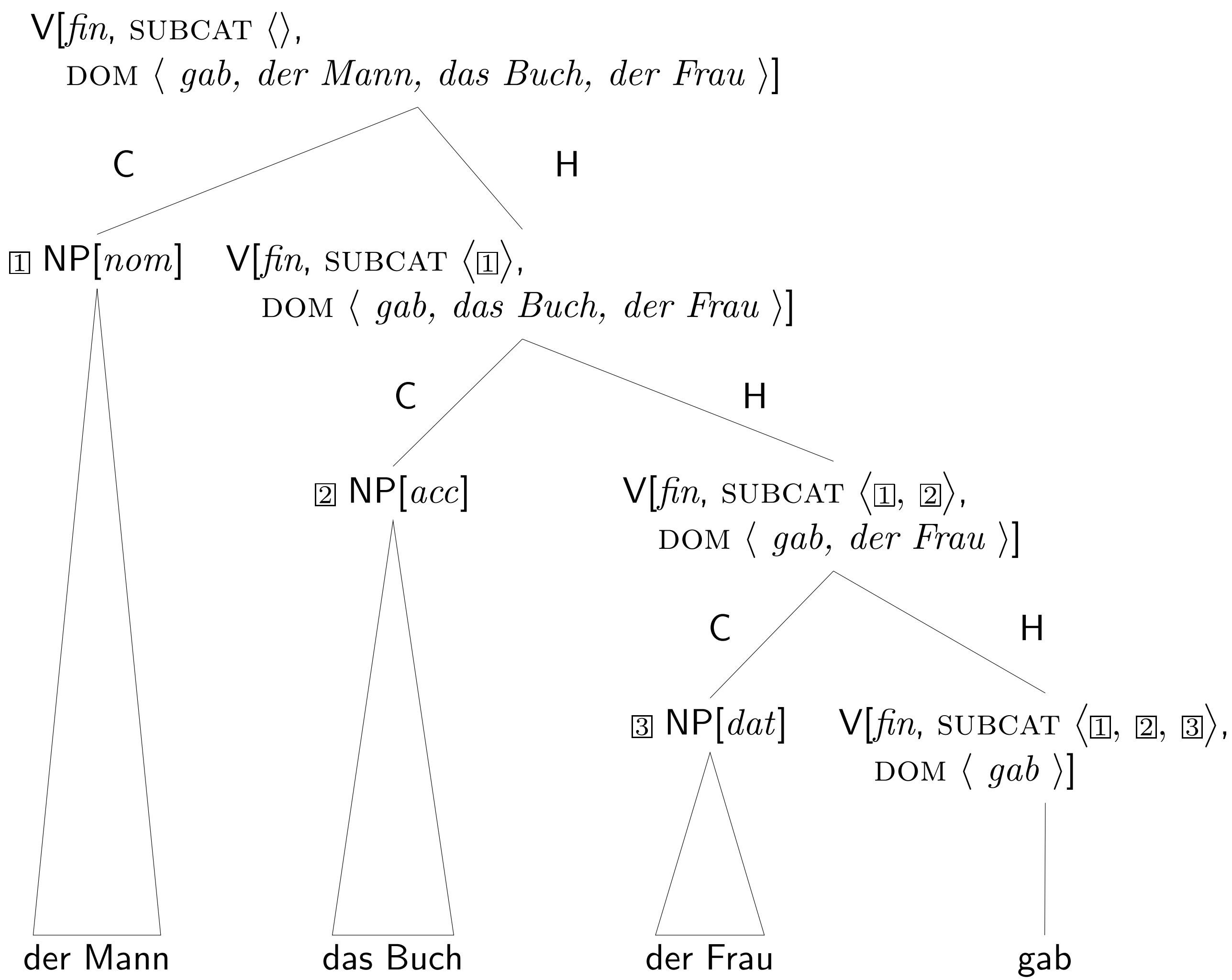
Example: Continuous Constituents



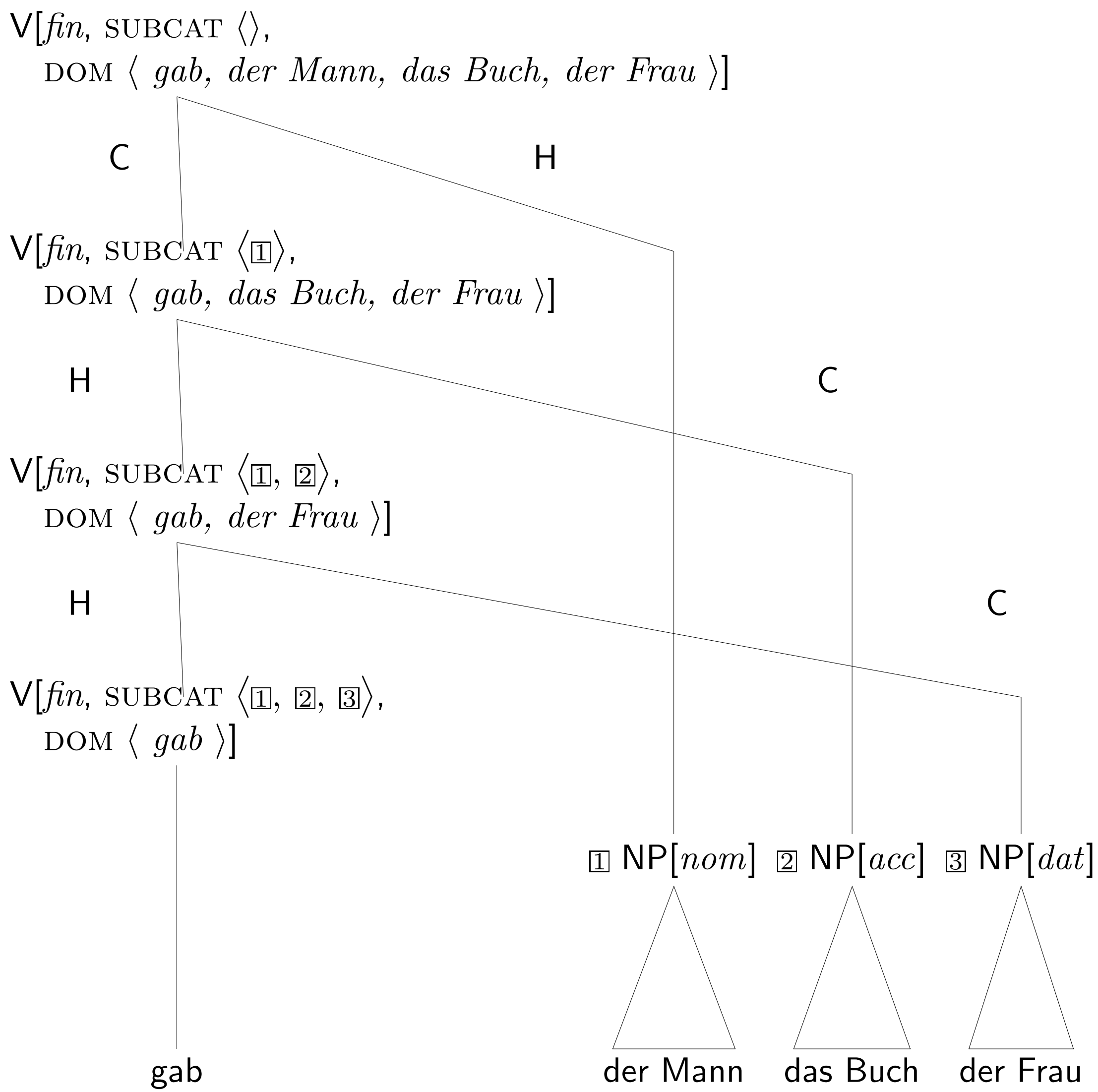
Example: Discontinuous Constituents / Permutation of NPs



Example: Discontinuous Constituents / Verb Placement



Verb Placement with Leaves in Surface Order



A Remark

- The dominance structures for all sentences in (38) are the same:

(38) a. der Mann der Frau das Buch gab.
b. der Mann das Buch der Frau gab.
c. Gab der Mann das Buch der Frau.

- Only the serialization of the elements in the order domains differs.

Discontinuous Constituents and the Predicate Complex

- Both sentences have the same dominance structure:

(39) a. weil es ihm jemand zu lesen versprochen hat.
because it$_{acc}$ him$_{dat}$ somebody$_{nom}$ to read promised has
'because somebody promised him to read it.'

b. Hat es ihm jemand zu lesen versprochen?
has it$_{acc}$ him$_{dat}$ somebody$_{nom}$ to read promised
'Did somebody promise him to read it?'

In both sentences we combine *hat* and *zu lesen versprochen* first.
→ We know what further arguments we are looking for.

What Phenomena Should be Accounted for with Linearization?

- Reape invented order domains to account for scrambling involving arguments of different verbal heads.

What Phenomena Should be Accounted for with Linearization?

- Reape invented order domains to account for scrambling involving arguments of different verbal heads.

- Why did we use argument attraction instead of Domain Union?

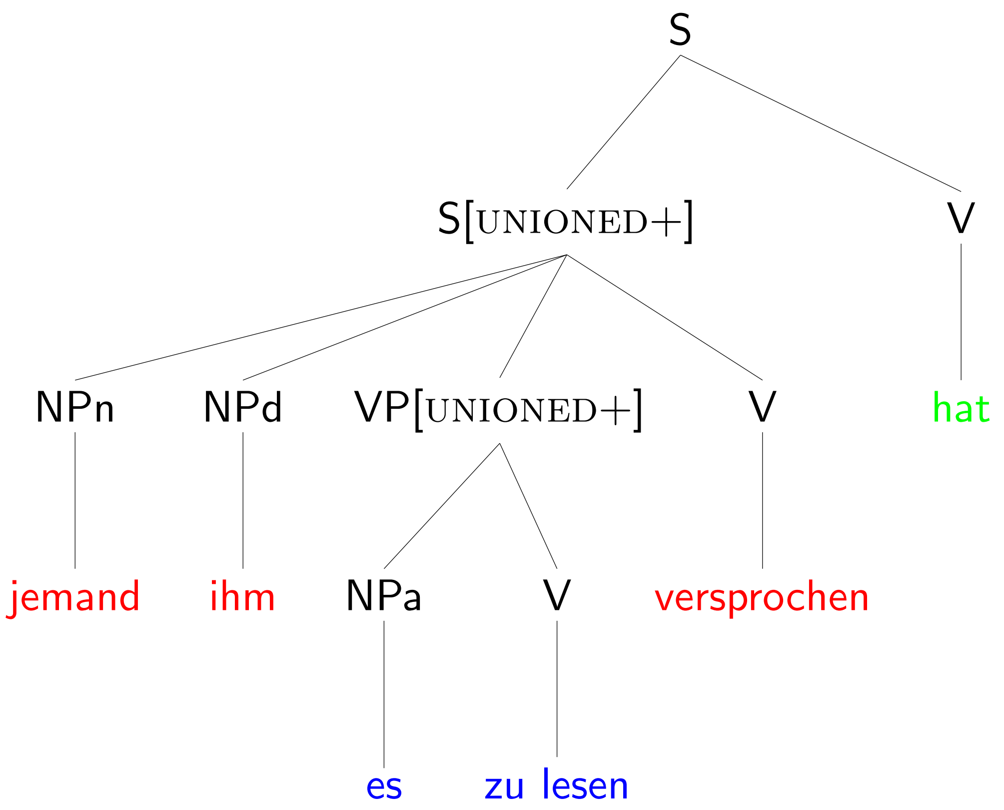
What Phenomena Should be Accounted for with Linearization?

- Reape invented order domains to account for scrambling involving arguments of different verbal heads.

- Why did we use argument attraction instead of Domain Union?

- Reape's Approach: All verbs form (possibly discontinuous) VPs.
  - Arguments and adjuncts of a verb are serialized in the domain of the verb.
  - When a verb is combined with the verbal projection it embeds, the domain elements of the embedded verbal projection are domain-unioned into the domain of the higher verb.

Example: Domain Union



(40) weil es ihm jemand zu lesen versprochen hat.
because it$_{acc}$ him$_{dat}$ somebody$_{nom}$ to read promised has
'because somebody promised him to read it.'

Argument Attraction vs. Domain Union (I)

- Argument Attraction:

  Raising Verbs:
  $\left[\text{SUBCAT}\ \boxed{1} \oplus \left\langle \text{V}\left[\text{SUBCAT}\ \boxed{1}\right]\right\rangle\right]$

Argument Attraction vs. Domain Union (I)

- Argument Attraction:

  Raising Verbs:
  $\big[\textsc{subcat}\ \boxed{1} \oplus \big\langle \mathrm{V}[\textsc{subcat}\ \boxed{1}]\big\rangle\big]$
  Control Verbs (permutation):
  $\big[\textsc{subcat}\ \big\langle \mathrm{NP}_{\boxed{1}}\big\rangle \oplus \boxed{2} \oplus \big\langle \mathrm{V}[\textsc{subcat}\ \langle\ \mathrm{NP}_{\boxed{1}}\ \rangle \oplus \boxed{2}]\big\rangle\big]$

Argument Attraction vs. Domain Union (I)

- Argument Attraction:

  Raising Verbs:
  $\left[\text{SUBCAT } \boxed{1} \oplus \left\langle \text{V}[\text{SUBCAT } \boxed{1}] \right\rangle \right]$
  Control Verbs (permutation):
  $\left[\text{SUBCAT } \left\langle \text{NP}_{\boxed{1}} \right\rangle \oplus \boxed{2} \oplus \left\langle \text{V}[\text{SUBCAT } \langle \text{ NP}_{\boxed{1}} \rangle \oplus \boxed{2}] \right\rangle \right]$
  Control Verbs (no permutation):
  $\left[\text{SUBCAT } \left\langle \text{NP}_{\boxed{1}}, \text{ V}[\text{SUBCAT } \langle \text{ NP}_{\boxed{1}} \rangle] \right\rangle \right]$

Argument Attraction vs. Domain Union (I)

- Argument Attraction:

  Raising Verbs:
  $\left[\text{SUBCAT } \boxed{1} \oplus \left\langle \text{V}[\text{SUBCAT } \boxed{1}] \right\rangle \right]$
  Control Verbs (permutation):
  $\left[\text{SUBCAT } \left\langle \text{NP}_{\boxed{1}} \right\rangle \oplus \boxed{2} \oplus \left\langle \text{V}[\text{SUBCAT } \langle \text{ NP}_{\boxed{1}} \rangle \oplus \boxed{2}] \right\rangle \right]$
  Control Verbs (no permutation):
  $\left[\text{SUBCAT } \left\langle \text{NP}_{\boxed{1}}, \text{ V}[\text{SUBCAT } \langle \text{ NP}_{\boxed{1}} \rangle] \right\rangle \right]$

- Domain Union:

  Raising Verbs:
  $\left[\text{SUBCAT } \left\langle \text{V}[\text{SUBCAT } \langle\rangle, \text{ UNIONED}+] \right\rangle \right]$

Argument Attraction vs. Domain Union (I)

- Argument Attraction:

  Raising Verbs:
  $\left[\text{SUBCAT } \boxed{1} \oplus \left\langle \text{V}[\text{SUBCAT } \boxed{1}] \right\rangle\right]$
  Control Verbs (permutation):
  $\left[\text{SUBCAT } \left\langle \text{NP}_{\boxed{1}} \right\rangle \oplus \boxed{2} \oplus \left\langle \text{V}[\text{SUBCAT } \langle \text{ NP}_{\boxed{1}} \rangle \oplus \boxed{2}] \right\rangle\right]$
  Control Verbs (no permutation):
  $\left[\text{SUBCAT } \left\langle \text{NP}_{\boxed{1}}, \text{ V}[\text{SUBCAT } \langle \text{ NP}_{\boxed{1}} \rangle] \right\rangle\right]$

- Domain Union:

  Raising Verbs:
  $\left[\text{SUBCAT } \left\langle \text{V}[\text{SUBCAT } \langle\rangle, \text{ UNIONED}+] \right\rangle\right]$
  Control Verbs (permutation):
  $\left[\text{SUBCAT } \left\langle \text{NP}_{\boxed{1}}, \text{ V}[\text{SUBCAT } \langle \text{ NP}_{\boxed{1}} \rangle, \text{ UNIONED}+] \right\rangle\right]$

Argument Attraction vs. Domain Union (I)

- Argument Attraction:

  Raising Verbs:
  $\left[\text{SUBCAT } \boxed{1} \oplus \left\langle \text{V}[\text{SUBCAT } \boxed{1}] \right\rangle\right]$
  Control Verbs (permutation):
  $\left[\text{SUBCAT } \left\langle \text{NP}_{\boxed{1}} \right\rangle \oplus \boxed{2} \oplus \left\langle \text{V}[\text{SUBCAT } \langle \text{ NP}_{\boxed{1}} \rangle \oplus \boxed{2}] \right\rangle\right]$
  Control Verbs (no permutation):
  $\left[\text{SUBCAT } \left\langle \text{NP}_{\boxed{1}}, \text{ V}[\text{SUBCAT } \langle \text{ NP}_{\boxed{1}} \rangle] \right\rangle\right]$

- Domain Union:

  Raising Verbs:
  $\left[\text{SUBCAT } \left\langle \text{V}[\text{SUBCAT } \langle\rangle, \text{ UNIONED}+] \right\rangle\right]$
  Control Verbs (permutation):
  $\left[\text{SUBCAT } \left\langle \text{NP}_{\boxed{1}}, \text{ V}[\text{SUBCAT } \langle \text{ NP}_{\boxed{1}} \rangle, \text{ UNIONED}+] \right\rangle\right]$
  Control Verbs (no permutation):
  $\left[\text{SUBCAT } \left\langle \text{NP}_{\boxed{1}}, \text{ V}[\text{SUBCAT } \langle \text{ NP}_{\boxed{1}} \rangle, \text{ UNIONED}-] \right\rangle\right]$

Reape's Domain Union

Reape (1994) divides between functors and arguments and states:
$\left[\text{DTRS}\ \textit{functor-argument-structure}\right] \rightarrow$

$$\begin{bmatrix} \text{DTRS} & \begin{bmatrix} \text{FUN-DTR} & \boxed{1} \\ \text{ARG-DTRS} & \left\langle \boxed{2}\ [\text{UNIONED}-], \ldots, \boxed{i}\ [\text{UNIONED}-] \right\rangle \end{bmatrix} \\ \text{DOM} & \left\langle \boxed{1} \right\rangle \circ \left\langle \boxed{2} \right\rangle \circ \ldots \circ \left\langle \boxed{i} \right\rangle \end{bmatrix}$$

Reape's Domain Union

Reape (1994) divides between functors and arguments and states:
$\left[\text{DTRS } \textit{functor-argument-structure}\right] \rightarrow$

$$\begin{bmatrix} \text{DTRS} & \begin{bmatrix} \text{FUN-DTR} & \boxed{1} \\ \text{ARG-DTRS} & \left\langle \boxed{2} \left[\text{UNIONED}-\right], \ldots, \boxed{i} \left[\text{UNIONED}-\right] \right\rangle \end{bmatrix} \\ \text{DOM} & \left\langle \boxed{1} \right\rangle \circ \left\langle \boxed{2} \right\rangle \circ \ldots \circ \left\langle \boxed{i} \right\rangle \end{bmatrix}$$

- The resulting domain includes the functor ($\boxed{1}$) and all elements that are not domain-unioned ($\boxed{2}$ – $\boxed{i}$).

Reape's Domain Union

Reape (1994) divides between functors and arguments and states:
$\left[\text{DTRS } \textit{functor-argument-structure}\right] \rightarrow$

$$\left[\begin{array}{ll}\text{DTRS} & \left[\begin{array}{ll}\text{FUN-DTR} & \boxed{1} \\ \text{ARG-DTRS} & \left\langle \boxed{2}\ [\text{UNIONED}-], \ldots, \boxed{i}\ [\text{UNIONED}-] \right\rangle \circ \\ & \left\langle [\text{UNIONED}+,\ \text{DOM}\ \boxed{i+1}], \ldots, [\text{UNIONED}+,\ \text{DOM}\ \boxed{n}] \right\rangle \end{array}\right] \\ \text{DOM} & \left\langle \boxed{1} \right\rangle \circ \left\langle \boxed{2} \right\rangle \circ \ldots \circ \left\langle \boxed{i} \right\rangle \circ \boxed{i+1} \circ \ldots \circ \boxed{n} \end{array}\right]$$

- The resulting domain includes the functor ($\boxed{1}$) and all elements that are not domain-unioned ($\boxed{2}$ – $\boxed{i}$).

- The domain elements of all ARG-DTRS that are UNION+ ($\boxed{i+1}$ – $\boxed{n}$) are also shuffled into the resulting domain.

Argument Attraction vs. Domain Union (II)

Raising verbs allow for the embedding of subjectless verbs and of verbs taking a subject:

(41) a. weil heute gearbeitet zu werden scheint.
because today worked to be seems
'There seems to be working today.'

b. weil er zu arbeiten scheint.
because he to work seems
'because he seems to work.'

According to Reape the subject is combined with the embedded verb:

(42) weil er zu arbeiten scheint.
because he to work seems
'because he seems to work.'

scheinen embeds an S.

Argument Attraction vs. Domain Union: Agreement

Kathol (1998): In Reape's proposal subject-verb agreement cannot be established locally:

(43) a.

weil	heute	gearbeitet	zu	werden	scheint.
because	today	worked	to	be	seems

'There seems to be working today.'

b.

Du	scheinst	/ *scheint	mal	wieder	nichts	zu	verstehen.
you	seem-2.sg	/ seems-3.sg	yet	again	nothing	to	understand

'You don't seem to understand anything again.'

- Verbal inflection in subjectless constructions is 3rd sg.
- In (43b) we have 2nd sg, i.e., agreement with the subject.
- In Reape's proposal the subject belongs to the projection of the embedded verb. → problems for accounting for agreement locally.

Argument Attraction vs. Domain Union: Agreement

Kathol (1998): In Reape's proposal subject-verb agreement cannot be established locally:

(43) a. weil heute gearbeitet zu werden scheint.
because today worked to be seems
'There seems to be working today.'

b. Du scheinst / *scheint mal wieder nichts zu verstehen.
you seem-2.sg / seems-3.sg yet again nothing to understand
'You don't seem to understand anything again.'

- Verbal inflection in subjectless constructions is 3rd sg.
- In (43b) we have 2nd sg, i.e., agreement with the subject.
- In Reape's proposal the subject belongs to the projection of the embedded verb. $\rightarrow$ problems for accounting for agreement locally.

(But see Meurers, 2000 for the suggestion to project information about subjects.)

Argument Attraction vs. Domain Union: Coordination

In general approaches that do not assume verbal complexes have problems accounting for coordinations as in (44):

(44) Ich liebte ihn, und ich fühlte, daß er mich auch geliebt hat oder doch,

daß	er	mich	hätte	[lieben	wollen]	oder	[lieben	müssen].
that	he	me	would.have	love	want.to	or	love	must

'I loved him, and I felt that he loved me too, or at least that he would have wanted to or would have had to love me.' (Werner Bergengruen, *Das Tempelchen*. Zürich, 1950, p. 423, quoted from Hoberg, 1981, p. 36)

In Reape's approach *mich lieben* and *wollen* are inserted into the domain of *hätte*. It is unclear how the second conjunct should be treated, how the conjuncts are realized with respect to the coordinating conjunction and so on.

Argument Attraction vs. Domain Union: Coordination

In general approaches that do not assume verbal complexes have problems accounting for coordinations as in (44):

(44) Ich liebte ihn, und ich fühlte, daß er mich auch geliebt hat oder doch,

daß	er	mich	hätte	[lieben	wollen]	oder	[lieben	müssen].
that	he	me	would.have	love	want.to	or	love	must

'I loved him, and I felt that he loved me too, or at least that he would have wanted to or would have had to love me.' (Werner Bergengruen, *Das Tempelchen*. Zürich, 1950, p. 423, quoted from Hoberg, 1981, p. 36)

In Reape's approach *mich lieben* and *wollen* are inserted into the domain of *hätte*. It is unclear how the second conjunct should be treated, how the conjuncts are realized with respect to the coordinating conjunction and so on.

See also Müller, 2002, Chapter 2.3.2.

Argument Attraction vs. Domain Union: Remote Passive

Kathol (1998): Reape's treatment cannot explain remote passives like (45b):

(45) a. weil Mechaniker den Wagen oft zu reparieren versucht haben.
because mechanics$_{nom}$ the car$_{acc}$ often to repair tried have
'because mechanics often tried to repair the car.'

b. weil der Wagen oft zu reparieren versucht wurde.
because the car$_{nom}$ often to repair tried was
'because many attempts were made to repair the car.'

This data can be explained by an analysis of the German passive that suppresses the subject θ role and a case assignment principle that assigns nominative to the first NP with structural case on SUBCAT. The least oblique NP that ist realized as accusative in the active sentence is realized as nominative.

Argument Attraction vs. Domain Union: Remote Passive

Kathol (1998): Reape's treatment cannot explain remote passives like (45b):

(45) a. weil Mechaniker den Wagen oft zu reparieren versucht haben.
because mechanics$_{nom}$ the car$_{acc}$ often to repair tried have
'because mechanics often tried to repair the car.'

b. weil der Wagen oft zu reparieren versucht wurde.
because the car$_{nom}$ often to repair tried was
'because many attempts were made to repair the car.'

This data can be explained by an analysis of the German passive that suppresses the subject θ role and a case assignment principle that assigns nominative to the first NP with structural case on SUBCAT. The least oblique NP that ist realized as accusative in the active sentence is realized as nominative.

If we have a verbal complex that takes the accusative object of the embedded verb ($reparieren$) as its argument, this can be promoted to subject.

Argument Attraction vs. Domain Union: Remote Passive

Kathol (1998): Reape's treatment cannot explain remote passives like (45b):

(45) a. weil Mechaniker den Wagen oft zu reparieren versucht haben.
because mechanics$_{nom}$ the car$_{acc}$ often to repair tried have
'because mechanics often tried to repair the car.'

b. weil der Wagen oft zu reparieren versucht wurde.
because the car$_{nom}$ often to repair tried was
'because many attempts were made to repair the car.'

This data can be explained by an analysis of the German passive that suppresses the subject θ role and a case assignment principle that assigns nominative to the first NP with structural case on SUBCAT. The least oblique NP that ist realized as accusative in the active sentence is realized as nominative.

If we have a verbal complex that takes the accusative object of the embedded verb (*reparieren*) as its argument, this can be promoted to subject. If we embed complete VPs, the nominative in (45b) cannot be explained.

Argument Attraction vs. Domain Union: Dative Passive

Kathol (1998): Reape's analysis excludes analyses of the dative passive in which the auxiliary does the change in valence structure.

(46) a. Der Mann hat den Ball dem Jungen geschenkt.
the man$_{nom}$ has the ball$_{acc}$ the boy$_{dat}$ given
'The man gave the ball to the boy.'

b. Der Junge bekam den Ball geschenkt.
the boy$_{nom}$ got the ball$_{acc}$ given
'The boy got the ball as a present.'

In argument attraction approaches, *bekommen* can attract the arguments of the embedded verb and realize the dative object as its subject.

This is excluded in Reape's approach.

One needs diacritics to distinguish the different lexical items of the various participle forms (perfect, accusative passive, dative passive).

Outline

- Phrase Structure
- A Topological Model for German
- From Phrase Structure to GPSG and HPSG
- German Word Order Phenomena and their Analyses in Standard-HPSG/GPSG
- Discontinuous Constituents
- Parsing
  - With Continuous Constituents
  - With Discontinuous Constituents
- Syntactically Annotated Corpora and Discontinuity
- Summary

CFG Parsing: The Cocke Younger Kasami Algorithm

- Grammar has to be in Chomsky Normal Form (CNF), only
  - RHS with a single terminal: $A \rightarrow a$
  - RHS with two non-terminals: $A \rightarrow BC$
  - no ϵ rules ($A \rightarrow \epsilon$)
- A representation of the string showing positions and word indices:

$$\cdot_0\ w_1\ \cdot_1\ w_2\ \cdot_2\ w_3\ \cdot_3\ w_4\ \cdot_4\ w_5\ \cdot_5\ w_6\ \cdot_6$$

For example: $\cdot_0$ the $\cdot_1$ young $\cdot_2$ boy $\cdot_3$ saw $\cdot_4$ the $\cdot_5$ dragon $\cdot_6$

The Chart

- The well-formed substring table, henceforth (passive) chart, for a string of length n is an $n \times n$ matrix.
- The field (i, j) of the chart encodes the set of all categories of constituents that start at position i and end at position j, i.e.
  chart(i,j) $= \{A \mid A \Rightarrow^* \; w_{i+1} \ldots w_j\}$
- The matrix is triangular since no constituent ends before it starts.

Coverage Represented in the Chart

An input sentence with 6 words:

$$\cdot_0\ w_1\ \cdot_1\ w_2\ \cdot_2\ w_3\ \cdot_3\ w_4\ \cdot_4\ w_5\ \cdot_5\ w_6\ \cdot_6$$

Coverage represented in the chart:

TO:

FROM:	1	2	3	4	5	6
0	0–1	0–2	0–3	0–4	0–5	0–6
1		1–2	1–3	1–4	1–5	1–6
2			2–3	2–4	2–5	2–6
3				3–4	3–5	3–6
4					4–5	4–6
5						5–6

Example for Coverage Represented in Chart

Example sentence:

$$\cdot_0 \text{ the } \cdot_1 \text{ young } \cdot_2 \text{ boy } \cdot_3 \text{ saw } \cdot_4 \text{ the } \cdot_5 \text{ dragon } \cdot_6$$

Coverage represented in chart:

	1	2	3	4	5	6
0	the	the young	the young boy	the young boy saw	the young boy saw the	the young boy saw the dragon
1		young	young boy	young boy saw	young boy saw the	young boy saw the dragon
2			boy	boy saw	boy saw the	boy saw the dragon
3				saw	saw the	saw the dragon
4					the	the dragon
5						dragon

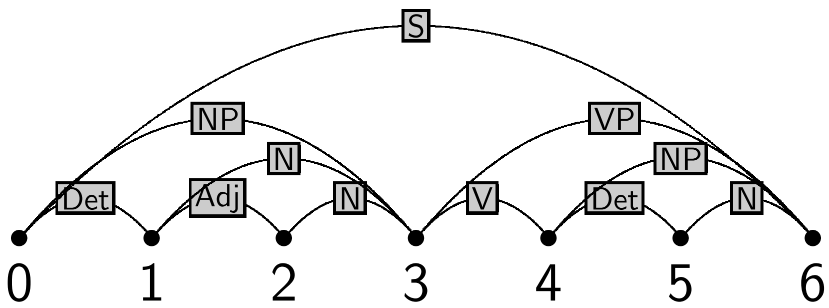
An Example for a Filled-in Chart

Input sentence:

$\cdot_0$ the $\cdot_1$ young $\cdot_2$ boy $\cdot_3$ saw $\cdot_4$ the $\cdot_5$ dragon $\cdot_6$

Chart:

	1	2	3	4	5	6
0	{Det}	{}	{NP}	{}	{}	{S}
1		{Adj}	{N}	{}	{}	{}
2			{N}	{}	{}	{}
3				{V, N}	{}	{VP}
4					{Det}	{NP}
5						{N}



Grammar:

S → NP VP
VP → Vt NP
NP → Det N
N → Adj N
Vt → saw
Det → the
Det → a
N → dragon
N → boy
N → saw
Adj → young

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0						
1						
2						
3						
4						
5						

for j := 1 to length($string$)
    lexical_chart_fill($j-1, j$)
    for $i := j-2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**					
1						
2						
3						
4						
5						

for j := 1 to length($string$)
    lexical_chart_fill($j-1, j$)
    for $i := j-2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**					
1		**2**				
2						
3						
4						
5						

for $j := 1$ to length($string$)
    lexical_chart_fill($j-1, j$)
    for $i := j-2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.

- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**				
1		**2**				
2						
3						
4						
5						

for j := 1 to length($string$)
    lexical_chart_fill($j-1, j$)
    for $i := j-2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**				
1		**2**				
2			**4**			
3						
4						
5						

for $j := 1$ to length($string$)
    lexical_chart_fill($j-1, j$)
    for $i := j-2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**				
1		**2**	**5**			
2			**4**			
3						
4						
5						

for $j := 1$ to length($string$)
    lexical_chart_fill$(j-1, j)$
    for $i := j-2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**	**6**			
1		**2**	**5**			
2			**4**			
3						
4						
5						

for $j := 1$ to length($string$)
    lexical_chart_fill$(j-1, j)$
    for $i := j-2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**	**6**			
1		**2**	**5**			
2			**4**			
3				**7**		
4						
5						

for $j := 1$ to length($string$)
    lexical_chart_fill$(j-1, j)$
    for $i := j-2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**	**6**			
1		**2**	**5**			
2			**4**	**8**		
3				**7**		
4						
5						

for j := 1 to length($string$)
  lexical_chart_fill($j - 1, j$)
  for $i := j - 2$ down to 0
    syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**	**6**			
1		**2**	**5**	**9**		
2			**4**	**8**		
3				**7**		
4						
5						

for j := 1 to length($string$)
  lexical_chart_fill($j-1, j$)
  for i := $j-2$ down to 0
    syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**	**6**	**10**		
1		**2**	**5**	**9**		
2			**4**	**8**		
3				**7**		
4						
5						

for j := 1 to length($string$)
    lexical_chart_fill($j - 1, j$)
    for i := $j - 2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**	**6**	**10**		
1		**2**	**5**	**9**		
2			**4**	**8**		
3				**7**		
4					**11**	
5						

for j := 1 to length($string$)
    lexical_chart_fill($j-1, j$)
    for $i := j-2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**	**6**	**10**		
1		**2**	**5**	**9**		
2			**4**	**8**		
3				**7**	**12**	
4					**11**	
5						

for j := 1 to length($string$)
  lexical_chart_fill($j-1, j$)
  for i := $j-2$ down to 0
    syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**	**6**	**10**		
1		**2**	**5**	**9**		
2			**4**	**8**	**13**	
3				**7**	**12**	
4					**11**	
5						

```
for j := 1 to length(string)
    lexical_chart_fill(j − 1, j)
    for i := j − 2 down to 0
        syntactic_chart_fill(i, j)
```

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**	**6**	**10**		
1		**2**	**5**	**9**	**14**	
2			**4**	**8**	**13**	
3				**7**	**12**	
4					**11**	
5						

for j := 1 to length($string$)
    lexical_chart_fill($j-1, j$)
    for $i := j-2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**	**6**	**10**	**15**	
1		**2**	**5**	**9**	**14**	
2			**4**	**8**	**13**	
3				**7**	**12**	
4					**11**	
5						

for j := 1 to length($string$)
    lexical_chart_fill($j-1, j$)
    for $i := j-2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**	**6**	**10**	**15**	
1		**2**	**5**	**9**	**14**	
2			**4**	**8**	**13**	
3				**7**	**12**	
4					**11**	
5						**16**

for j := 1 to length($string$)
    lexical_chart_fill($j-1, j$)
    for $i := j-2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**	**6**	**10**	**15**	
1		**2**	**5**	**9**	**14**	
2			**4**	**8**	**13**	
3				**7**	**12**	
4					**11**	**17**
5						**16**

for $j := 1$ to length($string$)
    lexical_chart_fill($j-1, j$)
    for $i := j - 2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**	**6**	**10**	**15**	
1		**2**	**5**	**9**	**14**	
2			**4**	**8**	**13**	
3				**7**	**12**	**18**
4					**11**	**17**
5						**16**

for j := 1 to length($string$)
    lexical_chart_fill($j-1, j$)
    for $i := j-2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**	**6**	**10**	**15**	
1		**2**	**5**	**9**	**14**	
2			**4**	**8**	**13**	**19**
3				**7**	**12**	**18**
4					**11**	**17**
5						**16**

for j := 1 to length($string$)
    lexical_chart_fill($j-1, j$)
    for i := $j-2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**	**6**	**10**	**15**	
1		**2**	**5**	**9**	**14**	**20**
2			**4**	**8**	**13**	**19**
3				**7**	**12**	**18**
4					**11**	**17**
5						**16**

for j := 1 to length($string$)
    lexical_chart_fill($j-1, j$)
    for $i := j - 2$ down to 0
        syntactic_chart_fill(i, j)

Filling in the Chart

- It is important to fill in the chart systematically.
- We build all constituents that end at a certain point before we build constituents that end at a later point.

	1	2	3	4	5	6
0	**1**	**3**	**6**	**10**	**15**	**21**
1		**2**	**5**	**9**	**14**	**20**
2			**4**	**8**	**13**	**19**
3				**7**	**12**	**18**
4					**11**	**17**
5						**16**

for $j := 1$ to length($string$)
    lexical_chart_fill$(j-1, j)$
    for $i := j-2$ down to 0
        syntactic_chart_fill(i, j)

lexical_chart_fill(j-1,j)

- Idea: Lexical lookup. Fill the field $(j-1, j)$ in the chart with the preterminal category dominating word j.

- Realized as:

$$chart(j-1, j) := \{\mathsf{X} \mid \mathsf{X} \rightarrow \mathsf{word}_j \in \mathsf{P}\}$$

syntactic_chart_fill(i,j)

- Idea: Perform all reduction step using syntactic rules such that the reduced symbol covers the string from i to j.

- Realized as: $chart(i,j) = \left\{ A \;\middle|\; \begin{array}{l} A \rightarrow BC \in P, \\ i < k < j, \\ B \in chart(i,k), \\ C \in chart(k,j) \end{array} \right\}$

- Explicit loops over every possible value of k and every context free rule:

  $chart(i,j) := \{\}.$
  for $k := i+1$ to $j-1$
  for every $A \rightarrow BC \in P$
  if $B \in chart(i,k)$ and $C \in chart(k,j)$ then
  $chart(i,j) := chart(i,j) \cup \{\mathsf{A}\}.$

The Complete CYK Algorithm

Input: start category S and input $string$

$n :=$ length($string$)

for $j := 1$ to n
    $chart(j-1, j) := \{$X | X $\rightarrow$ word$_j \in$ P$\}$
    for $i := j-2$ down to 0
        $chart(i, j) := \{\}$
        for $k := i+1$ to $j-1$
            for every $A \rightarrow BC \in P$
                if $B \in chart(i, k)$ and $C \in chart(k, j)$ then
                    $chart(i, j) := chart(i, j) \cup \{A\}$

Output: if $S \in chart(0, n)$ then accept else reject

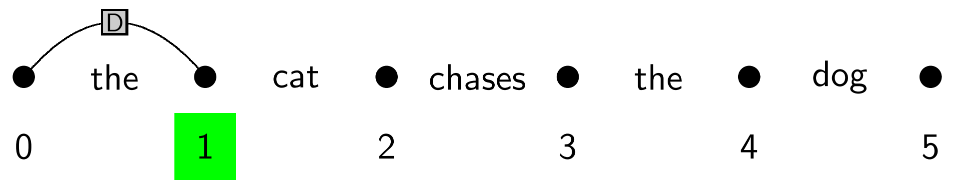
Example Application of the CYK Algorithm

s $\longrightarrow$ np vp
np $\longrightarrow$ d n
vp $\longrightarrow$ v np

d $\longrightarrow$ the
n $\longrightarrow$ dog
n $\longrightarrow$ cat
v $\longrightarrow$ chases

Lexical Entry: *the* ($j = 1$, field chart(0,1))

	1	2	3	4	5
0	d				
1					
2					
3					
4					



Example Application of the CYK Algorithm

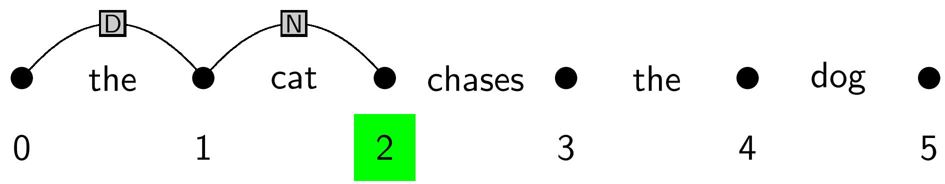
s $\longrightarrow$ np vp
np $\longrightarrow$ d n
vp $\longrightarrow$ v np

d $\longrightarrow$ the
n $\longrightarrow$ dog
n $\longrightarrow$ cat
v $\longrightarrow$ chases

Lexical Entry: *cat*

($j = 2$, field chart(1,2))

	1	2	3	4	5
0	d				
1		n			
2					
3					
4					



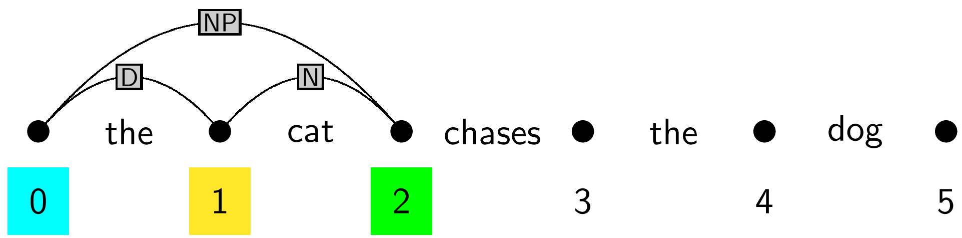
Example Application of the CYK Algorithm

s $\longrightarrow$ np vp
np $\longrightarrow$ d n
vp $\longrightarrow$ v np

d $\longrightarrow$ the
n $\longrightarrow$ dog
n $\longrightarrow$ cat
v $\longrightarrow$ chases

$j = 2$
$i = 0$
$k = 1$

	1	2	3	4	5
0	d	np			
1		n			
2					
3					
4					



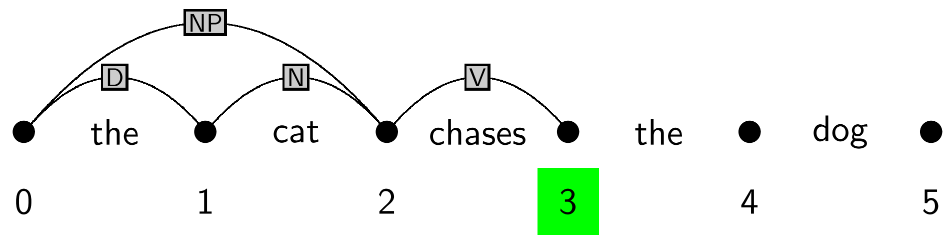
Example Application of the CYK Algorithm

s → np vp
np → d n
vp → v np

d → the
n → dog
n → cat
v → chases

Lexical Entry: *chases* ($j = 3$, field chart(2,3))

	1	2	3	4	5
0	d	np			
1		n			
2			v		
3					
4					



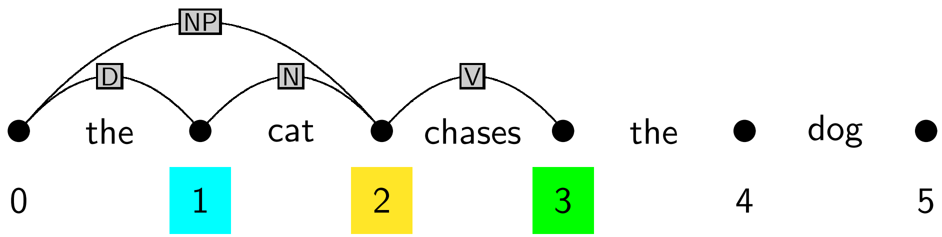
Example Application of the CYK Algorithm

s $\longrightarrow$ np vp
np $\longrightarrow$ d n
vp $\longrightarrow$ v np

d $\longrightarrow$ the
n $\longrightarrow$ dog
n $\longrightarrow$ cat
v $\longrightarrow$ chases

j = 3
i = 1
k = 2

	1	2	3	4	5
0	d	np			
1		n			
2			v		
3					
4					



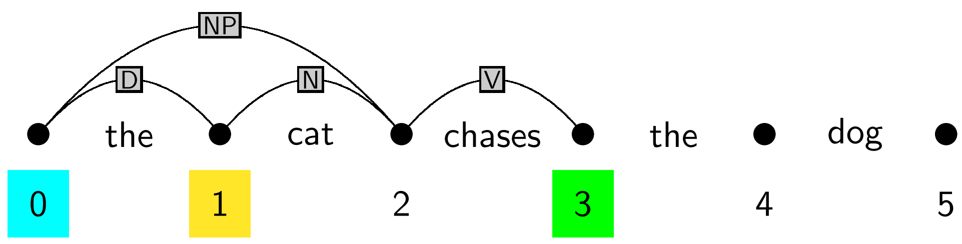
Example Application of the CYK Algorithm

s ⟶ np vp
np ⟶ d n
vp ⟶ v np

d ⟶ the
n ⟶ dog
n ⟶ cat
v ⟶ chases

j = 3
i = 0
k = 1

	1	2	3	4	5
0	d	np			
1		n			
2			v		
3					
4					



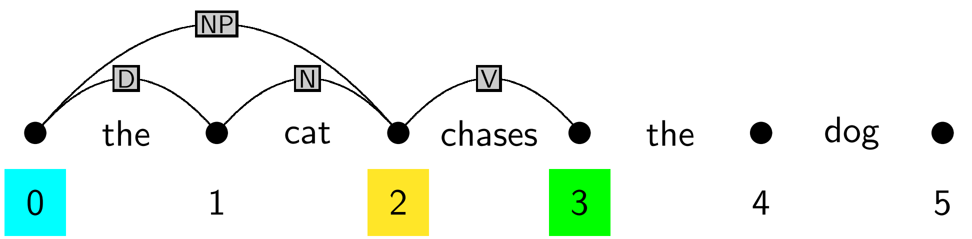
Example Application of the CYK Algorithm

s ⟶ np vp
np ⟶ d n
vp ⟶ v np

d ⟶ the
n ⟶ dog
n ⟶ cat
v ⟶ chases

$j = 3$
$i = 0$
$k = 2$

	1	2	3	4	5
0	d	np			
1		n			
2			v		
3					
4					



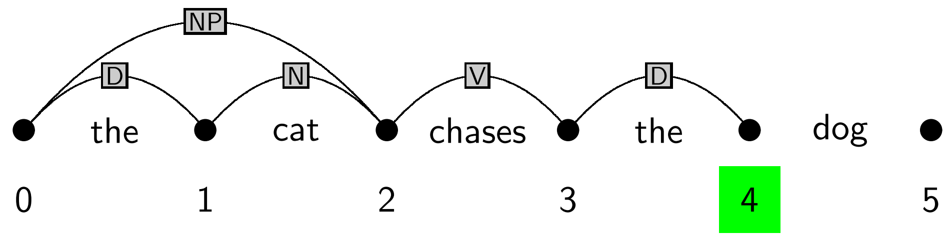
Example Application of the CYK Algorithm

s → np vp
np → d n
vp → v np

d → the
n → dog
n → cat
v → chases

Lexical Entry: *the* ($j = 4$, field chart(3,4))

	1	2	3	4	5
0	d	np			
1		n			
2			v		
3				d	
4					



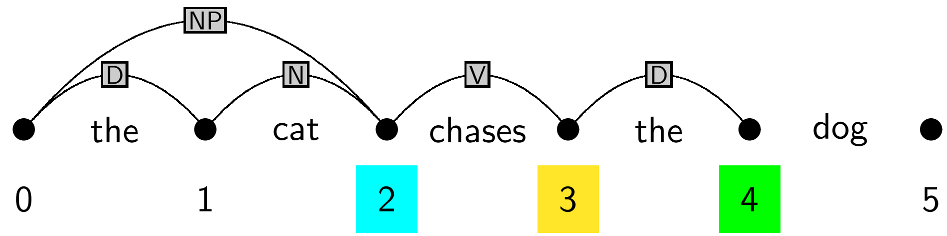
Example Application of the CYK Algorithm

s ⟶ np vp
np ⟶ d n
vp ⟶ v np

d ⟶ the
n ⟶ dog
n ⟶ cat
v ⟶ chases

$j = 4$
$i = 2$
$k = 3$

	1	2	3	4	5
0	d	np			
1		n			
2			v		
3				d	
4					



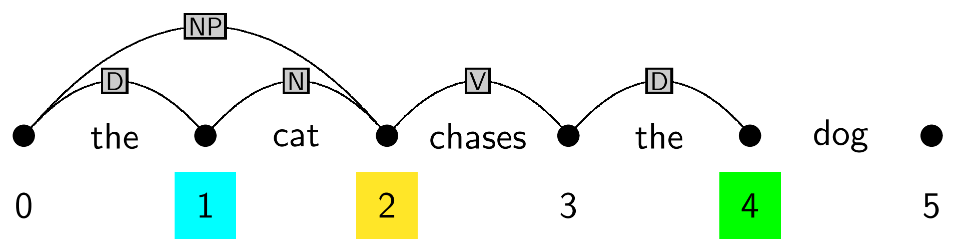
Example Application of the CYK Algorithm

s → np vp
np → d n
vp → v np

d → the
n → dog
n → cat
v → chases

j = 4
i = 1
k = 2

	1	2	3	4	5
0	d	np			
1		n			
2			v		
3				d	
4					



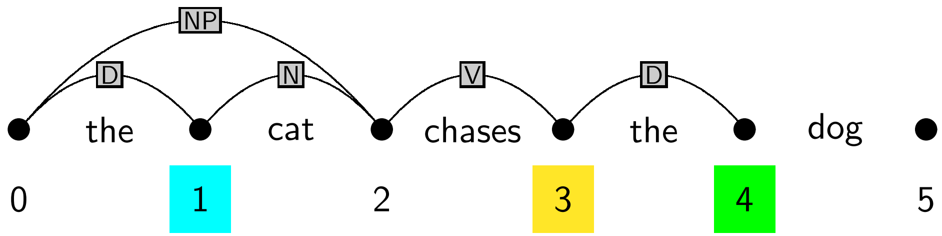
Example Application of the CYK Algorithm

s $\longrightarrow$ np vp
np $\longrightarrow$ d n
vp $\longrightarrow$ v np

d $\longrightarrow$ the
n $\longrightarrow$ dog
n $\longrightarrow$ cat
v $\longrightarrow$ chases

$j = 4$
$i = 1$
$k = 3$

	1	2	3	4	5
0	d	np			
1		n			
2			v		
3				d	
4					



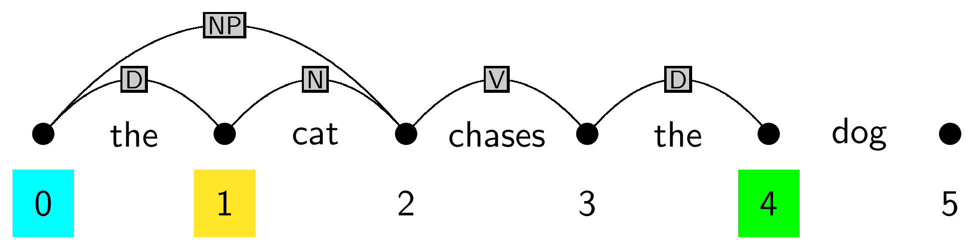
Example Application of the CYK Algorithm

s $\longrightarrow$ np vp
np $\longrightarrow$ d n
vp $\longrightarrow$ v np

d $\longrightarrow$ the
n $\longrightarrow$ dog
n $\longrightarrow$ cat
v $\longrightarrow$ chases

$j = 4$
$i = 0$
$k = 1$

	1	2	3	4	5
0	d	np			
1		n			
2			v		
3				d	
4					



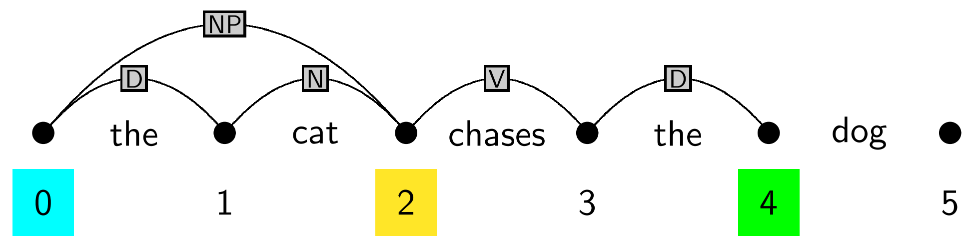
Example Application of the CYK Algorithm

s $\longrightarrow$ np vp
np $\longrightarrow$ d n
vp $\longrightarrow$ v np

d $\longrightarrow$ the
n $\longrightarrow$ dog
n $\longrightarrow$ cat
v $\longrightarrow$ chases

$j = 4$
$i = 0$
$k = 2$

	1	2	3	4	5
0	d	np			
1		n			
2			v		
3				d	
4					



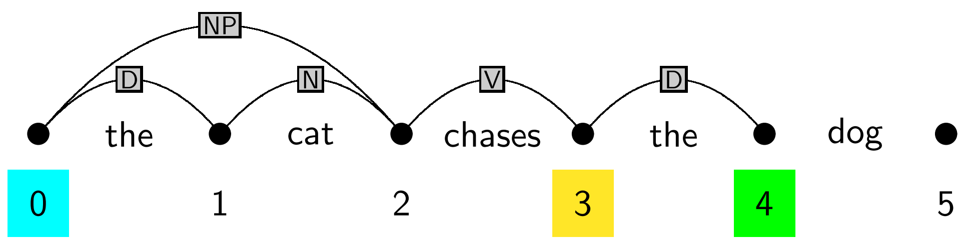
Example Application of the CYK Algorithm

s ⟶ np vp
np ⟶ d n
vp ⟶ v np

d ⟶ the
n ⟶ dog
n ⟶ cat
v ⟶ chases

j = 4
i = 0
k = 3

	1	2	3	4	5
0	d	np			
1		n			
2			v		
3				d	
4					



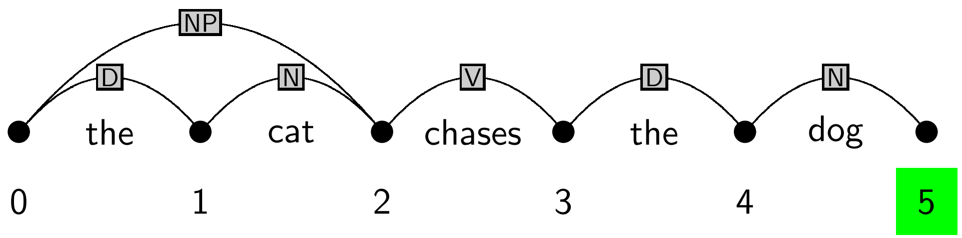
Example Application of the CYK Algorithm

s $\longrightarrow$ np vp
np $\longrightarrow$ d n
vp $\longrightarrow$ v np

d $\longrightarrow$ the
n $\longrightarrow$ dog
n $\longrightarrow$ cat
v $\longrightarrow$ chases

Lexical Entry: *dog* (j = 5 , field chart(4,5))

	1	2	3	4	5
0	d	np			
1		n			
2			v		
3				d	
4					n



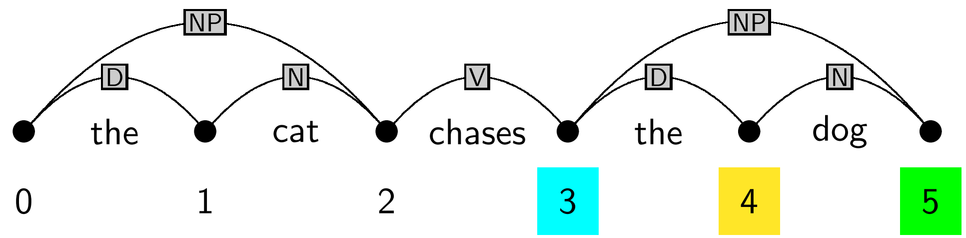
Example Application of the CYK Algorithm

s $\longrightarrow$ np vp
np $\longrightarrow$ d n
vp $\longrightarrow$ v np

d $\longrightarrow$ the
n $\longrightarrow$ dog
n $\longrightarrow$ cat
v $\longrightarrow$ chases

j = 5
i = 3
k = 4

	1	2	3	4	5
0	d	np			
1		n			
2			v		
3				d	np
4					n



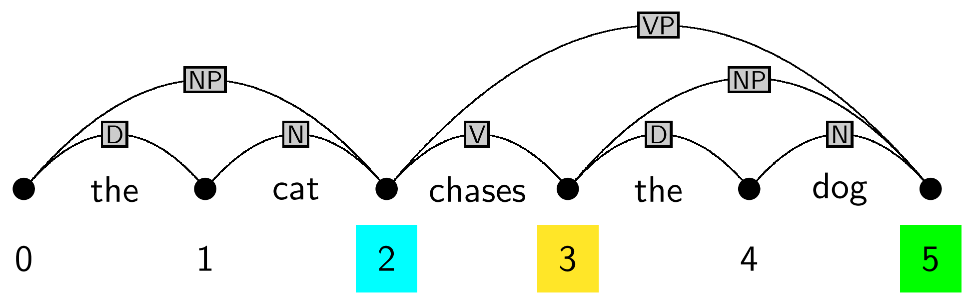
Example Application of the CYK Algorithm

s → np vp
np → d n
vp → v np

d → the
n → dog
n → cat
v → chases

j = 5
i = 2
k = 3

	1	2	3	4	5
0	d	np			
1		n			
2			v		vp
3				d	np
4					n



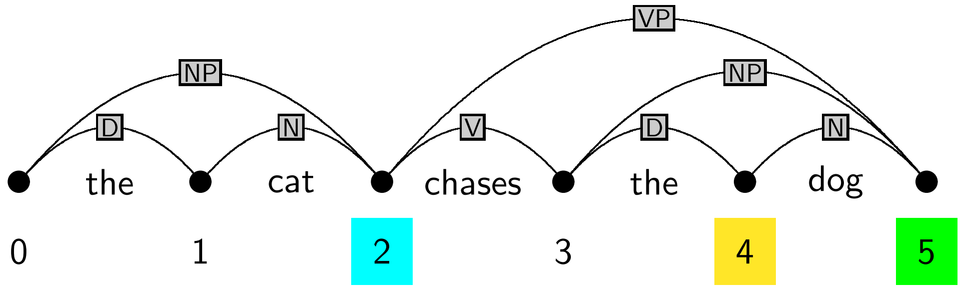
Example Application of the CYK Algorithm

s $\longrightarrow$ np vp
np $\longrightarrow$ d n
vp $\longrightarrow$ v np

d $\longrightarrow$ the
n $\longrightarrow$ dog
n $\longrightarrow$ cat
v $\longrightarrow$ chases

$j = 5$
$i = 2$
$k = 4$

	1	2	3	4	5
0	d	np			
1		n			
2			v		vp
3				d	np
4					n



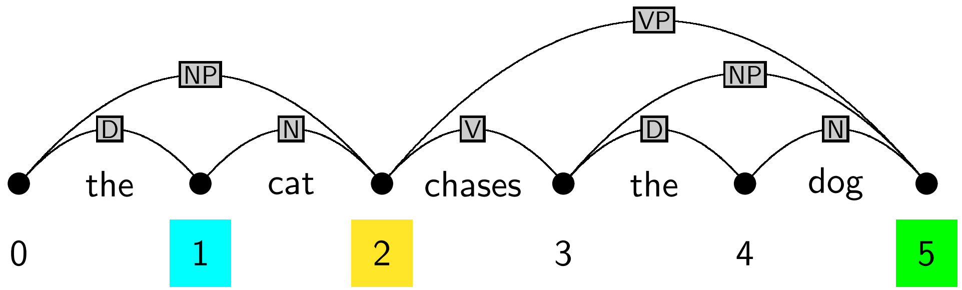
Example Application of the CYK Algorithm

s ⟶ np vp
np ⟶ d n
vp ⟶ v np

d ⟶ the
n ⟶ dog
n ⟶ cat
v ⟶ chases

j = 5
i = 1
k = 2

	1	2	3	4	5
0	d	np			
1		n			
2			v		vp
3				d	np
4					n



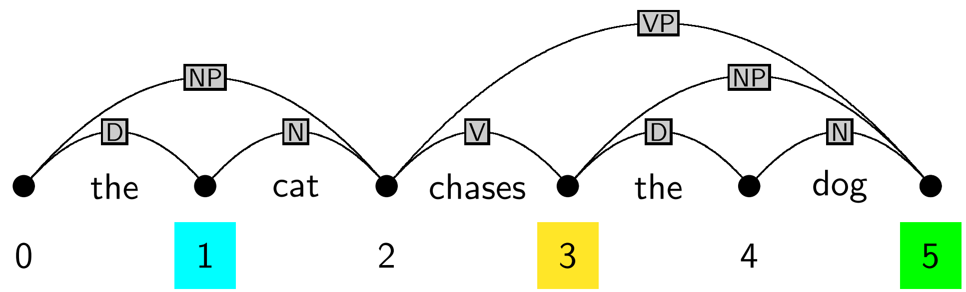
Example Application of the CYK Algorithm

s → np vp
np → d n
vp → v np

d → the
n → dog
n → cat
v → chases

$j = 5$
$i = 1$
$k = 3$

	1	2	3	4	5
0	d	np			
1		n			
2			v		vp
3				d	np
4					n



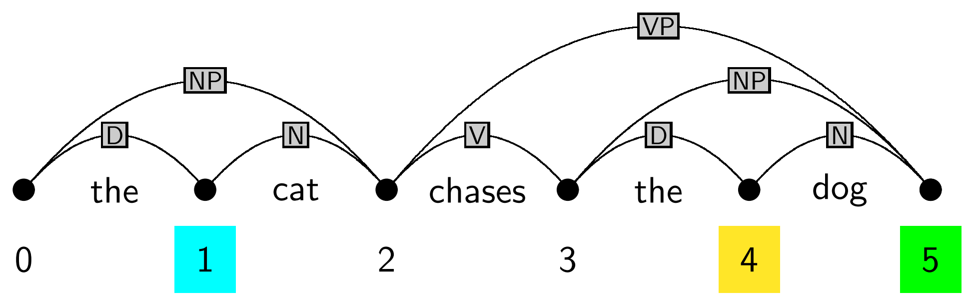
Example Application of the CYK Algorithm

s → np vp
np → d n
vp → v np

d → the
n → dog
n → cat
v → chases

j = 5
i = 1
k = 4

	1	2	3	4	5
0	d	np			
1		n			
2			v		vp
3				d	np
4					n



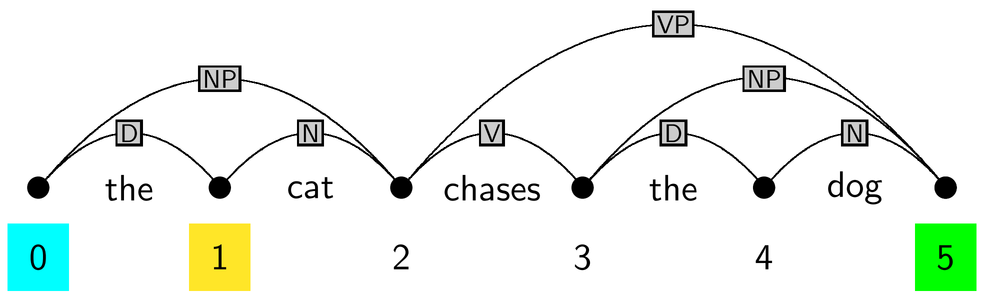
Example Application of the CYK Algorithm

s ⟶ np vp
np ⟶ d n
vp ⟶ v np

d ⟶ the
n ⟶ dog
n ⟶ cat
v ⟶ chases

j = 5
i = 0
k = 1

	1	2	3	4	5
0	d	np			
1		n			
2			v		vp
3				d	np
4					n



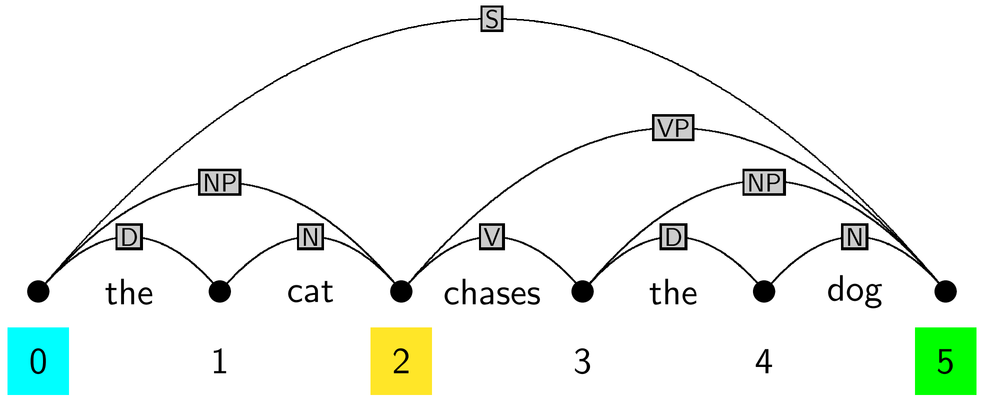
Example Application of the CYK Algorithm

s → np vp
np → d n
vp → v np

d → the
n → dog
n → cat
v → chases

$j = 5$
$i = 0$
$k = 2$

	1	2	3	4	5
0	d	np			s
1		n			
2			v		vp
3				d	np
4					n

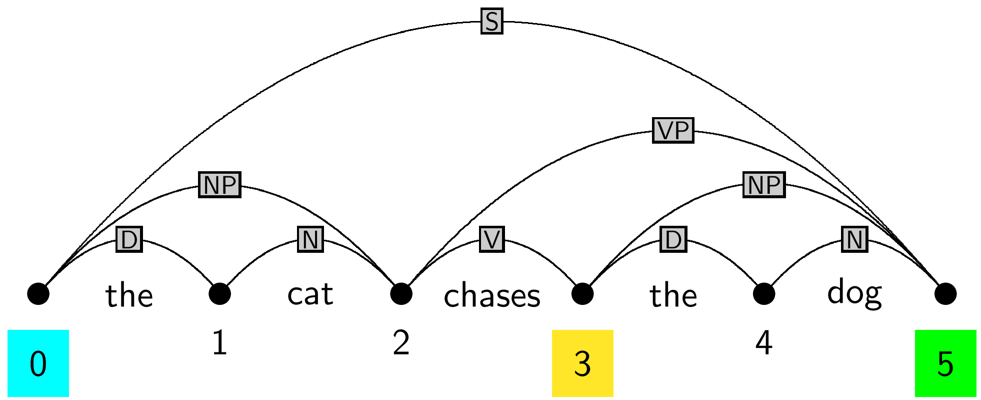


Example Application of the CYK Algorithm

$s \longrightarrow np\ vp$ $d \longrightarrow the$
$np \longrightarrow d\ n$ $n \longrightarrow dog$
$vp \longrightarrow v\ np$ $n \longrightarrow cat$
$v \longrightarrow chases$

$j = 5$
$i = 0$
$k = 3$

	1	2	3	4	5
0	d	np			s
1		n			
2			v		vp
3				d	np
4					n



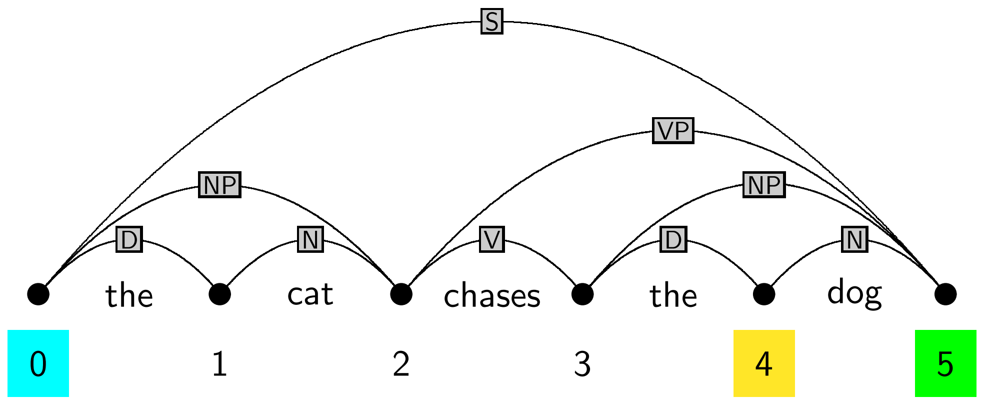
Example Application of the CYK Algorithm

s → np vp
np → d n
vp → v np

d → the
n → dog
n → cat
v → chases

$j = 5$
$i = 0$
$k = 4$

	1	2	3	4	5
0	d	np			s
1		n			
2			v		vp
3				d	np
4					n



Outline

- Phrase Structure
- A Topological Model for German
- From Phrase Structure to GPSG and HPSG
- German Word Order Phenomena and their Analyses in Standard-HPSG/GPSG
- Discontinuous Constituents
- Parsing
  - With Continuous Constituents
  - With Discontinuous Constituents
- Syntactically Annotated Corpora and Discontinuity
- Summary

Representing Discontinuous Constituents

(47) a. John **picked up** his son from school yesterday.

Representing Discontinuous Constituents

(47) a. John **picked up** his son from school yesterday.

b. John **picked** his son **up** from school yesterday.

Representing Discontinuous Constituents

(47) a. John **picked up** his son from school yesterday.
b. John **picked** his son **up** from school yesterday.

So how can the coverage of the constituent **picked up** in (47b) be represented?

Representing Discontinuous Constituents

(47) a. John **picked up** his son from school yesterday.
b. John **picked** his son **up** from school yesterday.

So how can the coverage of the constituent **picked up** in (47b) be represented?

John **picked** his son **up** from school yesterday

Representing Discontinuous Constituents

(47) a. John **picked up** his son from school yesterday.
b. John **picked** his son **up** from school yesterday.

So how can the coverage of the constituent **picked up** in (47b) be represented?

$\cdot_0$ John $\cdot_1$ **picked** $\cdot_2$ his $\cdot_3$ son $\cdot_4$ **up** $\cdot_5$ from $\cdot_6$ school $\cdot_7$ yesterday $\cdot_8$

Representing Discontinuous Constituents

(47) a. John **picked up** his son from school yesterday.
b. John **picked** his son **up** from school yesterday.

So how can the coverage of the constituent **picked up** in (47b) be represented?

$\cdot_0$ John $\cdot_1$ **picked** $\cdot_2$ his $\cdot_3$ son $\cdot_4$ **up** $\cdot_5$ from $\cdot_6$ school $\cdot_7$ yesterday $\cdot_8$

- Interval lists (Johnson, 1985): `[[1,2],[4,5]]`

Representing Discontinuous Constituents

(47) a. John **picked up** his son from school yesterday.
b. John **picked** his son **up** from school yesterday.

So how can the coverage of the constituent **picked up** in (47b) be represented?

$\cdot_0$ John $\cdot_1$ **picked** $\cdot_2$ his $\cdot_3$ son $\cdot_4$ **up** $\cdot_5$ from $\cdot_6$ school $\cdot_7$ yesterday $\cdot_8$
1 2 3 4 5 6 7 8

- Interval lists (Johnson, 1985): `[[1,2],[4,5]]`

Representing Discontinuous Constituents

(47) a. John **picked up** his son from school yesterday.
b. John **picked** his son **up** from school yesterday.

So how can the coverage of the constituent **picked up** in (47b) be represented?

$\cdot_0$ John $\cdot_1$ **picked** $\cdot_2$ his $\cdot_3$ son $\cdot_4$ **up** $\cdot_5$ from $\cdot_6$ school $\cdot_7$ yesterday $\cdot_8$
1 2 3 4 5 6 7 8

- Interval lists (Johnson, 1985): `[[1,2],[4,5]]`
- Bit lists (Reape, 1991): `[0,1,0,0,1,0,0,0]`

Representing Discontinuous Constituents

(47) a. John **picked up** his son from school yesterday.

b. John **picked** his son **up** from school yesterday.

So how can the coverage of the constituent **picked up** in (47b) be represented?

$\cdot_0$ John $\cdot_1$ **picked** $\cdot_2$ his $\cdot_3$ son $\cdot_4$ **up** $\cdot_5$ from $\cdot_6$ school $\cdot_7$ yesterday $\cdot_8$

1 2 3 4 5 6 7 8

- Interval lists (Johnson, 1985): `[[1,2],[4,5]]`
- Bit lists (Reape, 1991): `[0,1,0,0,1,0,0,0]`
- Binary numbers: 00010010_2 ($= 17_{10}$)

Representing Discontinuous Constituents

(47) a. John **picked up** his son from school yesterday.
b. John **picked** his son **up** from school yesterday.

So how can the coverage of the constituent **picked up** in (47b) be represented?

$\cdot_0$ John $\cdot_1$ **picked** $\cdot_2$ his $\cdot_3$ son $\cdot_4$ **up** $\cdot_5$ from $\cdot_6$ school $\cdot_7$ yesterday $\cdot_8$
1 2 3 4 5 6 7 8

- Interval lists (Johnson, 1985): `[[1,2],[4,5]]`
- Bit lists (Reape, 1991): `[0,1,0,0,1,0,0,0]`
- Binary numbers: 00010010_2 ($= 17_{10}$)
  - The left-most word in the input corresponds to
  - the least significant bit (= right-most) of the number.

Towards Parsing with Discontinuous Constituents

- Fundamental questions:

  a) When can two constituents be combined?
  b) What is the coverage of the resulting new constituent?

Towards Parsing with Discontinuous Constituents

- Fundamental questions:

  a) When can two constituents be combined?
  b) What is the coverage of the resulting new constituent?

- Traditional parsing:

  a) First constituent ends where the second one starts
     e.g., $[i, B, k] + [k, C, j]$

Towards Parsing with Discontinuous Constituents

- Fundamental questions:

  a) When can two constituents be combined?
  b) What is the coverage of the resulting new constituent?

- Traditional parsing:

  a) First constituent ends where the second one starts
  e.g., $[i, B, k] + [k, C, j]$
  b) Covers interval from start of the first constituent to end of the second.
  e.g., $[i, B, k] + [k, C, j] = [i, A, j]$ given a rule $A \rightarrow BC$

Parsing with Discontinuous Constituents

(48) John **picked** his son **up** from school yesterday.

Parsing with Discontinuous Constituents

(48) John **picked** his son **up** from school yesterday.

Relevant grammar rules:

(1) $\mathsf{V}_{trans} \rightarrow_{dom} picked\ up$
(2) $\mathsf{NP} \rightarrow_{dom} his\ son$
(3) $\mathsf{VP} \rightarrow_{dom} \mathsf{V}_{trans}\ \mathsf{NP}$

Parsing with Discontinuous Constituents

(48) John **picked** his son **up** from school yesterday.

Relevant grammar rules:

(1) $\mathsf{V}_{trans} \rightarrow_{dom} picked\ up$
(2) $\mathsf{NP} \rightarrow_{dom} his\ son$
(3) $\mathsf{VP} \rightarrow_{dom} \mathsf{V}_{trans}\ \mathsf{NP}$

Combining two constituents using rule (3):

Parsing with Discontinuous Constituents

(48) John **picked** his son **up** from school yesterday.

Relevant grammar rules:

(1) $\mathsf{V}_{trans} \rightarrow_{dom} picked\ up$
(2) $\mathsf{NP} \rightarrow_{dom} his\ son$
(3) $\mathsf{VP} \rightarrow_{dom} \mathsf{V}_{trans}\ \mathsf{NP}$

Combining two constituents using rule (3):

picked up (V) 0 1 0 0 1 0 0 0

Parsing with Discontinuous Constituents

(48) John **picked** his son **up** from school yesterday.

Relevant grammar rules:

(1) $\mathsf{V}_{trans} \rightarrow_{dom} picked\ up$
(2) $\mathsf{NP} \rightarrow_{dom} his\ son$
(3) $\mathsf{VP} \rightarrow_{dom} \mathsf{V}_{trans}\ \mathsf{NP}$

Combining two constituents using rule (3):

picked up (V)	0	1	0	0	1	0	0	0
his son (NP)	0	0	1	1	0	0	0	0

Parsing with Discontinuous Constituents

(48) John **picked** his son **up** from school yesterday.

Relevant grammar rules:

(1) $\text{V}_{trans} \rightarrow_{dom}$ *picked up*
(2) $\text{NP} \rightarrow_{dom}$ *his son*
(3) $\text{VP} \rightarrow_{dom} \text{V}_{trans}\ \text{NP}$

Combining two constituents using rule (3):

picked up (V)	0	1	0	0	1	0	0	0
his son (NP)	0	0	1	1	0	0	0	0
picked his son **up** (VP)	0	1	1	1	1	0	0	0

Parsing with Discontinuous Constituents

(48) John **picked** his son **up** from school yesterday.

Relevant grammar rules:

(1) $\mathsf{V}_{trans} \rightarrow_{dom} picked\ up$
(2) $\mathsf{NP} \rightarrow_{dom} his\ son$
(3) $\mathsf{VP} \rightarrow_{dom} \mathsf{V}_{trans}\ \mathsf{NP}$

Combining two constituents using rule (3):

picked up (V)	0	1	0	0	1	0	0	0
his son (NP)	0	0	1	1	0	0	0	0
picked his son **up** (VP)	0	1	1	1	1	0	0	0

The combined coverage of two edges:

The combined coverage is the bitwise-or of the bit vectors of its parts.

When are two Discontinuous Constituents Compatible?

(49)	daß	der	Mann	der	Königin	die	Krone	überreichte.
	that	[the	man	[the$_{dat}$	queen	[the	crown	gave]]]
		[the	husband	of.the$_{gen}$	queen]			

When are two Discontinuous Constituents Compatible?

(49)	daß	der	Mann	der	Königin	die	Krone	überreichte.	
	that	[the	man	[the$_{dat}$	queen	[the	crown	gave]]]	
		[the	husband	of.the$_{gen}$	queen]				

Relevant grammar rules:

(1) NP $\rightarrow_{dom}$ *der Mann der Königin*
(2) VP $\rightarrow_{dom}$ *der Königin die Krone überreichte*
(3) S $\rightarrow_{dom}$ NP VP

When are two Discontinuous Constituents Compatible?

(49)	daß	der	Mann	der	Königin	die	Krone	überreichte.
	that	[the	man	[the$_{dat}$	queen	[the	crown	gave]]]
		[the	husband	of.the$_{gen}$	queen]			

Relevant grammar rules:

(1) NP $\rightarrow_{dom}$ *der Mann der Königin*
(2) VP $\rightarrow_{dom}$ *der Königin die Krone überreichte*
(3) S $\rightarrow_{dom}$ NP VP

Combining two constituents using rule (3):

When are two Discontinuous Constituents Compatible?

(49)	daß	der	Mann	der	Königin	die	Krone	überreichte.
	that	[the	man	[the$_{dat}$	queen	[the	crown	gave]]]
		[the	husband	of.the$_{gen}$	queen]			

Relevant grammar rules:

(1) NP $\rightarrow_{dom}$ *der Mann der Königin*
(2) VP $\rightarrow_{dom}$ *der Königin die Krone überreichte*
(3) S $\rightarrow_{dom}$ NP VP

Combining two constituents using rule (3):

der Mann der Königin (NP) 0 1 1 1 1 0 0 0

When are two Discontinuous Constituents Compatible?

(49)	daß	der	Mann	der	Königin	die	Krone	überreichte.
	that	[the	man	[the$_{dat}$	queen	[the	crown	gave]]]
		[the	husband	of.the$_{gen}$	queen]			

Relevant grammar rules:

(1) NP $\rightarrow_{dom}$ *der Mann der Königin*
(2) VP $\rightarrow_{dom}$ *der Königin die Krone überreichte*
(3) S $\rightarrow_{dom}$ NP VP

Combining two constituents using rule (3):

der Mann der Königin (NP)	0	1	1	1	1	0	0	0
der Königin die Krone überreichte (VP)	0	0	0	1	1	1	1	1

When are two Discontinuous Constituents Compatible?

(49)	daß	der	Mann	der	Königin	die	Krone überreichte.
	that	[the	man	[the$_{dat}$	queen	[the crown gave]]]	
		[the	husband	of.the$_{gen}$	queen]		

Relevant grammar rules:

(1) NP $\rightarrow_{dom}$ *der Mann der Königin*
(2) VP $\rightarrow_{dom}$ *der Königin die Krone überreichte*
(3) S $\rightarrow_{dom}$ NP VP

Combining two constituents using rule (3):

der Mann der Königin (NP)	0	1	1	1	1	0	0	0
der Königin die Krone überreichte (VP)	0	0	0	1	1	1	1	1
der Mann der Königin die Krone überreichte (S)	not compatible!							

When are two Discontinuous Constituents Compatible?

(49) daß der Mann der Königin die Krone überreichte.
that [the man [the$_{dat}$ queen [the crown gave]]]
[the husband of.the$_{gen}$ queen]

Relevant grammar rules:

(1) NP $\rightarrow_{dom}$ *der Mann der Königin*
(2) VP $\rightarrow_{dom}$ *der Königin die Krone überreichte*
(3) S $\rightarrow_{dom}$ NP VP

Combining two constituents using rule (3):

der Mann der Königin (NP)	0 1 1 1 1 0 0 0
der Königin die Krone überreichte (VP)	0 0 0 1 1 1 1 1
der Mann der Königin die Krone überreichte (S)	not compatible!

Compatibility of two edges:

Two edges are compatible if the bitwise-and of the two edges is 0.

Worst-Case Complexity of Linearization Parsing

- continuous constituents: characterized by the covered string interval

Worst-Case Complexity of Linearization Parsing

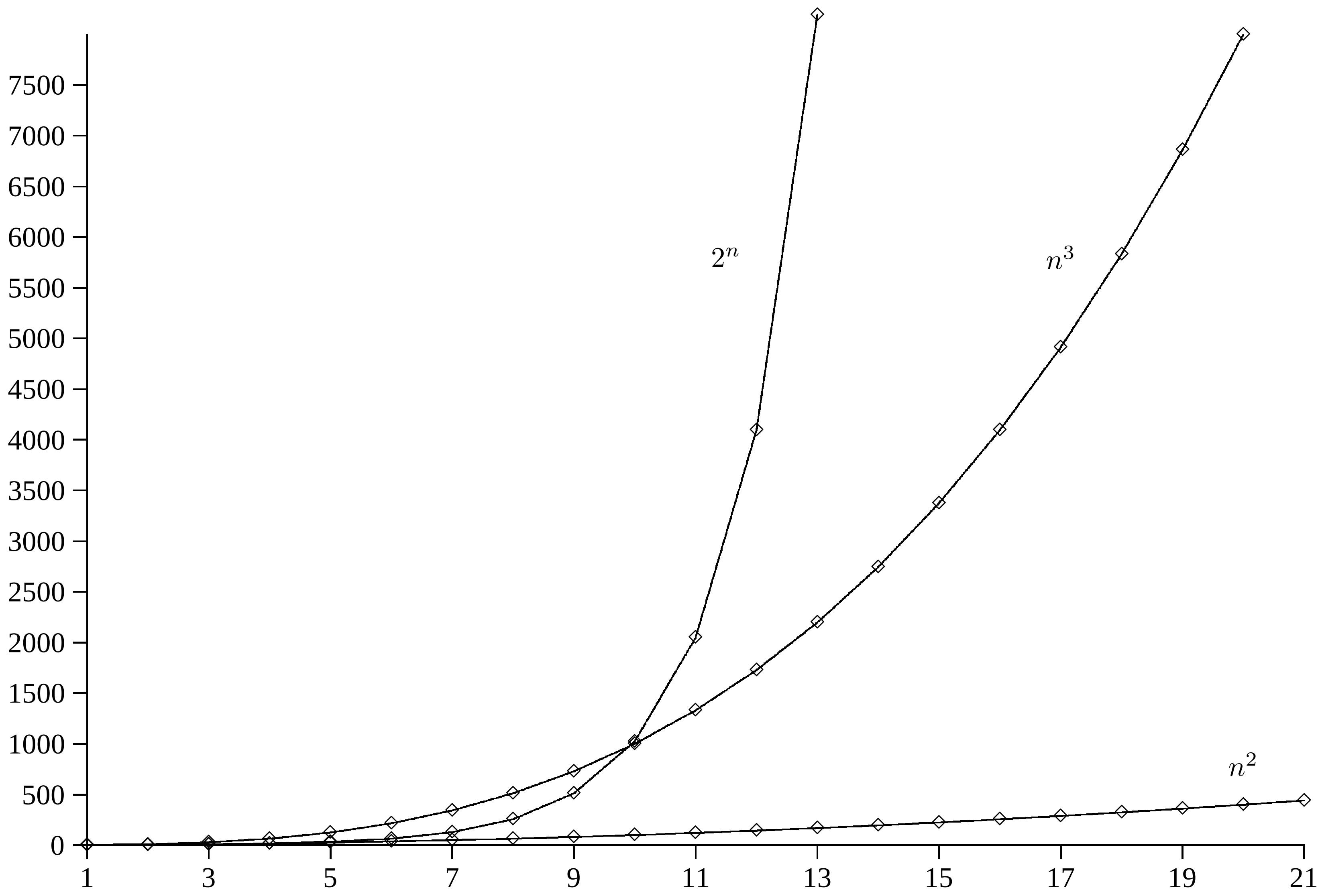
- continuous constituents: characterized by the covered string interval
- discontinuous constituents: characterized by the covered subset of the input

Worst-Case Complexity of Linearization Parsing

- continuous constituents: characterized by the covered string interval
- discontinuous constituents: characterized by the covered subset of the input
- worst-case number of constituents the parser needs to consider:

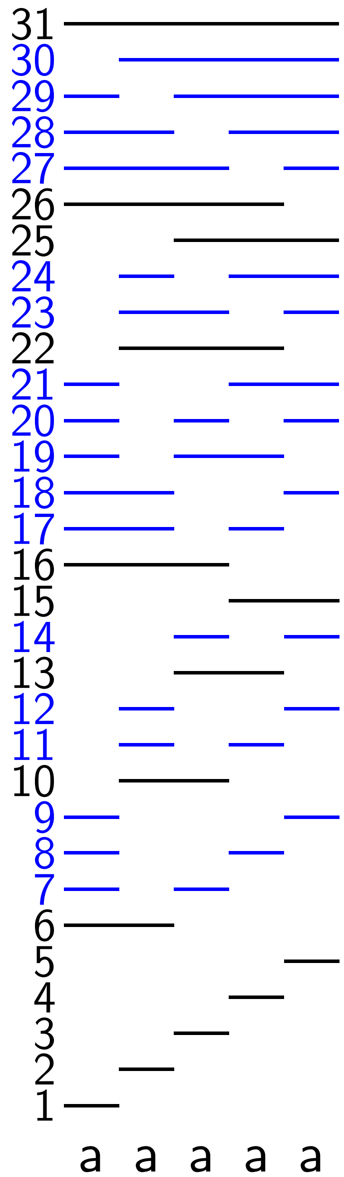
Length of sentence	Possible sub-intervals	Possible subsets
5	15	32
10	55	1024
15	120	32768
20	210	1048576
n	$O(n^2)$	$O(2^n)$

Worst-Case Complexity (II)



Illustrating Worst-Case Complexity

Licensed constituents:



Example grammar:

S $\rightarrow_{dom}$ a
S $\rightarrow_{dom}$ SS

Where does the Complexity Arise?

- Suhre (1999): membership problem for a formally-equivalent grammar formalism is NP-complete
  - general membership problem: considering grammar plus string as input
  - fixed membership problem: considering only string as input

Where does the Complexity Arise?

- Suhre (1999): membership problem for a formally-equivalent grammar formalism is NP-complete
  - general membership problem: considering grammar plus string as input
  - fixed membership problem: considering only string as input
- Huynh (1983): general membership problem for unordered context-free grammars (ID/LP grammars without LP statements) is also NP-complete

Where does the Complexity Arise?

- Suhre (1999): membership problem for a formally-equivalent grammar formalism is NP-complete
  - general membership problem: considering grammar plus string as input
  - fixed membership problem: considering only string as input
- Huynh (1983): general membership problem for unordered context-free grammars (ID/LP grammars without LP statements) is also NP-complete
- Suhre (1999, 61ff): fixed membership problem stems from potential from recursive growth of discontinuities

Escaping the Worst-Case Complexity

- Suhre (1999): When the the number of discontinuities introduced by a recursive non-terminal is bounded by some constant, the fixed membership problem becomes polynomial.

Escaping the Worst-Case Complexity

- Suhre (1999): When the the number of discontinuities introduced by a recursive non-terminal is bounded by some constant, the fixed membership problem becomes polynomial.

- Continuity constraint for linearization-based HPSG (Müller, 1999): require saturated phrasal elements (that is, maximal projections) to be continuous

Escaping the Worst-Case Complexity

- Suhre (1999): When the the number of discontinuities introduced by a recursive non-terminal is bounded by some constant, the fixed membership problem becomes polynomial.
- Continuity constraint for linearization-based HPSG (Müller, 1999): require saturated phrasal elements (that is, maximal projections) to be continuous
  - formally weaker than Suhre's condition since recursion on level of adjunction not restricted

Escaping the Worst-Case Complexity

- Suhre (1999): When the the number of discontinuities introduced by a recursive non-terminal is bounded by some constant, the fixed membership problem becomes polynomial.

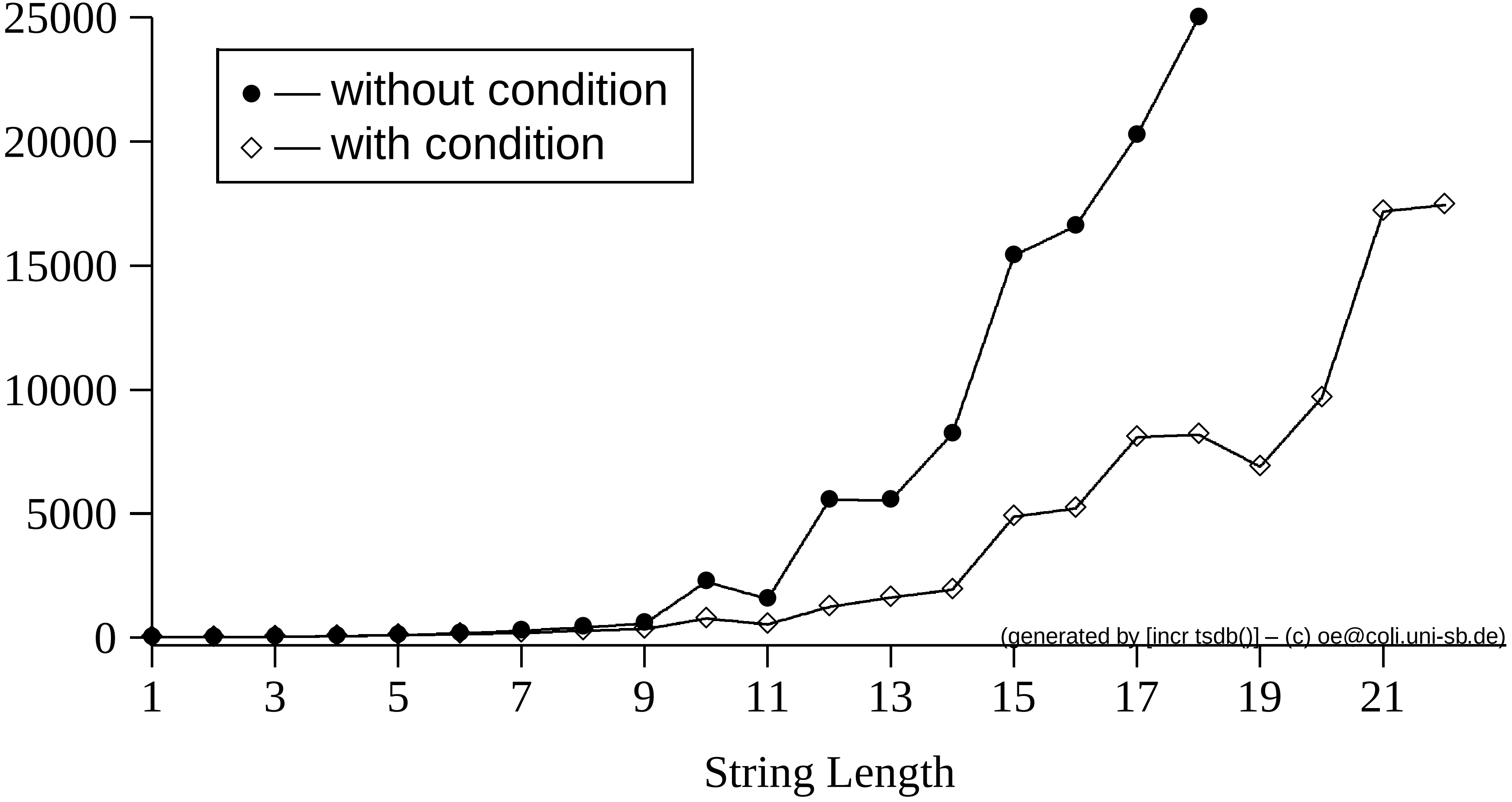
- Continuity constraint for linearization-based HPSG (Müller, 1999): require saturated phrasal elements (that is, maximal projections) to be continuous
  - formally weaker than Suhre's condition since recursion on level of adjunction not restricted
  - grammars incorporating the $\overline{\text{X}}$-schema (Jackendoff, 1977) require non-head constituents to be maximal projections

Escaping the Worst-Case Complexity

- Suhre (1999): When the the number of discontinuities introduced by a recursive non-terminal is bounded by some constant, the fixed membership problem becomes polynomial.

- Continuity constraint for linearization-based HPSG (Müller, 1999): require saturated phrasal elements (that is, maximal projections) to be continuous
  - formally weaker than Suhre's condition since recursion on level of adjunction not restricted
  - grammars incorporating the $\overline{\text{X}}$-schema (Jackendoff, 1977) require non-head constituents to be maximal projections
  - Linguistically-motivated continuity constraints have the potential of leading to efficiently parsable linearization grammars.

Illustrating the Potential: The Effect of Continuity Constraints



Passive edges vs. string length for full parses

Illustrating the Potential: Babel vs. Verb*mobil* grammar

- Two grammars for German were developed:
  - linearization grammar in Babel system (Müller, 1996)

Illustrating the Potential: Babel vs. Verb*mobil* grammar

- Two grammars for German were developed:
  - linearization grammar in Babel system (Müller, 1996)
  - phrase structure grammar in the Verb*mobil* project (Müller and Kasper, 2000) run under the PET system (Callmeier, 2000)

Illustrating the Potential: Babel vs. Verb*mobil* grammar

- Two grammars for German were developed:
  - linearization grammar in Babel system (Müller, 1996)
  - phrase structure grammar in the Verb*mobil* project (Müller and Kasper, 2000) run under the PET system (Callmeier, 2000)
- Coverage:
  - Both grammars cover more than 80% of the Verb*mobil* data

Illustrating the Potential: Babel vs. Verb*mobil* grammar

- Two grammars for German were developed:
  - linearization grammar in Babel system (Müller, 1996)
  - phrase structure grammar in the Verb*mobil* project (Müller and Kasper, 2000) run under the PET system (Callmeier, 2000)
- Coverage:
  - Both grammars cover more than 80% of the Verb*mobil* data
  - Babel grammar covers additional phenomena not needed for the Verb*mobil* domain (Müller, To Appear a, p. 33–39, see also reader p. 17):

Illustrating the Potential: Babel vs. Verb*mobil* grammar

- Two grammars for German were developed:
  - linearization grammar in Babel system (Müller, 1996)
  - phrase structure grammar in the Verb*mobil* project (Müller and Kasper, 2000) run under the PET system (Callmeier, 2000)

- Coverage:
  - Both grammars cover more than 80% of the Verb*mobil* data
  - Babel grammar covers additional phenomena not needed for the Verb*mobil* domain (Müller, To Appear a, p. 33–39, see also reader p. 17):
    - modal infinitives

Illustrating the Potential: Babel vs. Verb*mobil* grammar

- Two grammars for German were developed:
  - linearization grammar in Babel system (Müller, 1996)
  - phrase structure grammar in the Verb*mobil* project (Müller and Kasper, 2000) run under the PET system (Callmeier, 2000)
- Coverage:
  - Both grammars cover more than 80% of the Verb*mobil* data
  - Babel grammar covers additional phenomena not needed for the Verb*mobil* domain (Müller, To Appear a, p. 33–39, see also reader p. 17):
    - modal infinitives
    - depictives

Illustrating the Potential: Babel vs. Verb*mobil* grammar

- Two grammars for German were developed:
  - linearization grammar in Babel system (Müller, 1996)
  - phrase structure grammar in the Verb*mobil* project (Müller and Kasper, 2000) run under the PET system (Callmeier, 2000)
- Coverage:
  - Both grammars cover more than 80% of the Verb*mobil* data
  - Babel grammar covers additional phenomena not needed for the Verb*mobil* domain (Müller, To Appear a, p. 33–39, see also reader p. 17):
    - modal infinitives
    - depictives
    - subclasses of extraposition with and without correlates

Illustrating the Potential: Babel vs. Verb*mobil* grammar

- Two grammars for German were developed:
  - linearization grammar in Babel system (Müller, 1996)
  - phrase structure grammar in the Verb*mobil* project (Müller and Kasper, 2000) run under the PET system (Callmeier, 2000)

- Coverage:
  - Both grammars cover more than 80% of the Verb*mobil* data
  - Babel grammar covers additional phenomena not needed for the Verb*mobil* domain (Müller, To Appear a, p. 33–39, see also reader p. 17):
    - modal infinitives
    - depictives
    - subclasses of extraposition with and without correlates
    - preposition stranding

Illustrating the Potential: Babel vs. Verb*mobil* grammar

- Two grammars for German were developed:
  - linearization grammar in Babel system (Müller, 1996)
  - phrase structure grammar in the Verb*mobil* project (Müller and Kasper, 2000) run under the PET system (Callmeier, 2000)

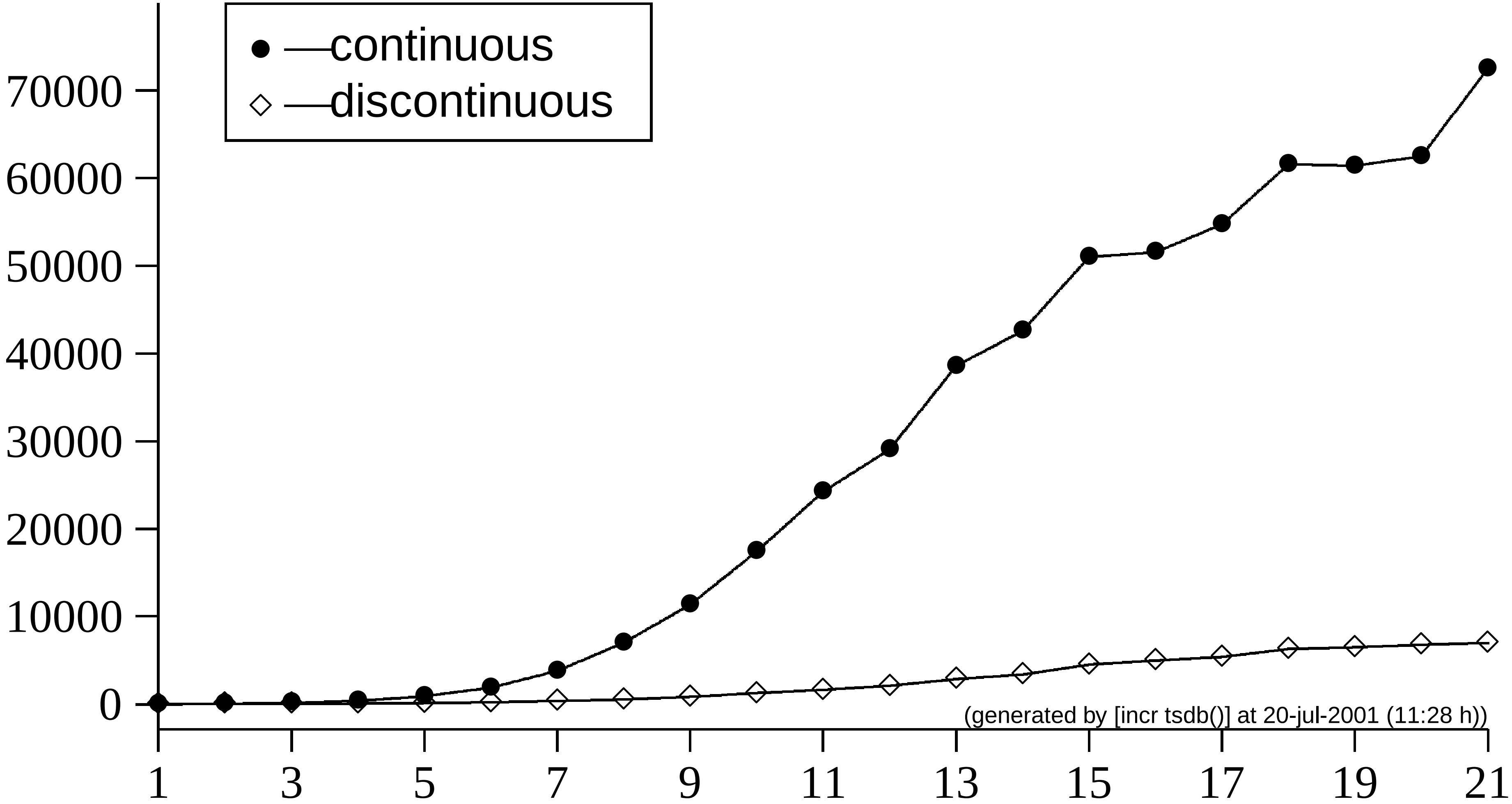
- Coverage:
  - Both grammars cover more than 80% of the Verb*mobil* data
  - Babel grammar covers additional phenomena not needed for the Verb*mobil* domain (Müller, To Appear a, p. 33–39, see also reader p. 17):
    - ∗ modal infinitives
    - ∗ depictives
    - ∗ subclasses of extraposition with and without correlates
    - ∗ preposition stranding
    - ∗ optional coherence

Illustrating the Potential: Babel vs. Verb*mobil* grammar

- Two grammars for German were developed:
  - linearization grammar in Babel system (Müller, 1996)
  - phrase structure grammar in the Verb*mobil* project (Müller and Kasper, 2000) run under the PET system (Callmeier, 2000)

- Coverage:
  - Both grammars cover more than 80% of the Verb*mobil* data
  - Babel grammar covers additional phenomena not needed for the Verb*mobil* domain (Müller, To Appear a, p. 33–39, see also reader p. 17):
    - ∗ modal infinitives
    - ∗ depictives
    - ∗ subclasses of extraposition with and without correlates
    - ∗ preposition stranding
    - ∗ optional coherence
    - ∗ . . .

Passive Edges for a Complete Parse



Further Ideas on a Comparison

- Ideally, a runtime comparison of a traditional and a linearization parsing approach would be based on the same processing environment (to hold other factors constant). Currently there is no system that can handle both. In 2004, the Trale system (Meurers, Penn and Richter, 2002) is intended to support a direct comparison.

Further Ideas on a Comparison

- Ideally, a runtime comparison of a traditional and a linearization parsing approach would be based on the same processing environment (to hold other factors constant). Currently there is no system that can handle both. In 2004, the Trale system (Meurers, Penn and Richter, 2002) is intended to support a direct comparison.

- One can attempt to factor out the general difference in efficiency by comparing runtimes for the same grammar encoded in both systems.

Further Ideas on a Comparison

- Ideally, a runtime comparison of a traditional and a linearization parsing approach would be based on the same processing environment (to hold other factors constant). Currently there is no system that can handle both. In 2004, the Trale system (Meurers, Penn and Richter, 2002) is intended to support a direct comparison.

- One can attempt to factor out the general difference in efficiency by comparing runtimes for the same grammar encoded in both systems.

  – A PP-attachment toy grammar was implemented for PET and Babel.

Further Ideas on a Comparison

- Ideally, a runtime comparison of a traditional and a linearization parsing approach would be based on the same processing environment (to hold other factors constant). Currently there is no system that can handle both. In 2004, the Trale system (Meurers, Penn and Richter, 2002) is intended to support a direct comparison.

- One can attempt to factor out the general difference in efficiency by comparing runtimes for the same grammar encoded in both systems.
  - A PP-attachment toy grammar was implemented for PET and Babel.
  - Parsing a sentence with 9 PPs in PET was 2591 times faster than in Babel.

Further Ideas on a Comparison

- Ideally, a runtime comparison of a traditional and a linearization parsing approach would be based on the same processing environment (to hold other factors constant). Currently there is no system that can handle both. In 2004, the Trale system (Meurers, Penn and Richter, 2002) is intended to support a direct comparison.
- One can attempt to factor out the general difference in efficiency by comparing runtimes for the same grammar encoded in both systems.
  - A PP-attachment toy grammar was implemented for PET and Babel.
  - Parsing a sentence with 9 PPs in PET was 2591 times faster than in Babel.
  - Parsing with the German phrase structure grammar in PET was only 13 times faster than with the German Babel grammar with discontinuities.

Further Ideas on a Comparison

- Ideally, a runtime comparison of a traditional and a linearization parsing approach would be based on the same processing environment (to hold other factors constant). Currently there is no system that can handle both. In 2004, the Trale system (Meurers, Penn and Richter, 2002) is intended to support a direct comparison.

- One can attempt to factor out the general difference in efficiency by comparing runtimes for the same grammar encoded in both systems.
  - A PP-attachment toy grammar was implemented for PET and Babel.
  - Parsing a sentence with 9 PPs in PET was 2591 times faster than in Babel.
  - Parsing with the German phrase structure grammar in PET was only 13 times faster than with the German Babel grammar with discontinuities.

- Current developments:
  - Efficiency improvements in the German phrase structure analysis (Crysmann, 2003).

Further Ideas on a Comparison

- Ideally, a runtime comparison of a traditional and a linearization parsing approach would be based on the same processing environment (to hold other factors constant). Currently there is no system that can handle both. In 2004, the Trale system (Meurers, Penn and Richter, 2002) is intended to support a direct comparison.

- One can attempt to factor out the general difference in efficiency by comparing runtimes for the same grammar encoded in both systems.
  - A PP-attachment toy grammar was implemented for PET and Babel.
  - Parsing a sentence with 9 PPs in PET was 2591 times faster than in Babel.
  - Parsing with the German phrase structure grammar in PET was only 13 times faster than with the German Babel grammar with discontinuities.

- Current developments:
  - Efficiency improvements in the German phrase structure analysis (Crysmann, 2003).
  - Research on efficient parsing with discontinuous constituents (cf., Ramsay, 1999b; Fouvry and Meurers, 2000; Daniels and Meurers, 2002).

Towards Efficient Processing with Discontinuous Constituents

Starting point of Daniels and Meurers (2002):

- The complexity of the parsing algorithm must adapt to the complexity of the specific problem at hand.

Towards Efficient Processing with Discontinuous Constituents

Starting point of Daniels and Meurers (2002):

- The complexity of the parsing algorithm must adapt to the complexity of the specific problem at hand.
- For example, Earley (1970) shows that his parsing algorithm requires linear, quadratic, or cubic effort depending on the complexity of the grammar.

Towards Efficient Processing with Discontinuous Constituents

Starting point of Daniels and Meurers (2002):

- The complexity of the parsing algorithm must adapt to the complexity of the specific problem at hand.
- For example, Earley (1970) shows that his parsing algorithm requires linear, quadratic, or cubic effort depending on the complexity of the grammar.
- Effort = number of edge insertion attempts (whether successful or not)

Towards Efficient Processing with Discontinuous Constituents

Starting point of Daniels and Meurers (2002):

- The complexity of the parsing algorithm must adapt to the complexity of the specific problem at hand.
- For example, Earley (1970) shows that his parsing algorithm requires linear, quadratic, or cubic effort depending on the complexity of the grammar.
- Effort = number of edge insertion attempts (whether successful or not)
- To achieve this goal, we need to make all relevant linearization information accessible in the grammar formalism: GIDLP Grammars

GIDLP Grammars

- The generalized ID/LP grammar format is an extension of the ID/LP format (Gazdar, Klein, Pullum and Sag, 1985).

GIDLP Grammars

- The generalized ID/LP grammar format is an extension of the ID/LP format (Gazdar, Klein, Pullum and Sag, 1985).

- It is inspired by the Linear Specification Language (Götz and Penn, 1997) and the small LSL fragment used by Suhre (1999).

GIDLP Grammars

- The generalized ID/LP grammar format is an extension of the ID/LP format (Gazdar, Klein, Pullum and Sag, 1985).
- It is inspired by the Linear Specification Language (Götz and Penn, 1997) and the small LSL fragment used by Suhre (1999).
- It supports:
  - separation of ID and LP

GIDLP Grammars

- The generalized ID/LP grammar format is an extension of the ID/LP format (Gazdar, Klein, Pullum and Sag, 1985).

- It is inspired by the Linear Specification Language (Götz and Penn, 1997) and the small LSL fragment used by Suhre (1999).

- It supports:
  - separation of ID and LP
  - linearization across local trees

GIDLP Grammars

- The generalized ID/LP grammar format is an extension of the ID/LP format (Gazdar, Klein, Pullum and Sag, 1985).
- It is inspired by the Linear Specification Language (Götz and Penn, 1997) and the small LSL fragment used by Suhre (1999).
- It supports:
  - separation of ID and LP
  - linearization across local trees
  - linearization constraints across local trees within the same domain

GIDLP Grammars

- The generalized ID/LP grammar format is an extension of the ID/LP format (Gazdar, Klein, Pullum and Sag, 1985).
- It is inspired by the Linear Specification Language (Götz and Penn, 1997) and the small LSL fragment used by Suhre (1999).
- It supports:
  - separation of ID and LP
  - linearization across local trees
  - linearization constraints across local trees within the same domain
  - Reape-style compaction of domains, incl. partial compaction

GIDLP Grammars

- The generalized ID/LP grammar format is an extension of the ID/LP format (Gazdar, Klein, Pullum and Sag, 1985).
- It is inspired by the Linear Specification Language (Götz and Penn, 1997) and the small LSL fragment used by Suhre (1999).
- It supports:
  - separation of ID and LP
  - linearization across local trees
  - linearization constraints across local trees within the same domain
  - Reape-style compaction of domains, incl. partial compaction
  - LP statements with reference to

GIDLP Grammars

- The generalized ID/LP grammar format is an extension of the ID/LP format (Gazdar, Klein, Pullum and Sag, 1985).
- It is inspired by the Linear Specification Language (Götz and Penn, 1997) and the small LSL fragment used by Suhre (1999).
- It supports:
  - separation of ID and LP
  - linearization across local trees
  - linearization constraints across local trees within the same domain
  - Reape-style compaction of domains, incl. partial compaction
  - LP statements with reference to
    * described domain elements

GIDLP Grammars

- The generalized ID/LP grammar format is an extension of the ID/LP format (Gazdar, Klein, Pullum and Sag, 1985).
- It is inspired by the Linear Specification Language (Götz and Penn, 1997) and the small LSL fragment used by Suhre (1999).
- It supports:
  - separation of ID and LP
  - linearization across local trees
  - linearization constraints across local trees within the same domain
  - Reape-style compaction of domains, incl. partial compaction
  - LP statements with reference to
    * described domain elements
    * specific right-hand-side elements of a rule

GIDLP Grammar Format

- **root declaration**: $root(S)$

GIDLP Grammar Format

- **root declaration**: $root(S)$
- **lexical entries**: $A \rightarrow t$

GIDLP Grammar Format

- **root declaration**: $root(S)$
- **lexical entries**: $A \rightarrow t$
- **grammar rules**: $A \rightarrow \alpha; L$

GIDLP Grammar Format

- **root declaration**: $root(S)$
- **lexical entries**: $A \rightarrow t$
- **grammar rules**: $A \rightarrow \alpha; L$
  - α is a **list of non-terminals**.

GIDLP Grammar Format

- **root declaration**: $root(S)$
- **lexical entries**: $A \rightarrow t$
- **grammar rules**: $A \rightarrow \alpha; L$
  - α is a **list of non-terminals**.
  - L is a set of **linearization constraints**:

GIDLP Grammar Format

- **root declaration**: $root(S)$
- **lexical entries**: $A \rightarrow t$
- **grammar rules**: $A \rightarrow \alpha; L$
  - α is a **list of non-terminals**.
  - L is a set of **linearization constraints**:
    - $A < B$ **(precedence)**: The terminals dominated by A all occur to the left of the terminals dominated by B.

GIDLP Grammar Format

- **root declaration**: $root(S)$
- **lexical entries**: $A \rightarrow t$
- **grammar rules**: $A \rightarrow \alpha; L$
  - α is a **list of non-terminals**.
  - L is a set of **linearization constraints**:
    * $A < B$ **(precedence)**: The terminals dominated by A all occur to the left of the terminals dominated by B.
    * $A \ll B$ **(immediate precedence)**: The rightmost terminal dominated by B occurs immediately to the left of the leftmost terminal dominated by B.

GIDLP Grammar Format

- **root declaration**: $root(S)$
- **lexical entries**: $A \rightarrow t$
- **grammar rules**: $A \rightarrow \alpha; L$
  - α is a **list of non-terminals**.
  - L is a set of **linearization constraints**:
    * $A < B$ **(precedence)**: The terminals dominated by A all occur to the left of the terminals dominated by B.
    * $A \ll B$ **(immediate precedence)**: The rightmost terminal dominated by B occurs immediately to the left of the leftmost terminal dominated by B.
    * $\langle\alpha$, L, A$\rangle$ **(compaction)**: α is a list of tokens, L is a list of domain-level order constraints, and A is the category that results from compacting α.

GIDLP Grammar Format

- **root declaration**: $root(S)$
- **lexical entries**: $A \rightarrow t$
- **grammar rules**: $A \rightarrow \alpha; L$
  - α is a **list of non-terminals**.
  - L is a set of **linearization constraints**:
    - $A < B$ **(precedence)**: The terminals dominated by A all occur to the left of the terminals dominated by B.
    - $A \ll B$ **(immediate precedence)**: The rightmost terminal dominated by B occurs immediately to the left of the leftmost terminal dominated by B.
    - $\langle\alpha$, L, A$\rangle$ **(compaction)**: α is a list of tokens, L is a list of domain-level order constraints, and A is the category that results from compacting α.

Syntactic sugar is introduced for simple compaction specification (A $\rightarrow$ [B] C) and global linearization constraints.

The Two Aspects of Compaction

Compaction has two effects:

The Two Aspects of Compaction

Compaction has two effects:

- **Contiguity**: The terminal yield of a compacted non-terminal contains all and only the terminal yield of the nodes it dominates—there are no holes or additional strings.

The Two Aspects of Compaction

Compaction has two effects:

- **Contiguity**: The terminal yield of a compacted non-terminal contains all and only the terminal yield of the nodes it dominates—there are no holes or additional strings.
- **Isolation**: Precedence statements only constrain the order among elements within a local domain. In other words, precedence constraints can never apply across domains.

A Simple Example

Grammar specification:

root(A)

A $\longrightarrow$ B, [C]
B $\longrightarrow$ F, G
C $\longrightarrow$ D, E

B $<$ D

A Simple Example

Grammar specification:	GIDLP grammar:
root(A)	root(A, $\{B < D\}$)
$A \rightarrow B, [C]$	$A \rightarrow B^1, C^2$; $\{\langle [2], [B < D], C \rangle\}$
$B \rightarrow F, G$	$B \rightarrow F^1, G^2$
$C \rightarrow D, E$	$C \rightarrow D^1, E^2$
$B < D$	

A Simple Example

Grammar specification:

root(A)

A $\rightarrow$ B, [C]
B $\rightarrow$ F, G
C $\rightarrow$ D, E

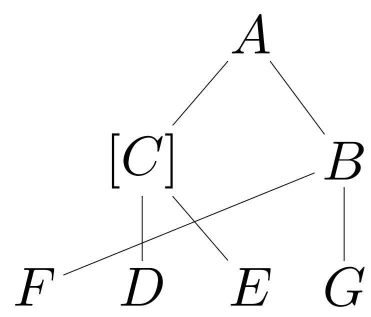
B $<$ D

GIDLP grammar:

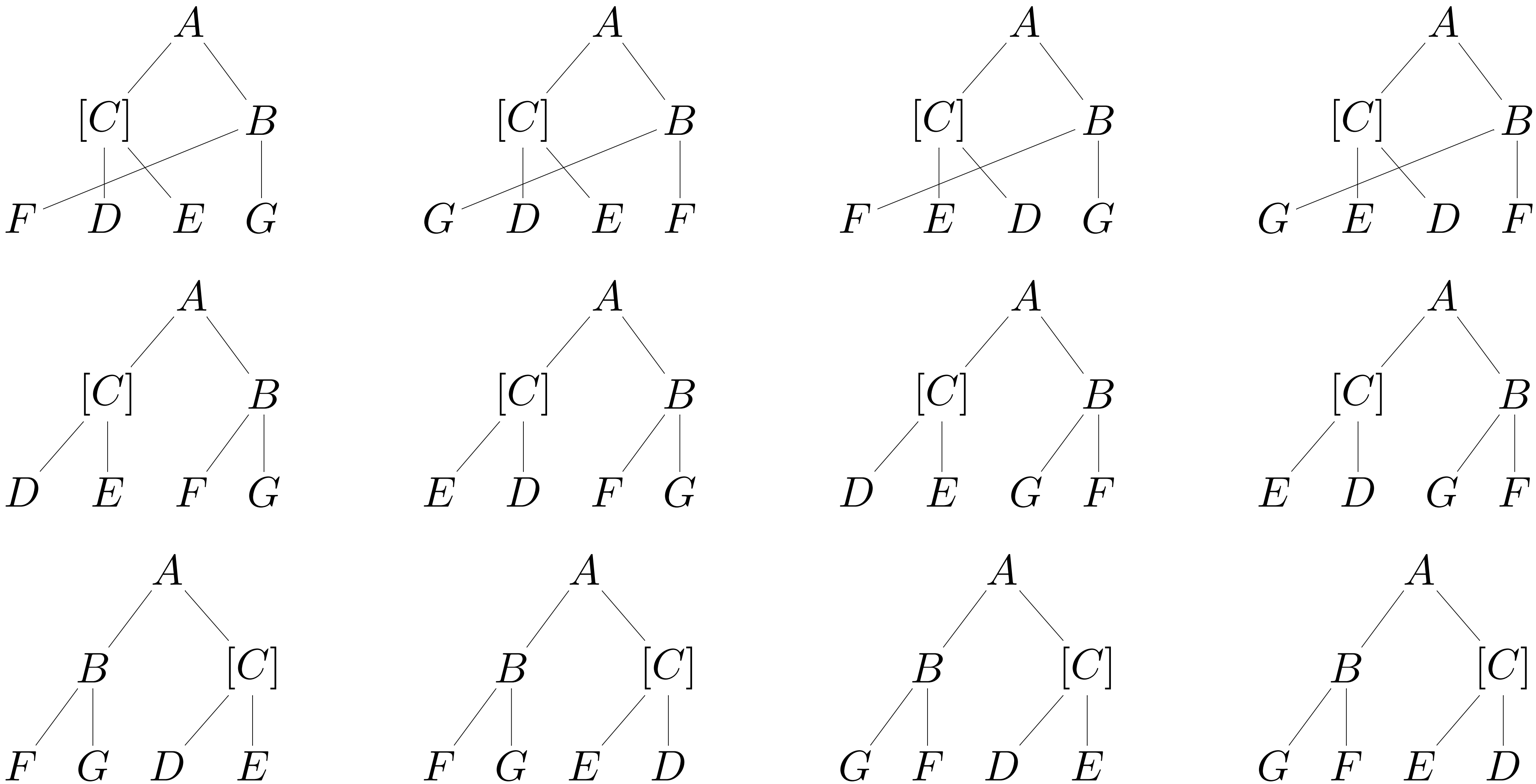
root(A, $\{$B $<$ D$\}$)

A $\rightarrow$ B^1, C^2 ; $\{\langle$[2], [B $<$ D], C$\rangle\}$
B $\rightarrow$ F^1, G^2
C $\rightarrow$ D^1, E^2

Example licensed:



Example Analyses

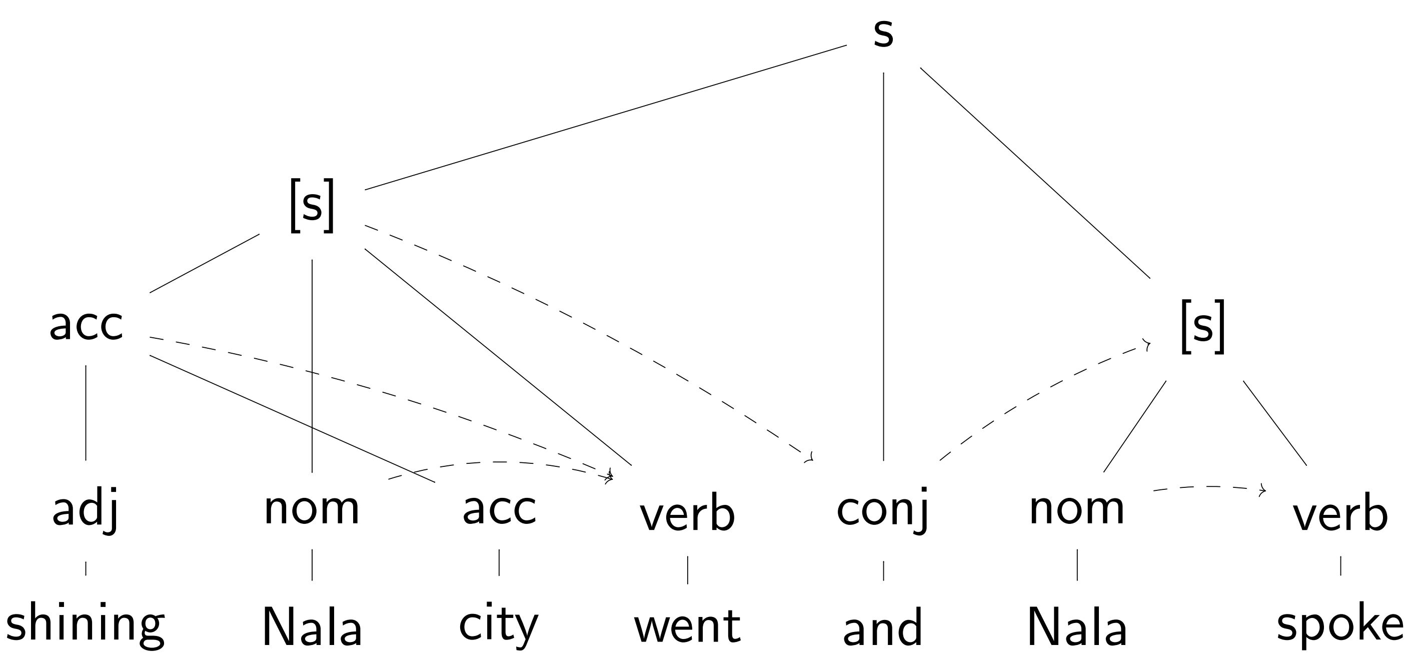


A More Linguistic Example: A (Simplified) Fragment of Sanskrit

1	s $\rightarrow$ verb$_1$ nom$_2$ acc$_3$; $\{2 < 1, 3 < 1\}$	
2	s $\rightarrow$ verb$_1$ nom$_2$; $\{2 < 1\}$	
3	s $\rightarrow$ conj$_1$ s$_2$ s$_3$; $\{2 \ll 1, 1 \ll 3, [2], [3]\}$	
4	acc $\rightarrow$ adj$_1$ acc$_2$; $\{\ \}$	
5	nom $\rightarrow$ *nalas*	‘Nala’ (a proper name)
6	acc $\rightarrow$ *nagaram*	‘city’
7	verb $\rightarrow$ *agacchat*	‘went’
8	conj $\rightarrow$ *caiva*	‘and then’
9	verb $\rightarrow$ *avadat*	‘spoke’
10	adj $\rightarrow$ *ruciram*	‘shining’

(50) रुचिरम् नलस् नगरम् अगच्छत् च नलस् अवदत्
shining Nala city went and Nala spoke
‘Nala went to the shining city and Nala spoke’
(Tokenized from रुचिरं नलो नगरमगच्छश्च नलो ऽवदत्.)

Example Analysis



Notation used:

- solid lines represent dominance
- dashed arrows represent precedence relations enforced by the grammar

Aspects of Parsing with GIDLP Grammars

- Daniels and Meurers (2002) developed an extension of Earley's context-free parsing algorithm (Earley, 1970) to GIDLP grammars.

Aspects of Parsing with GIDLP Grammars

- Daniels and Meurers (2002) developed an extension of Earley's context-free parsing algorithm (Earley, 1970) to GIDLP grammars.
- Aspects to be discussed here:
  - Determining processing order for right-hand side categories in rules

Aspects of Parsing with GIDLP Grammars

- Daniels and Meurers (2002) developed an extension of Earley's context-free parsing algorithm (Earley, 1970) to GIDLP grammars.
- Aspects to be discussed here:
  - Determining processing order for right-hand side categories in rules
  - Bitmasks as compiled linearization constraints on edges

The RHS Order in GIDLP Rules

- What function does the RHS of GIDLP rules play given that it is not used to express constituent order? $\Rightarrow$ Determines order in which categories are searched for.

The RHS Order in GIDLP Rules

- What function does the RHS of GIDLP rules play given that it is not used to express constituent order? $\Rightarrow$ Determines order in which categories are searched for.

  Examples from the Sanskrit grammar:

  · s $\rightarrow$ verb$_1$ nom$_2$ acc$_3$; $\{2 < 1, 3 < 1\}$
    requires the verb be parsed first, even though linearly it's the last element.

The RHS Order in GIDLP Rules

- What function does the RHS of GIDLP rules play given that it is not used to express constituent order? ⇒ Determines order in which categories are searched for.

  Examples from the Sanskrit grammar:

  · s → verb_1 nom_2 acc_3; $\{2 < 1,\ 3 < 1\}$
  requires the verb be parsed first, even though linearly it's the last element.

  · s → conj_1 s_2 s_3; $\{2 \ll 1, 1 \ll 3, [2], [3]\}$

The RHS Order in GIDLP Rules

- What function does the RHS of GIDLP rules play given that it is not used to express constituent order? $\Rightarrow$ Determines order in which categories are searched for.

  Examples from the Sanskrit grammar:

  · s $\rightarrow$ verb$_1$ nom$_2$ acc$_3$; $\{2 < 1,\ 3 < 1\}$
    requires the verb be parsed first, even though linearly it's the last element.

  · s $\rightarrow$ conj$_1$ s$_2$ s$_3$; $\{2 \ll 1, 1 \ll 3, [2], [3]\}$

- How should the RHS symbols be ordered? To first search for those categories which are easy to identify and convey information to guide the parsing process.

The RHS Order in GIDLP Rules

- What function does the RHS of GIDLP rules play given that it is not used to express constituent order? $\Rightarrow$ Determines order in which categories are searched for.

  Examples from the Sanskrit grammar:

  - s $\rightarrow$ verb$_1$ nom$_2$ acc$_3$; $\{2 < 1,\ 3 < 1\}$
    requires the verb be parsed first, even though linearly it's the last element.
  - s $\rightarrow$ conj$_1$ s$_2$ s$_3$; $\{2 \ll 1, 1 \ll 3, [2], [3]\}$

- How should the RHS symbols be ordered? To first search for those categories which are easy to identify and convey information to guide the parsing process.

  - acc $\rightarrow$ adj$_1$ acc$_2$ only enters recursion when adjective is found first

The RHS Order in GIDLP Rules

- What function does the RHS of GIDLP rules play given that it is not used to express constituent order? $\Rightarrow$ Determines order in which categories are searched for.

  Examples from the Sanskrit grammar:

  · s $\rightarrow$ verb$_1$ nom$_2$ acc$_3$; $\{2 < 1,\ 3 < 1\}$
  requires the verb be parsed first, even though linearly it's the last element.

  · s $\rightarrow$ conj$_1$ s$_2$ s$_3$; $\{2 \ll 1, 1 \ll 3, [2], [3]\}$

- How should the RHS symbols be ordered? To first search for those categories which are easy to identify and convey information to guide the parsing process.

  · acc $\rightarrow$ adj$_1$ acc$_2$ only enters recursion when adjective is found first

- Extension of the binary head/non-head distinction in Head-Driven Parsing (Kay, 1990; van Noord, 1997): All daughters are ordered.

Is Ordering all Daughters Useful? Raising-to-Object in English

Raising-to-Object verbs like *expect* can select a variety of predicates:

Is Ordering all Daughters Useful? Raising-to-Object in English

Raising-to-Object verbs like *expect* can select a variety of predicates:

- *leave* selects an NP argument:

  (51) I expect him [to leave].

Is Ordering all Daughters Useful? Raising-to-Object in English

Raising-to-Object verbs like *expect* can select a variety of predicates:

- *leave* selects an NP argument:

  (51) I expect him [to leave].

- *be trouble* selects an expletive subject:

  (52) I expect there [to be trouble in Iraq].

Is Ordering all Daughters Useful? Raising-to-Object in English

Raising-to-Object verbs like *expect* can select a variety of predicates:

- *leave* selects an NP argument:

  (51) I expect him [to leave].

- *be trouble* selects an expletive subject:

  (52) I expect there [to be trouble in Iraq].

- *bother* selects a sentential subject:

  (53) I expect [that there is a fly in the soup] [to bother him].

Is Ordering all Daughters Useful? Raising-to-Object in English

Raising-to-Object verbs like *expect* can select a variety of predicates:

- *leave* selects an NP argument:

  (51) I expect him [to leave].

- *be trouble* selects an expletive subject:

  (52) I expect there [to be trouble in Iraq].

- *bother* selects a sentential subject:

  (53) I expect [that there is a fly in the soup] [to bother him].

When parsing a (ternary) VP headed by a subject-to-object raising verb, the verbal complement should be considered before the raised element.

Is Ordering all Daughters Useful? Raising-to-Object in English

Raising-to-Object verbs like *expect* can select a variety of predicates:

- *leave* selects an NP argument:

  (51) I expect^1 him^3 [to leave]^2.

- *be trouble* selects an expletive subject:

  (52) I expect^1 there^3 [to be trouble in Iraq]^2.

- *bother* selects a sentential subject:

  (53) I expect^1 [that there is a fly in the soup]^3 [to bother him]^2.

When parsing a (ternary) VP headed by a subject-to-object raising verb, the verbal complement should be considered before the raised element.

Is ordering all daughters useful? Quirky Case in Icelandic

- Many verbs in Icelandic assign “quirky case” (i.e. non-nominative) to their subjects.

Is ordering all daughters useful? Quirky Case in Icelandic

- Many verbs in Icelandic assign “quirky case” (i.e. non-nominative) to their subjects.
- The case assignments persist when the subject is raised to be the subject or object of a matrix verb (cf., e.g., Andrews, 1982).

Is ordering all daughters useful? Quirky Case in Icelandic

- Many verbs in Icelandic assign "quirky case" (i.e. non-nominative) to their subjects.
- The case assignments persist when the subject is raised to be the subject or object of a matrix verb (cf., e.g., Andrews, 1982).
- From a parsing perspective, the embedded verb must be known before it can be determined whether a given noun phrase is an acceptable subject for the matrix verb.

Icelandic Data

Subjects show ordinary agreement with matrix verbs, but are marked

- accusative

(54) Hana vird. h.ist vanta peninga
her$_{acc}$ seems to-lack money
'She seems to lack money.'

Icelandic Data

Subjects show ordinary agreement with matrix verbs, but are marked

- accusative

(54)

Hana	vird. h.ist	vanta	peninga
her$_{acc}$	seems	to-lack	money

'She seems to lack money.'

- dative

(55)

Barninu	vird. h.ist	hafa	batnad. h.	veikin
the-child$_{dat}$	seems	to-have	recovered-from	the-disease

'The child seems to have recovered from the disease.'

Icelandic Data

Subjects show ordinary agreement with matrix verbs, but are marked

- accusative

(54) Hana vird. h.ist vanta peninga
her$_{acc}$ seems to-lack money
'She seems to lack money.'

- dative

(55) Barninu vird. h.ist hafa batnad. h. veikin
the-child$_{dat}$ seems to-have recovered-from the-disease
'The child seems to have recovered from the disease.'

- genitive

(56) Verkjanna vird. h.ist ekki gæta
the-pains$_{gen}$ seem not to-be-noticeable
'The pains don't seem to be noticeable.'

Raising-to-Object in Icelandic

(57) Hann telur mig vanta peninga
he.NOM believes me$_{acc}$ to-lack money
'He believes that I lack money.'

Raising-to-Object in Icelandic

(57)

Hann	telur	mig	vanta	peninga
he.NOM	believes	me$_{acc}$	to-lack	money

'He believes that I lack money.'

(58)

Hann	telur	barninu	hafa	batnad. h.	veikin
he	believes	the-child$_{dat}$	to-have	recovered-from	the-disease

'He believes the child to have recovered from the disease.'

Raising-to-Object in Icelandic

(57)

Hann	telur	mig	vanta	peninga
he.NOM	believes	me$_{acc}$	to-lack	money

'He believes that I lack money.'

(58)

Hann	telur	barninu	hafa	batnad. h.	veikin
he	believes	the-child$_{dat}$	to-have	recovered-from	the-disease

'He believes the child to have recovered from the disease.'

(59)

Hann	telur	verkjanna	ekki	gæta
he	believes	the-pains$_{gen}$	not	to-be-noticeable

'He believes the pains to be not noticeable.'

Bitmasks as Compiled Linearization Constraints on Edges

- We use bitvectors to encode constraints on coverage; bitvectors used this way are called **bitmasks**.

Bitmasks as Compiled Linearization Constraints on Edges

- We use bitvectors to encode constraints on coverage; bitvectors used this way are called **bitmasks**.
- We compute and store a negative and positive bitmask on each edge, representing different aspects of precedence information.

Bitmasks as Compiled Linearization Constraints on Edges

- We use bitvectors to encode constraints on coverage; bitvectors used this way are called **bitmasks**.

- We compute and store a negative and positive bitmask on each edge, representing different aspects of precedence information.

- Assume we've found a noun as the third word of a sentence:

Bitmasks as Compiled Linearization Constraints on Edges

- We use bitvectors to encode constraints on coverage; bitvectors used this way are called **bitmasks**.
- We compute and store a negative and positive bitmask on each edge, representing different aspects of precedence information.
- Assume we've found a noun as the third word of a sentence:
  - Negative: If we know that verbs follow nouns, we shouldn't look for verbs in the first or second position.

Bitmasks as Compiled Linearization Constraints on Edges

- We use bitvectors to encode constraints on coverage; bitvectors used this way are called **bitmasks**.
- We compute and store a negative and positive bitmask on each edge, representing different aspects of precedence information.
- Assume we've found a noun as the third word of a sentence:
  - Negative: If we know that verbs follow nouns, we shouldn't look for verbs in the first or second position.
  - Positive: If we know that verbs immediately follow nouns, then any verb we find must cover at least the fourth position.

Efficient Computation with Bitvectors

- Parsing requires a number of efficiently implemented bitwise operations: overlap, lbound, suffix, etc.

Efficient Computation with Bitvectors

- Parsing requires a number of efficiently implemented bitwise operations: overlap, lbound, suffix, etc.
- Operations on any list-based representation will take time proportional to the length of the list itself.

Efficient Computation with Bitvectors

- Parsing requires a number of efficiently implemented bitwise operations: overlap, lbound, suffix, etc.
- Operations on any list-based representation will take time proportional to the length of the list itself.
- Bitwise operations on bitvectors represented as numbers can be computed in parallel in a single processor instruction. For example, it takes just as long to compute the bitwise-AND of two 20-bit numbers as it does for two 30-bit numbers.

Efficient Computation with Bitvectors

- Parsing requires a number of efficiently implemented bitwise operations: overlap, lbound, suffix, etc.

- Operations on any list-based representation will take time proportional to the length of the list itself.

- Bitwise operations on bitvectors represented as numbers can be computed in parallel in a single processor instruction. For example, it takes just as long to compute the bitwise-AND of two 20-bit numbers as it does for two 30-bit numbers.

- In general, for a processor that has a word size of n, arithmetic computation time is proportional to the base-n logarithm of the inputs.

Useful Bitvector Operations

- $\textsc{singleton}(p)$: Gives the bitvector in which only p is occupied.
  Computed as 2^p
  Example: singleton(1) is 00010.

- $\textsc{overlap}(x, y)$: Is there any position occupied in both x and y?
  Computed as $\textsc{and}(x, y) \neq 0$
  Example: 10111 and 01010 have a bitwise-AND of 00010, and therefore overlap.

- $\textsc{combine}(x, y)$: Gives the union of x and y.
  Computed as $\textsc{or}(x, y)$
  Example: 10110 and 01010 combine to form 11110.

- $\textsc{rbound}(x)$: Most-significant occupied bit in x.
  Computed as $\lfloor \log_2(x) \rfloor$
  Example: the right bound of 01010 is 3, which is $\lfloor 3.32 \rfloor$.

- $\textsc{lbound}(x)$: Least-significant occupied bit in x.
  Computed as $\textsc{rbound}(\textsc{xor}(x, x-1))$
  Example: the left bound of 01010 is 1, which is the right bound of 00011 $=$ XOR(01010, 01001).

- PREFIX(p): Gives the bitvector covering all positions $\leq p$.
  Computed as $2^{p+1} - 1$
  Example: PREFIX(3) is 01111 (15).
- SUFFIX(p): Gives the bitvector covering all positions $\geq p$.
  Computed as NOT$(2^x - 1)$
  Example: SUFFIX(3) is NOT(00111) $=$ 11000.
- PRECEDE(x, y): Does x completely precede y (where x and y are assumed not to overlap)?
  Equivalent to numeric $x < y$
  Example: 00011 precedes 01100 tested through $3 < 12$.
- IPRECEDE(x, y): Does x immediately precede y (where x and y are assumed not to overlap)?
  Equivalent to $\text{RBOUND}(x) = \text{LBOUND}(y) - 1$
  Example: the right bound of 00011 is 1, and the left bound of 01100 is 2, so 00011 immediately precedes 01100.
- ISOLATED(x): Does x form a continuous unit?
  Equivalent to $x = \text{AND}(\text{PREFIX}(\text{RBOUND}(x)), \text{SUFFIX}(\text{LBOUND}(x)))$
  Example: With the vector 01101, AND(01111, 11111) is $01111 \neq 01101$, so 01101 is not isolated.

Outline

- Phrase Structure
- A Topological Model for German
- From Phrase Structure to GPSG and HPSG
- German Word Order Phenomena and their Analyses in Standard-HPSG/GPSG
- Discontinuous Constituents
- Parsing
- Syntactically Annotated Corpora and Discontinuity
- Summary

Syntactic Annotation for German

- German Verb*mobil* treebank (Hinrichs et. al., 2000; Stegmann et. al., 2000):

Syntactic Annotation for German

- German Verb*mobil* treebank (Hinrichs et. al., 2000; Stegmann et. al., 2000):
  - spoken language: dialogs in which two discourse participants negotiate business appointments.

Syntactic Annotation for German

- German Verb*mobil* treebank (Hinrichs et. al., 2000; Stegmann et. al., 2000):
  - spoken language: dialogs in which two discourse participants negotiate business appointments.
  - 38.000 syntactic units (dialog turns)
  - flat structures based on topological fields

Syntactic Annotation for German

- German Verb*mobil* treebank (Hinrichs et. al., 2000; Stegmann et. al., 2000):
  - spoken language: dialogs in which two discourse participants negotiate business appointments.
  - 38.000 syntactic units (dialog turns)
  - flat structures based on topological fields

- Negra Treebank (Skut et. al., 1997, 1998)

Syntactic Annotation for German

- German Verb*mobil* treebank (Hinrichs et. al., 2000; Stegmann et. al., 2000):
  - spoken language: dialogs in which two discourse participants negotiate business appointments.
  - 38.000 syntactic units (dialog turns)
  - flat structures based on topological fields
- Negra Treebank (Skut et. al., 1997, 1998)
  - written language: *Frankfurter Rundschau*, a national newspaper

Syntactic Annotation for German

- German Verb*mobil* treebank (Hinrichs et. al., 2000; Stegmann et. al., 2000):
  - spoken language: dialogs in which two discourse participants negotiate business appointments.
  - 38.000 syntactic units (dialog turns)
  - flat structures based on topological fields
- Negra Treebank (Skut et. al., 1997, 1998)
  - written language: *Frankfurter Rundschau*, a national newspaper
  - 20.000 sentences (350.000 tokens)

Syntactic Annotation for German

- German Verb*mobil* treebank (Hinrichs et. al., 2000; Stegmann et. al., 2000):
  - spoken language: dialogs in which two discourse participants negotiate business appointments.
  - 38.000 syntactic units (dialog turns)
  - flat structures based on topological fields
- Negra Treebank (Skut et. al., 1997, 1998)
  - written language: *Frankfurter Rundschau*, a national newspaper
  - 20.000 sentences (350.000 tokens)
    TIGER Treebank (Brants et. al., 2002): $>$ 35.000 sentences (with refined annotation)
  - flat structures as encoding of argument structure

Outline

- Phrase Structure
- A Topological Model for German
- From Phrase Structure to GPSG and HPSG
- German Word Order Phenomena and their Analyses in Standard-HPSG/GPSG
- Discontinuous Constituents
- Parsing
- Syntactically Annotated Corpora and Discontinuity
  - The German Verb*mobil* Treebank
  - The NEGRA Treebank
- Summary

The German Verb*mobil* Treebank

- annotation consists of tree structures with node and edge labels

The German Verb*mobil* Treebank

- annotation consists of tree structures with node and edge labels
- tree structure:
  - encodes
    - topological field structure at top-level

The German Verb*mobil* Treebank

- annotation consists of tree structures with node and edge labels
- tree structure:
  - encodes
    - topological field structure at top-level
    - syntactic categories

The German Verb*mobil* Treebank

- annotation consists of tree structures with node and edge labels
- tree structure:
  - encodes
    - topological field structure at top-level
    - syntactic categories
  - properties:
    - no branching edges and each daughter has one mother (some secondary edges)

The German Verb*mobil* Treebank

- annotation consists of tree structures with node and edge labels
- tree structure:
  - encodes
    - topological field structure at top-level
    - syntactic categories
  - properties:
    - no branching edges and each daughter has one mother (some secondary edges)
    - no empty terminal nodes

The German Verb*mobil* Treebank

- annotation consists of tree structures with node and edge labels
- tree structure:
  - encodes
    - topological field structure at top-level
    - syntactic categories
  - properties:
    - no branching edges and each daughter has one mother (some secondary edges)
    - no empty terminal nodes
- node and edge labels encode:
  - node labels

The German Verb*mobil* Treebank

- annotation consists of tree structures with node and edge labels
- tree structure:
  - encodes
    - topological field structure at top-level
    - syntactic categories
  - properties:
    - no branching edges and each daughter has one mother (some secondary edges)
    - no empty terminal nodes
- node and edge labels encode:
  - node labels
    - sentence level: turn type

The German Verb*mobil* Treebank

- annotation consists of tree structures with node and edge labels
- tree structure:
  - encodes
    - topological field structure at top-level
    - syntactic categories
  - properties:
    - no branching edges and each daughter has one mother (some secondary edges)
    - no empty terminal nodes
- node and edge labels encode:
  - node labels
    - sentence level: turn type
    - field level: topological field names

The German Verb*mobil* Treebank

- annotation consists of tree structures with node and edge labels
- tree structure:
  - encodes
    - topological field structure at top-level
    - syntactic categories
  - properties:
    - no branching edges and each daughter has one mother (some secondary edges)
    - no empty terminal nodes
- node and edge labels encode:
  - node labels
    - sentence level: turn type
    - field level: topological field names
    - phrase level: syntactic categories

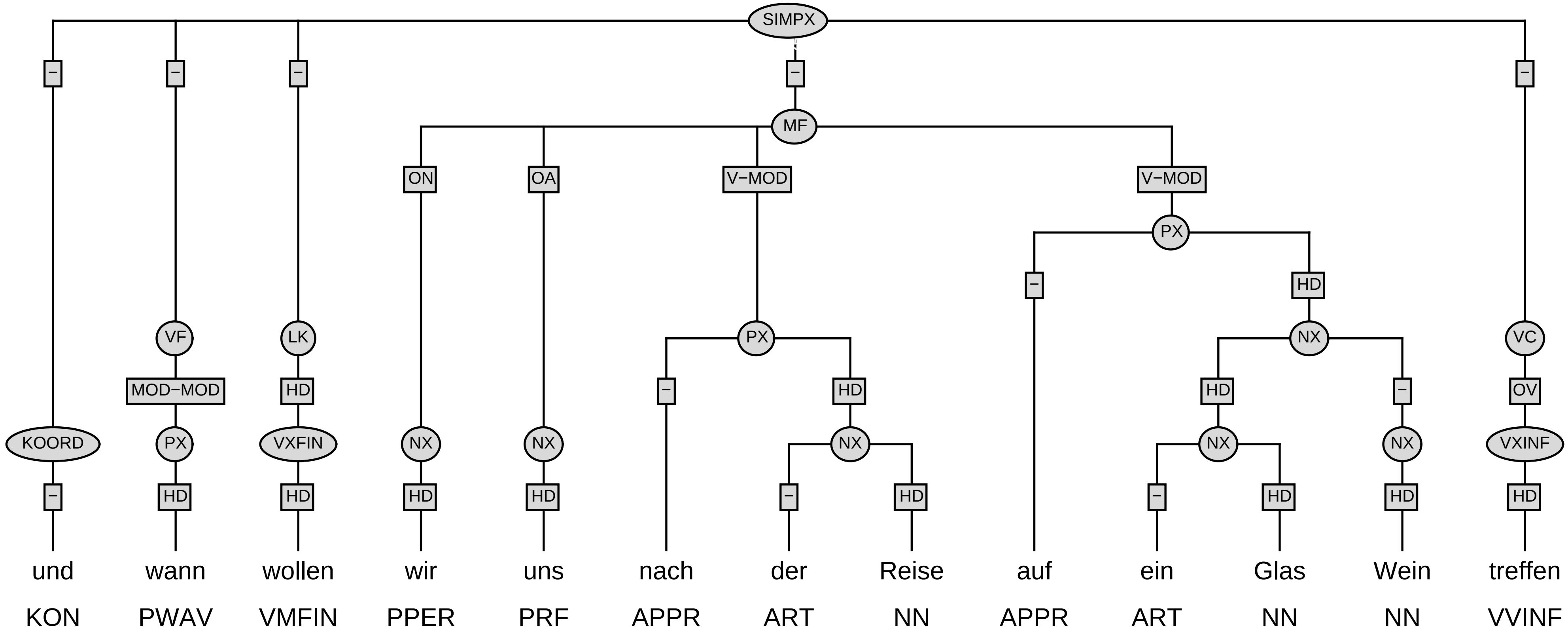
The German Verb*mobil* Treebank

- annotation consists of tree structures with node and edge labels
- tree structure:
  - encodes
    - topological field structure at top-level
    - syntactic categories
  - properties:
    - no branching edges and each daughter has one mother (some secondary edges)
    - no empty terminal nodes
- node and edge labels encode:
  - node labels
    - sentence level: turn type
    - field level: topological field names
    - phrase level: syntactic categories
    - lexical level: STTS part-of-speech (Schiller, Teufel and Thielen, 1995)

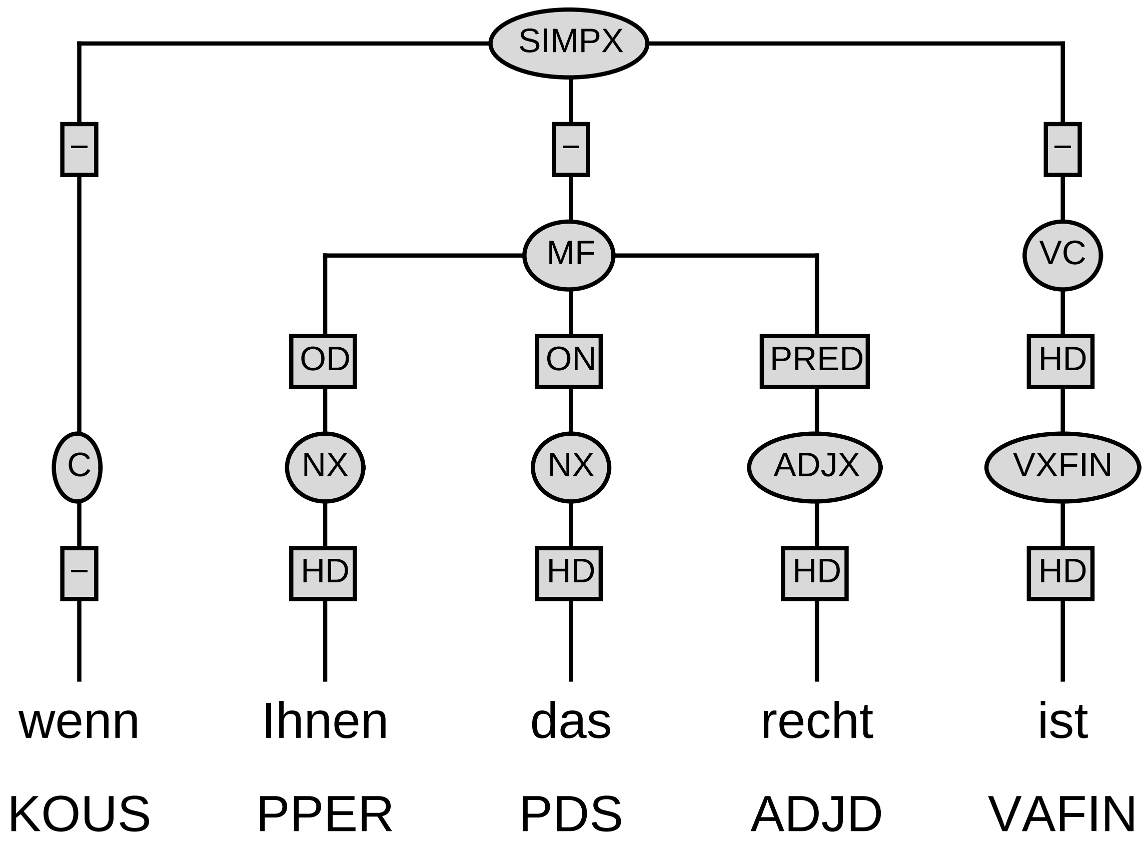
The German Verb*mobil* Treebank

- annotation consists of tree structures with node and edge labels
- tree structure:
  - encodes
    - topological field structure at top-level
    - syntactic categories
  - properties:
    - no branching edges and each daughter has one mother (some secondary edges)
    - no empty terminal nodes
- node and edge labels encode:
  - node labels
    - sentence level: turn type
    - field level: topological field names
    - phrase level: syntactic categories
    - lexical level: STTS part-of-speech (Schiller, Teufel and Thielen, 1995)
  - edge labels on phrase level: grammatical functions

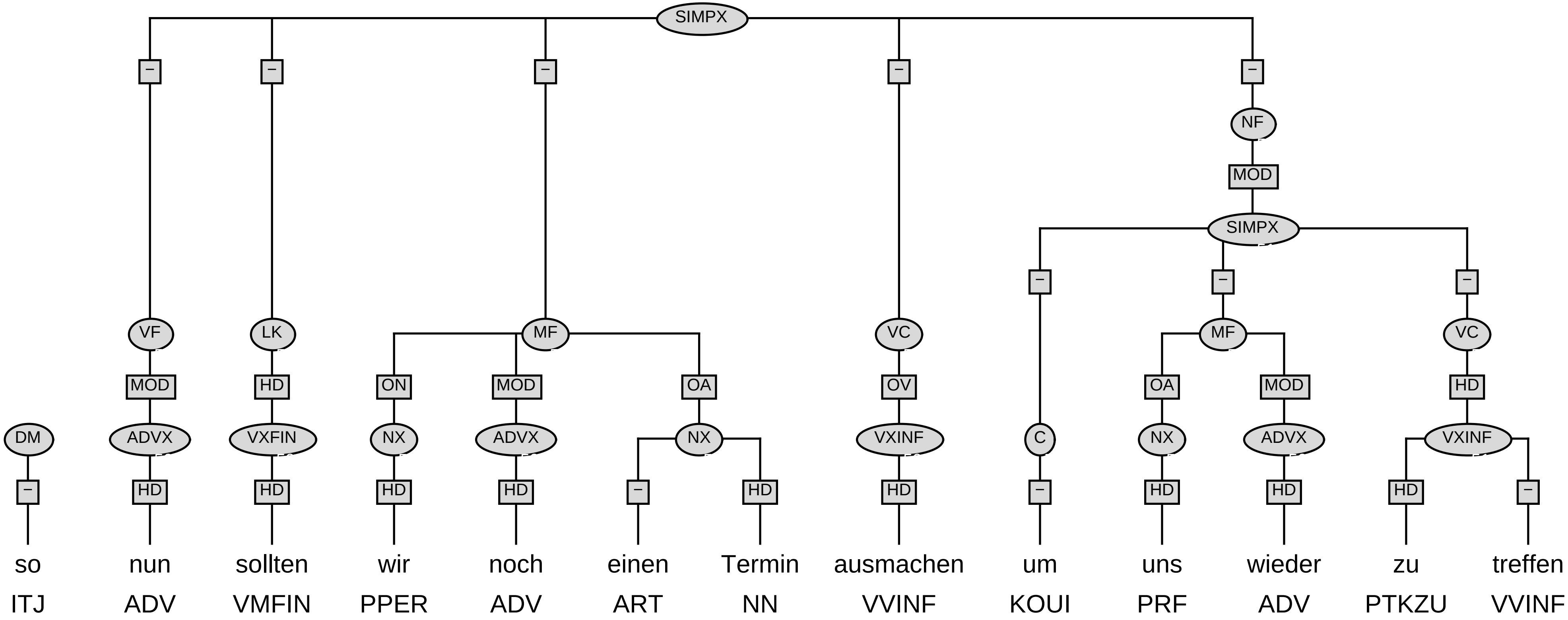
A Basic Example



A Verb-Last Example



An Example with Embedding



Dependencies across Categories/Fields

- remember: no crossing branches

Dependencies across Categories/Fields

- remember: no crossing branches
- where dependency relations cross the border between constituents or topological fields, reference is encoded by special naming conventions for edge labels

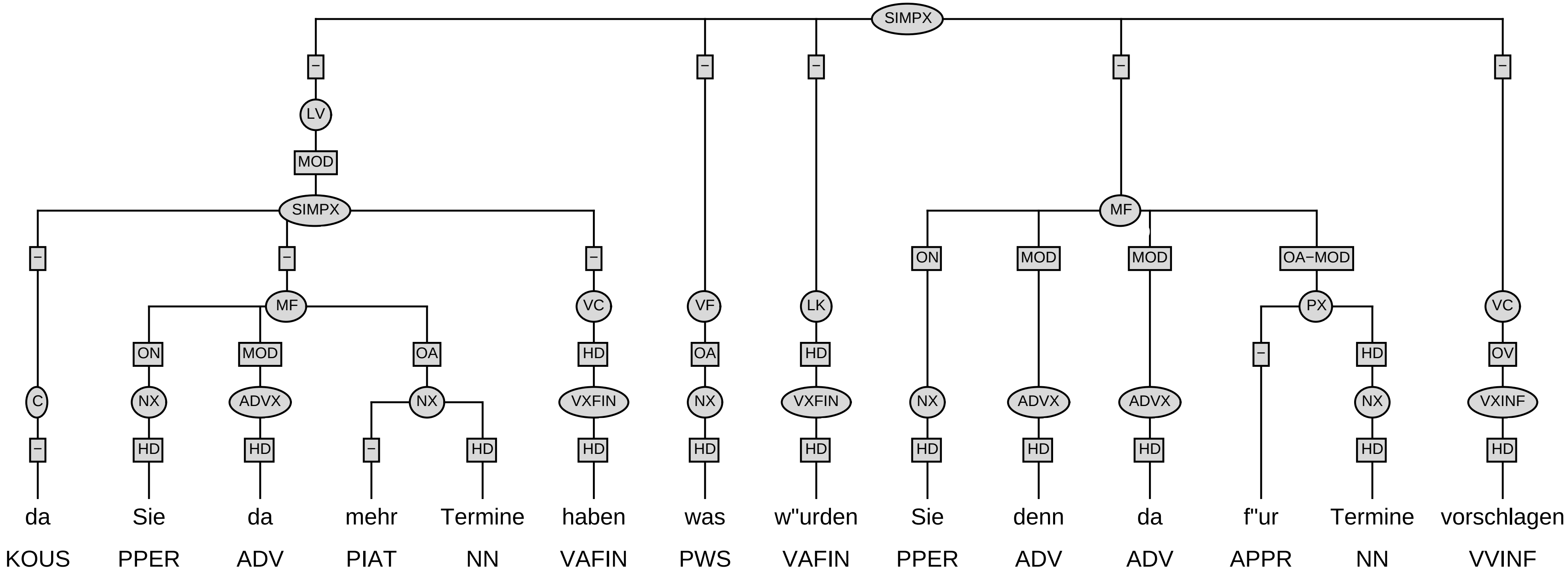
Dependencies across Categories/Fields

- remember: no crossing branches
- where dependency relations cross the border between constituents or topological fields, reference is encoded by special naming conventions for edge labels
- examples:
  - OA-MOD is a modifier of an OA occurring somewhere in the sentence

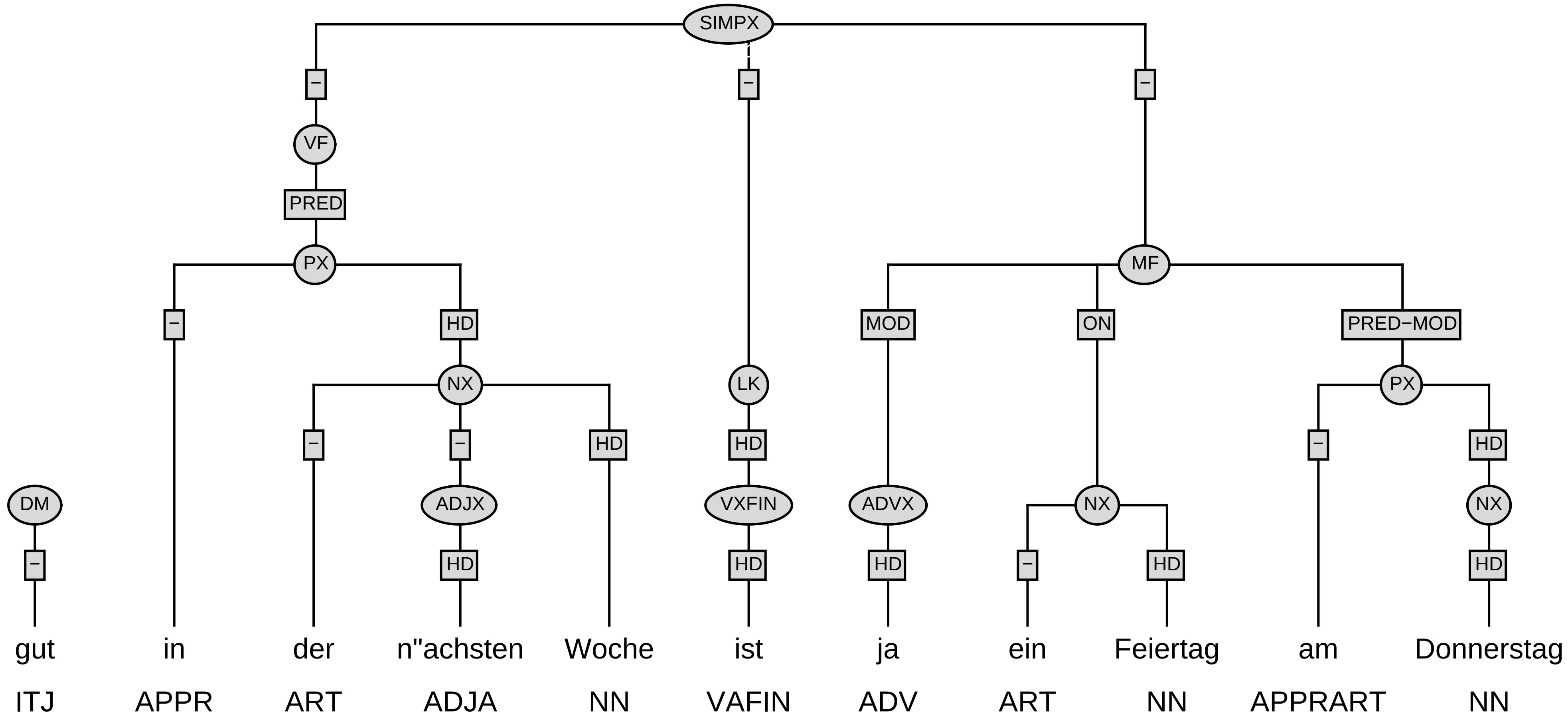
Dependencies across Categories/Fields

- remember: no crossing branches
- where dependency relations cross the border between constituents or topological fields, reference is encoded by special naming conventions for edge labels
- examples:
  - OA-MOD is a modifier of an OA occurring somewhere in the sentence
  - PRED-MOD is a modifier of a PRED

OA-MOD Example



PRED-MOD Example



Position of Finite Verb and the Verbal Complex

- finite verb is annotated in the left bracket or verbal cluster, following surface order

Position of Finite Verb and the Verbal Complex

- finite verb is annotated in the left bracket or verbal cluster, following surface order
- verbal complex is flat

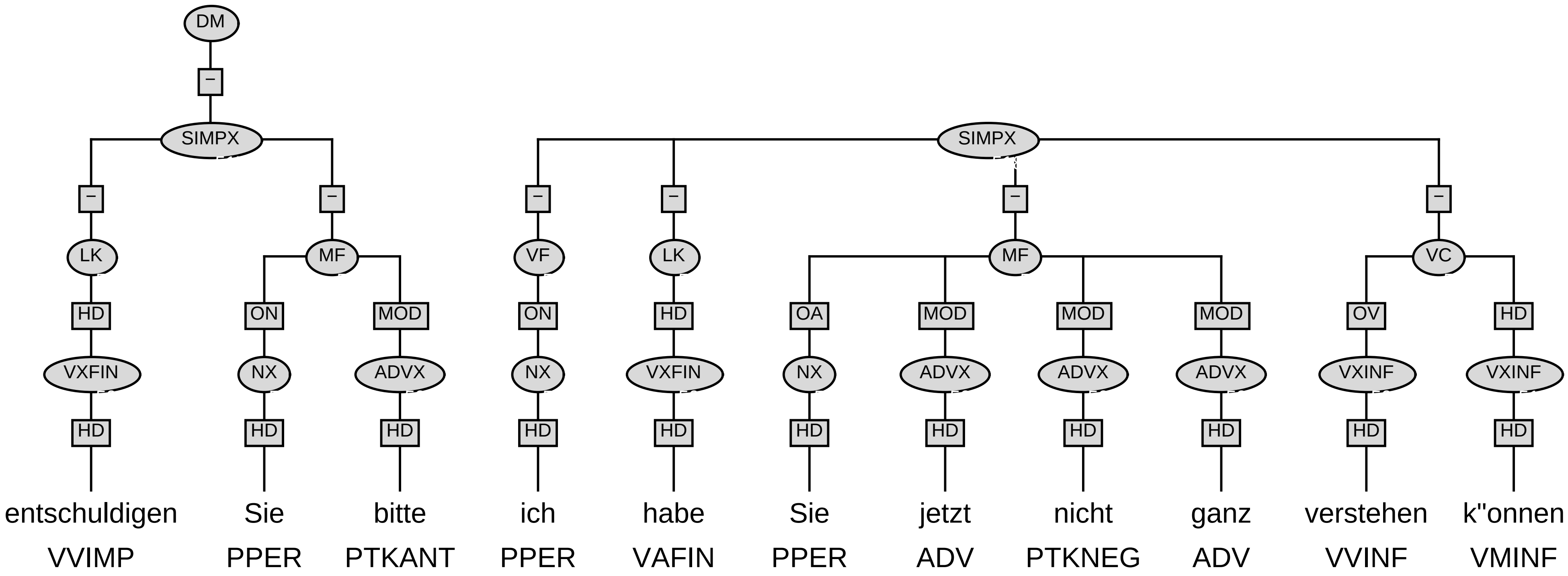
Position of Finite Verb and the Verbal Complex

- finite verb is annotated in the left bracket or verbal cluster, following surface order
- verbal complex is flat
- secondary edges mark relations in a verbal complex with more than two verbs

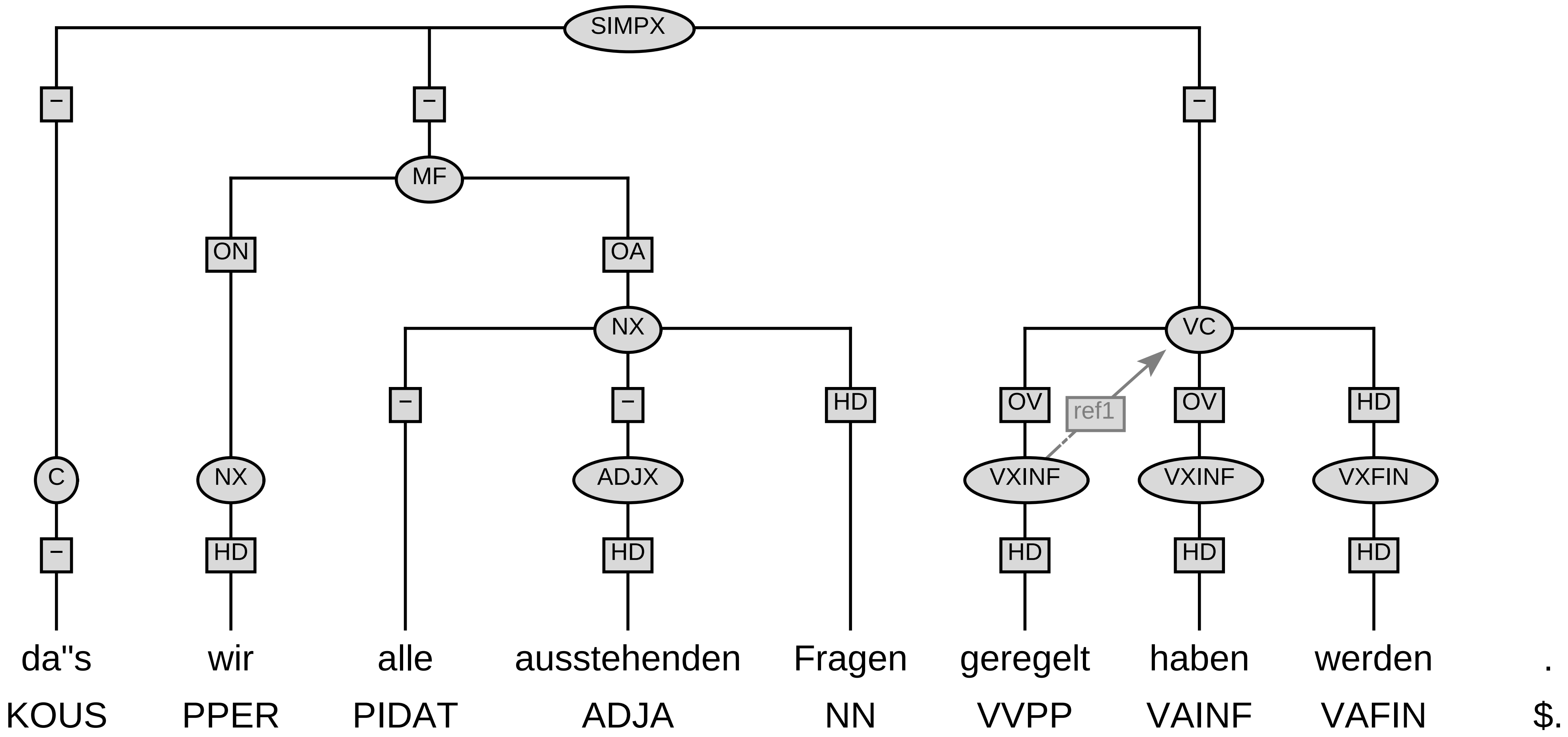
Position of Finite Verb and the Verbal Complex

- finite verb is annotated in the left bracket or verbal cluster, following surface order
- verbal complex is flat
- secondary edges mark relations in a verbal complex with more than two verbs
- relations between arguments and different verbs in a cluster not distinguished (hardly any complex non-finite constructions in Verb*mobil* data)

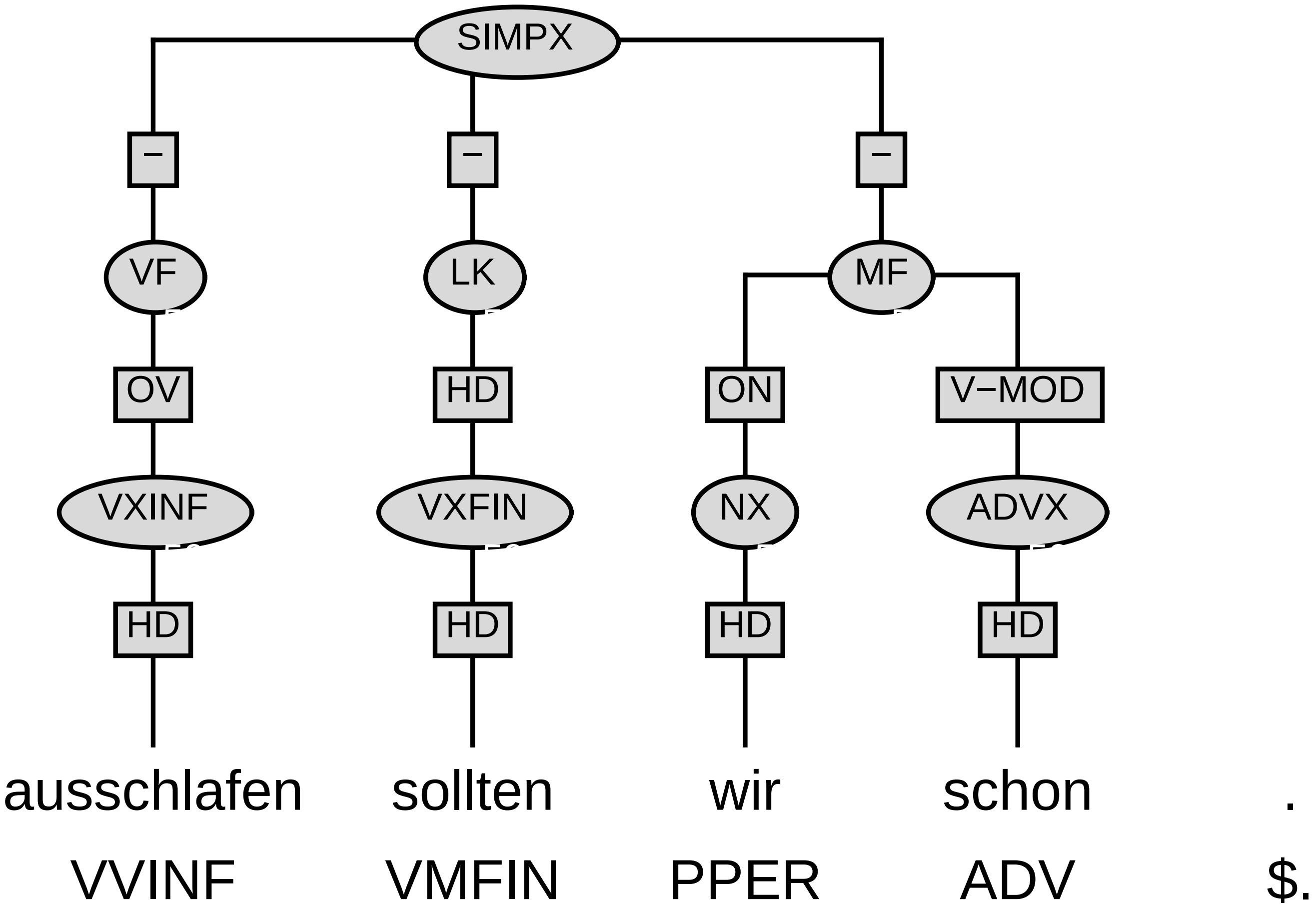
Verbal Complex with Two Verbs



Verbal Complex with Three Verbs



Fronting of a Non-Finite Verb



Outline

- Phrase Structure
- A Topological Model for German
- From Phrase Structure to GPSG and HPSG
- German Word Order Phenomena and their Analyses in Standard-HPSG/GPSG
- Discontinuous Constituents
- Parsing
- Syntactically Annotated Corpora and Discontinuity
  - The German Verb*mobil* Treebank
  - The NEGRA Treebank
- Summary

The NEGRA Treebank

- annotation consists of tree structures with node and edge labels

The NEGRA Treebank

- annotation consists of tree structures with node and edge labels
- tree structure:
  - encodes argument structure

The NEGRA Treebank

- annotation consists of tree structures with node and edge labels
- tree structure:
  - encodes argument structure
  - properties:
    - branching edges used extensively

The NEGRA Treebank

- annotation consists of tree structures with node and edge labels
- tree structure:
  - encodes argument structure
  - properties:
    * branching edges used extensively
    * no empty terminal nodes

The NEGRA Treebank

- annotation consists of tree structures with node and edge labels
- tree structure:
  - encodes argument structure
  - properties:
    * branching edges used extensively
    * no empty terminal nodes
    * each daughter has one mother (but some secondary edges)
- node and edge labels encode:

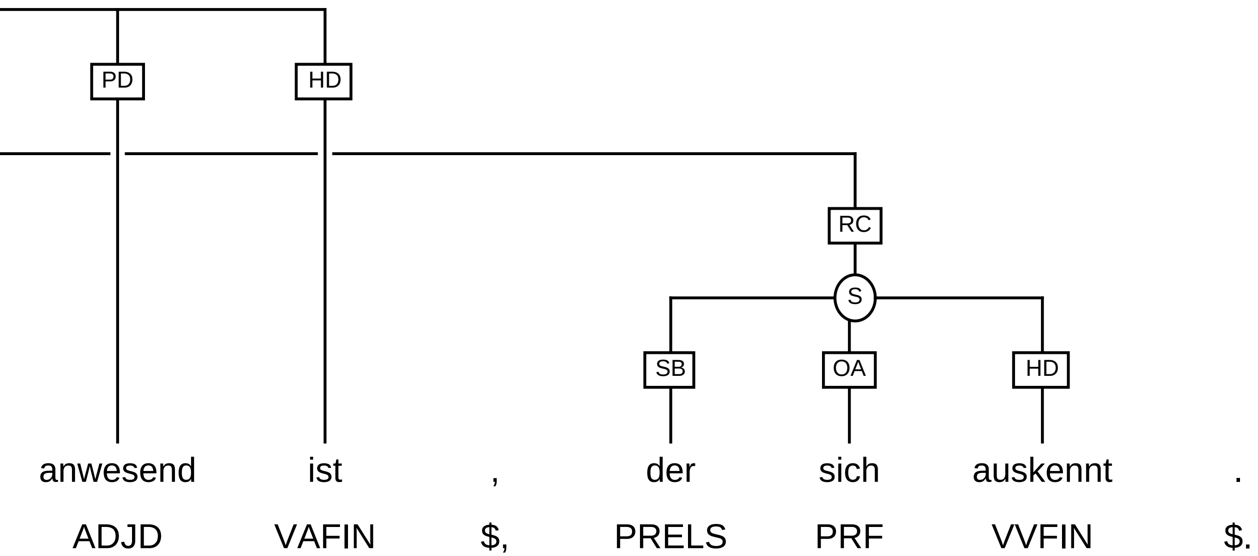
The NEGRA Treebank

- annotation consists of tree structures with node and edge labels
- tree structure:
  - encodes argument structure
  - properties:
    * branching edges used extensively
    * no empty terminal nodes
    * each daughter has one mother (but some secondary edges)
- node and edge labels encode:
  - phrase level: syntactic categories

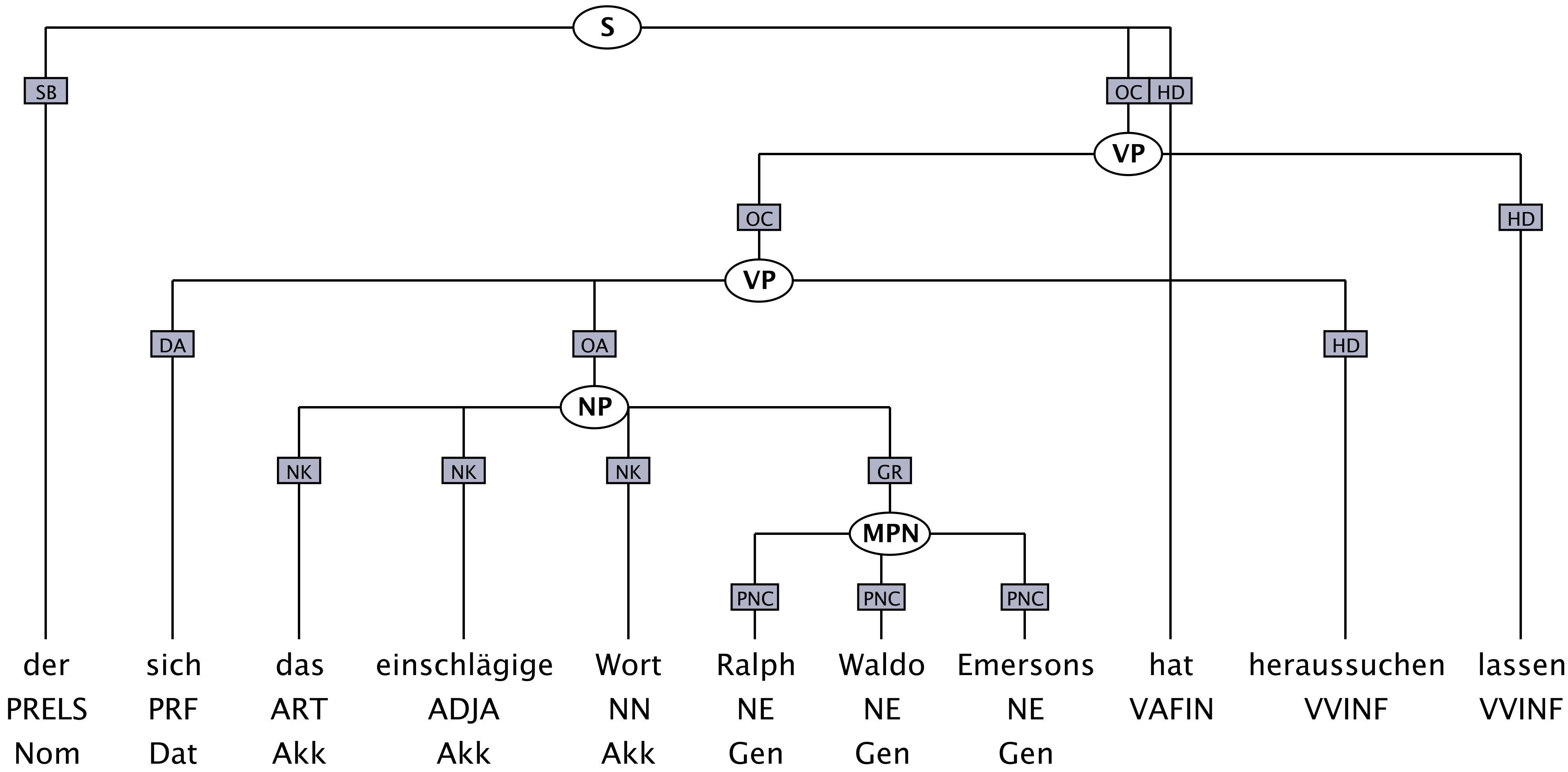
The NEGRA Treebank

- annotation consists of tree structures with node and edge labels
- tree structure:
  - encodes argument structure
  - properties:
    * branching edges used extensively
    * no empty terminal nodes
    * each daughter has one mother (but some secondary edges)
- node and edge labels encode:
  - phrase level: syntactic categories
  - lexical level: STTS part-of-speech (Schiller, Teufel and Thielen, 1995)

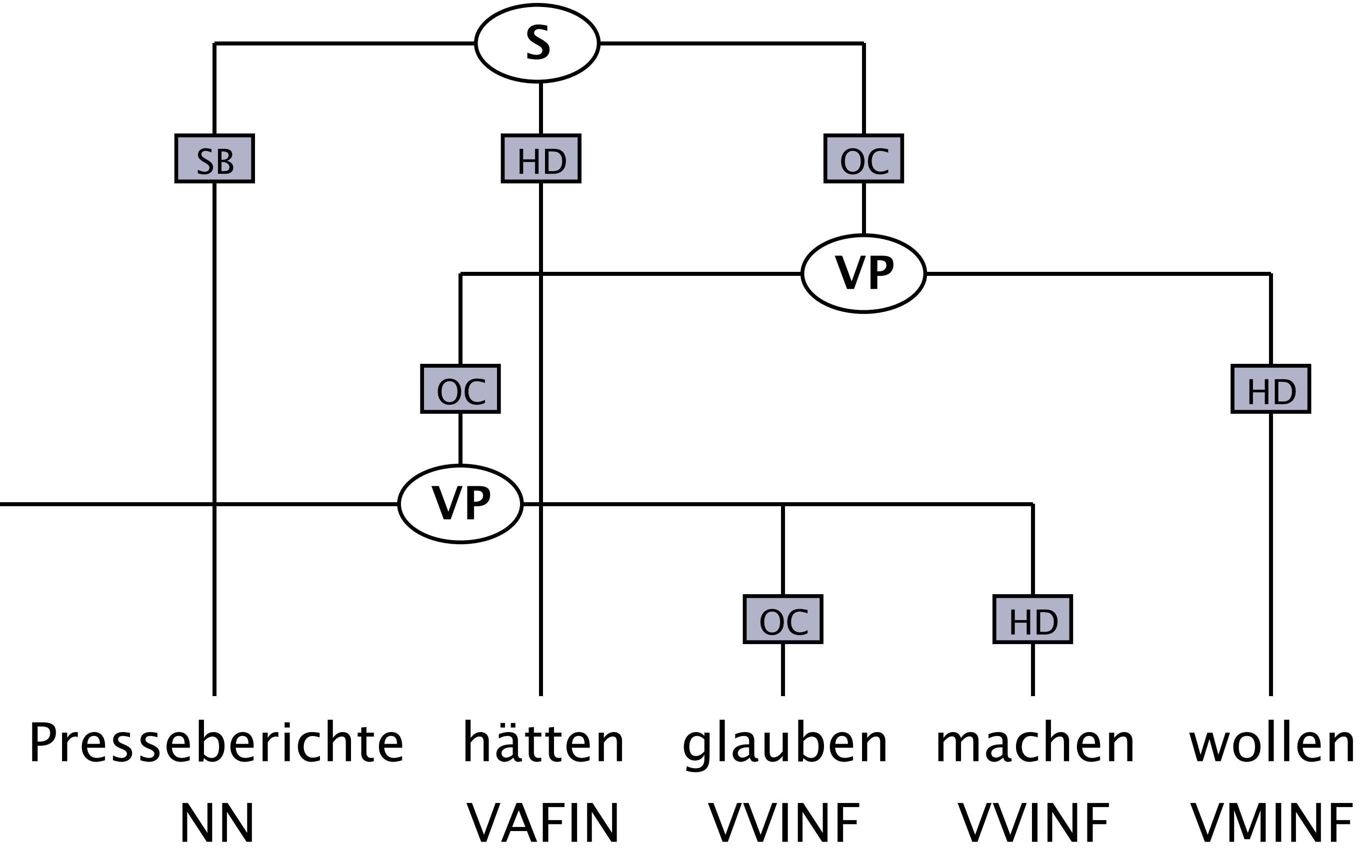
Example


PD
HD
RC
S
SB
OA
HD
anwesend
ist
,
der
sich
auskennt
.
ADJD
VAFIN
$,
PRELS
PRF
VVFIN
$.

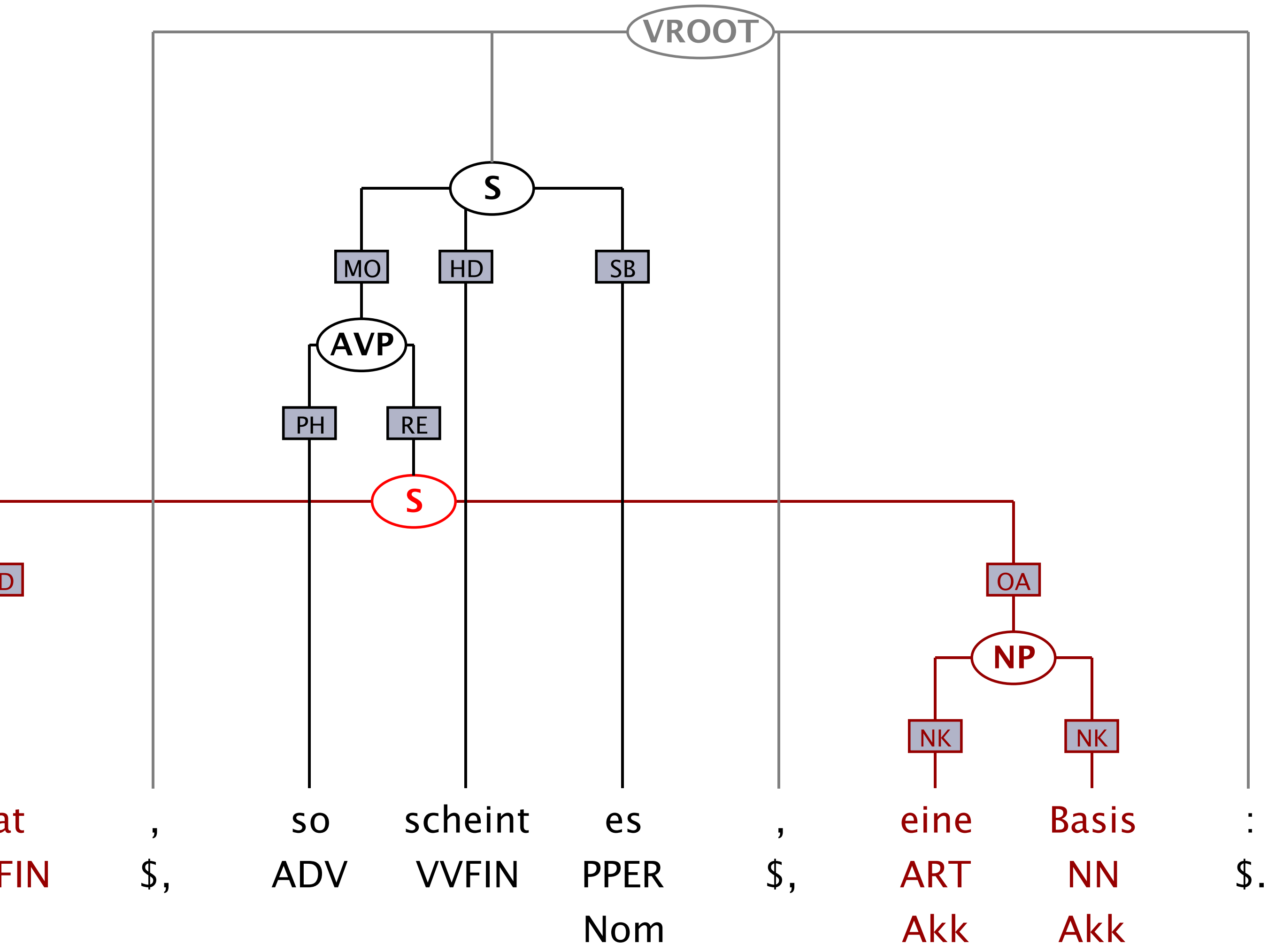
Verbal Complex (I)



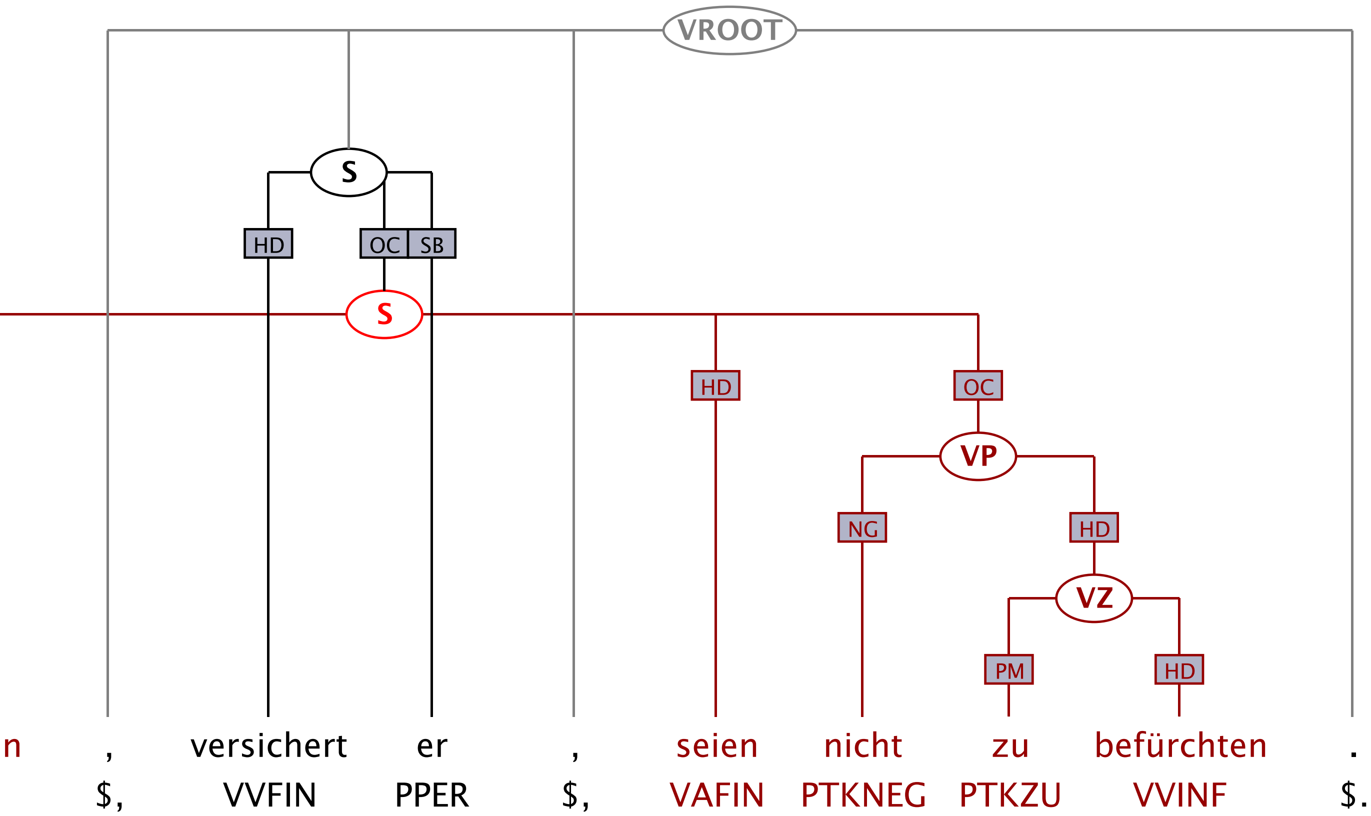
Verbal Complex (II)



Parentheticals (I)



Parentheticals (II)



Summary (I)

- Phrase Structure
- A Topological Model for German
- From Phrase Structure to GPSG and HPSG
  - Licensing Free Constituent Order with Phrase Structure
  - Immediate Dominance and Linear Precedence (GPSG)
  - HPSG: Valence, Obliqueness, and Constituent Order
- German Word Order Phenomena and their Analyses in Standard-HPSG/GPSG
  - Constituent Order in the Mittelfeld
  - The Mittelfeld and the Predicate Complex
  - The Position of the Finite Verb

Summary (II)

- Discontinuous Constituents
  - Head Projections as Linearization Domains
  - An Extension to Domain Union
- Parsing
  - With Continuous Constituents
  - With Discontinuous Constituents
- Syntactically Annotated Corpora and Discontinuity
  - The German Verb*mobil* Corpus
  - The Negra/Tiger Corpus

Good Bye

- We hope you enjoyed the course.

Good Bye

- We hope you enjoyed the course.
- We hope to have inspired you to work in the HPSG framework either theoretically or practically.

Good Bye

- We hope you enjoyed the course.
- We hope to have inspired you to work in the HPSG framework either theoretically or practically.
- If have questions do not hesitate to contact us.

Extraposition is a Nonlocal Dependency (I)

- extraposition is a nonlocal dependency, although this is often denied (for instance by Jacobson (1987), Grewendorf (1988), Haider (1996), and Rohrer (1996)):

(60) a. Karl hat mir [ein Bild [einer Frau $\_i$]] gegeben, [die schon lange tot ist]$_i$.
'Karl gave me a picture of a woman who has been dead for a long time.'

b. Karl hat mir [eine Fälschung [des Bildes [einer Frau $\_i$]]] gegeben, [die schon lange tot ist]$_i$.
'Karl gave me a forgery of the picture of a woman who has been dead for a long time.'

c. Karl hat mir [eine Kopie [einer Fälschung [des Bildes [einer Frau $\_i$]]]] gegeben, [die schon lange tot ist]$_i$.
'Karl gave me a copy of a forgery of the picture of a woman who has been dead for a long time.'

Extraposition is a Nonlocal Dependency (II)

Chomsky's claims in *Barriers* are wrong. He claims on the basis of (61) that extraposition may not cross two barriers.

(61) a. [$_{\text{NP}}$ Many books [$_{\text{PP}}$ with [stories t]] t'] were sold [that I wanted to read].

b. [$_{\text{NP}}$ Many proofs [$_{\text{PP}}$ of [the theorem t]] t'] appeared [that I wanted to think about].

According to him, the reading where the reltive clause corresponds to t is not availible.

Extraposition is a Nonlocal Dependency (II)

Chomsky's claims in *Barriers* are wrong. He claims on the basis of (61) that extraposition may not cross two barriers.

(61) a. [$_{\text{NP}}$ Many books [$_{\text{PP}}$ with [stories t]] t'] were sold [that I wanted to read].
b. [$_{\text{NP}}$ Many proofs [$_{\text{PP}}$ of [the theorem t]] t'] appeared [that I wanted to think about].

According to him, the reading where the reltive clause corresponds to t is not availible. This is just performance.

(62) a. weil viele Bücher mit Geschichten verkauft wurden, die ich noch lesen wollte.
b. weil viele Schallplatten mit Geschichten verkauft wurden, die ich noch lesen wollte.

(62a) has two readings, (62b) only one (if we assume that one does not read records).

Extraposition is a Nonlocal Dependency (III)

Some treat relative clause extraposition as adjunction + coreference, but complement extraposition is possible as well. One needs to obey subcategorizational requirements.

Examples from (Müller, 1999a, p. 206):

(63) a. Ich habe [von [der Vermutung $\_{i}$]] gehört,
[daß es Zahlen gibt, die die folgenden Bedingungen erfüllen]$_i$.
'I have heard of the assumption that there are numbers for which the following conditions hold.'

b. Ich habe [von [einem Beweis [der Vermutung $\_{i}$]]] gehört,
[daß es Zahlen gibt, die die folgenden Bedingungen erfüllen]$_i$.
'I have heard of a proof of the assumption that there are numbers for which the following conditions hold.'

c. Ich habe [von [dem Versuch [eines Beweises [der Vermutung $\_{i}$]]]] gehört,
[daß es Zahlen gibt, die die folgenden Bedingungen erfüllen]$_i$.
'I have heard of the attempt to prove the assumption that there are numbers for which the following conditions hold.'

Extraposition is a Nonlocal Dependency (III)

A corpus example:

(64) Für das Volk der Deutschen Demokratischen Republik ist dabei [die einmütige Bekräftigung [der Auffassung $\_{i}$]] wichtig, [daß es die Interessen des Friedens und der Sicherheit erfordern, daß alle Staaten gleichberechtigte Beziehungen auf völkerrechtlicher Grundlage zur Deutschen Demokratischen Republik aufnehmen und die bestehenden europäischen Staatsgrenzen einschließlich der Oder-Neiße-Grenze als endgültig und unantastbar anerkennen.]$_i$ (Neues Deutschland, 06.12.1969, p. 1)

'The unanimous confirmation of the opinion that the interests of peace and security require that all countries establish relationships to the German Democratic Republic on the basis of international law and that all countries accept the existing state borders including the Oder Neiße border as final and inviolable is important for the people of the German Democratic Republic.'

Extraposition is Bounded

- Although extraposition is clearly a nonlocal dependency it is (clause) bounded. In German finite clauses and VPs act as bounderies.

Kathol and Pollard 1995

Extraposition is Bounded

- Although extraposition is clearly a nonlocal dependency it is (clause) bounded. In German finite clauses and VPs act as bounderies.

- In this respect extraposition differs from fronting.

  Kathol and Pollard 1995

Extraposition is Bounded

- Although extraposition is clearly a nonlocal dependency it is (clause) bounded. In German finite clauses and VPs act as bounderies.
- In this respect extraposition differs from fronting.
- Kathol and Pollard (1995) suggested a treatment of extraposition via constituent order domains.
  The domain formation relations are more complicated. For instance they split an NP consisting of a Det, N, and Reletive Clause into an NP and a Relative Clause.

Particle Verbs in German Dialects (and Dutch)

Werner (1994, p. 356): examples spoken in the northwest of Sonneberg/Thuringia:

(65) a.

a	. . .	hot	aa	ze	schimpfm	gfanga
he		has	PART	to	get.angry	caught

‘He started to get angry.’

b.

die	ham	. . .	auf	zu	arwettn	ghört
they	have		PART	to	work	heard

‘They stopped working.’

c.

ham	sa	groud	aa	mit	assn	gfanga
have	they	just	PART	with	eat	caught

‘Did they just start to eat?’

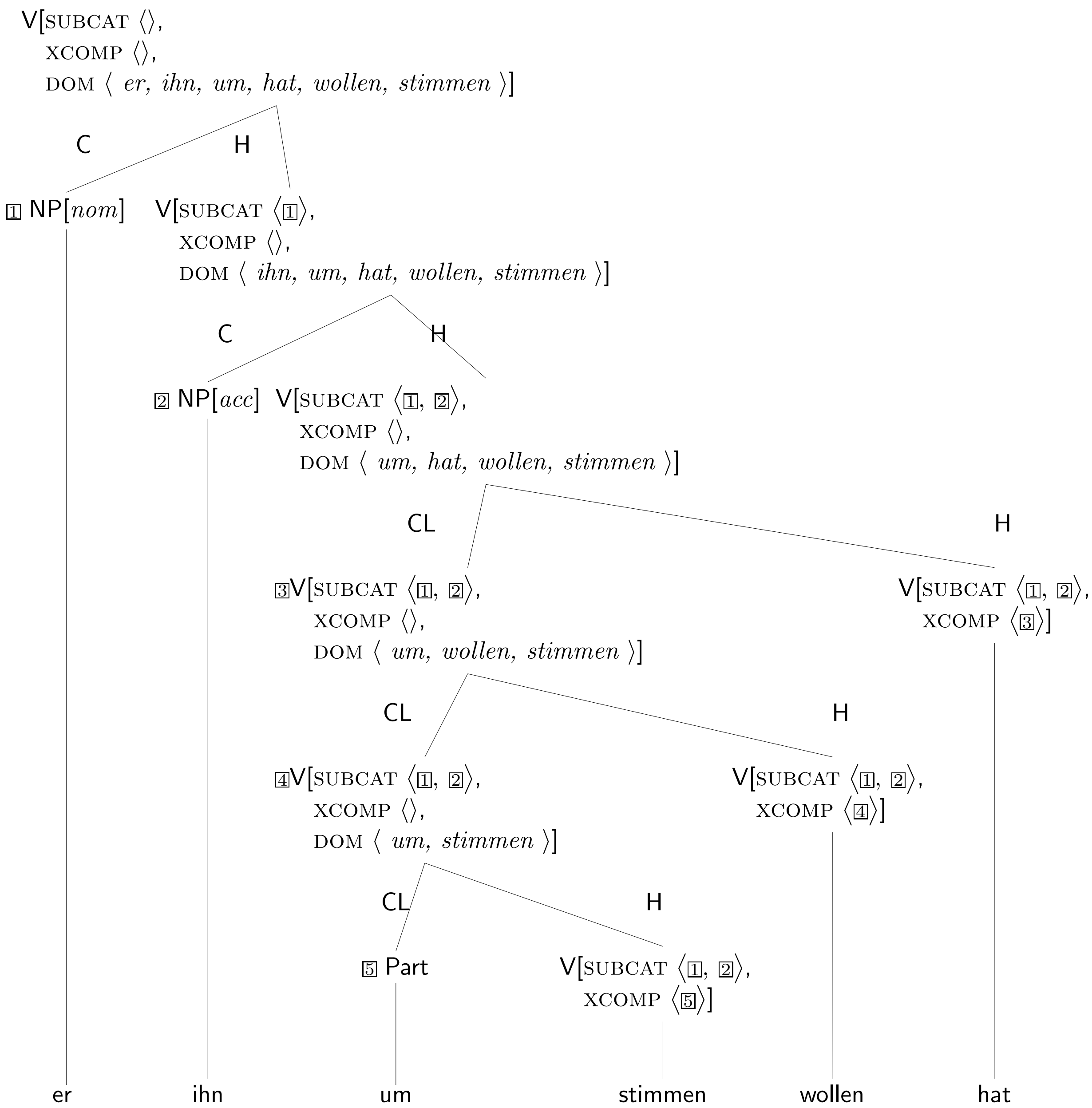
(66)

weil	er	ihn	um	hat	wollen	stimmen.
because	he	him	PART	has	want.to	tune

‘because he wanted to change his mind.’

With order domains we can have a structured verbal complex but a discontinuous serialization of the particle.

Particle Verbs: An Example Analysis



References

Andrews, Avery. 1982. The Representation of Case in Modern Icelandic. In Joan Bresnan (ed.), The Mental Representation of Grammatical Relations, MIT Press Series on Cognitive Theory and Mental Representation, pages 427–503, Cambridge: Massachusetts, London: England: The MIT Press.

Baker, Kathryn L. 1999. "Modal Flip" and Partial Verb Phrase Fronting in German. In Robert D. Levine and Georgia M. Green (eds.), Studies in Contemporary Phrase Structure Grammar, pages 161–198, Cambridge: Cambridge University Press.

Batliner, A., Feldhaus, A., Geissler, S., Kiss, T., Kompe, R. and Noeth, E. 1996. Prosody, Empty Categories and Parsing – A Succes Story. Technical Report, Erlangen.

Bausewein, Karin. 1990. Haben kopflose Relativsätze tatsächlich keine Köpfe? In Fanselow and Felix (1990), pages 144–158.

Behaghel, Otto. 1930. Von deutscher Wortstellung. Zeitschrift für Deutschkunde pages 81–89.

Bierwisch, Manfred. 1963. Grammatik des deutschen Verbs. studia grammatica II, Berlin: Akademie Verlag.

Blevins, James. 1990. Syntactic Complexity: Evidence for Discontinuity and Multidomination. Ph. D.thesis, University of Massachusetts, Amherst, MA.

Bonami, Olivier and Godard, Danièle. To Appear. Incidential Adjuncts: An Overlooked Type of Adjunction. In Stefan Müller (ed.), Proceedings of the HPSG-2003 Conference, Michigan State University, East Lansing, CSLI Publications, http://cslipublications.stanford.edu/HPSG/HPSG03/.

Brants, Sabine, Dipper, Stefanie, Hansen, Silvia, Lezius, Wolfgang and Smith, George. 2002. The TIGER Treebank. In Proceedings of the Workshop on Treebanks and Linguistic Theories, Sozopol, Bulgaria.

Bröker, Norbert. 1998. Separating Surface Order and Syntactic Relations in a Dependency Grammar. In Proceedings of the 17th International Conference on Computational Linguistics (COLING) and the 36th Annual meeting of the ACL (ACL), Montreal.

Bunt, Harry and van Horck, Arthur (eds.). 1996. Discontinuous Constituency. Natural Language Processing, No. 6, Berlin, New York: Mouton de Gruyter.

Callmeier, Ulrich. 2000. PET. A platform for experimentation with efficient HPSG processing techniques. Natural Language Engineering 6(1), 99–108, special Issue on Efficient processing with HPSG: Methods, systems, evaluation.

Campbell-Kibler, Kathryn. 2001. Bech's Problem, Again: The Dutch Word er. In HPSG-2001, Abstracts, Trondheim: NTNU, Trondheim, Department of Linguistics.

Chomsky, Noam. 1986. Barriers, volume 13 of Linguistic Inquiry Monographs. Cambridge, Massachusetts - London, England: MIT - Press.

Copestake, Ann. 1999. The (New) LKB System, ftp://www-csli.stanford.edu/˜aac/newdoc.pdf. 06.24.99.

Crysmann, Berthold. 2003. On the Treatment of Word Order in a Computational Grammar of German. Talk presented at the HPSG-Workshop on Germanic at the University of Bremen 07.08.–08.08.2003.

Daniels, Mike and Meurers, W. Detmar. 2002. Improving the Efficiency of Parsing with Discontinuous Constituents. In Shuly Wintner (ed.), Proceedings of NLULP'02: The 7th International Workshop on Natural Language Understanding and Logic Programming, Datalogiske Skrifter, No. 92, pages 49–68, Copenhagen: Roskilde Universitetscenter.

Donohue, Cathryn and Sag, Ivan A. 1999. Domains in Warlpiri. In Sixth International Conference on HPSG–Abstracts. 04–06 August 1999, pages 101–106, Edinburgh, http://www-csli.stanford.edu/~sag/papers/warlpiri.ps. 08.18.2002.

Dowty, David R. 1996. Towards a Minimalist Theory of Syntactic Structure. In Harry Bunt and Arthur van Horck (eds.), Discontinuous Constituency, volume 6 of Natural Language Processing, Berlin and New York, NY: Mouton de Gruyter.

Drach, Erich. 1937. Grundgedanken der deutschen Satzlehre. Darmstadt: Wissenschaftliche Buchgesellschaft, 4., unveränderte Auflage 1963.

Earley, Jay. 1970. An Efficient Context-Free Parsing Algorithm. Communications of the ACM 13(2), 94–102, also in Grosz et. al. (1986).

Engelkamp, Judith, Erbach, Gregor and Uszkoreit, Hans. 1992. Handling Linear Precedence Constraints by Unification. In Henry S. Thomson (ed.), 30th Annual Meeting of the Association for Computational Linguistics. Proceedings of the Conference, pages 201–208, Newark, Delaware: Association for Computational Linguistics, also appeared as CLAUS-Report, No. 19, University of the Saarland.

Fanselow, Gisbert and Felix, Sascha W. (eds.). 1990. Strukturen und Merkmale syntaktischer Kategorien. Studien zur deutschen Grammatik, No. 39, Tübingen: Gunter Narr Verlag.

Fourquet, Jean. 1957. Review of: Heinz Anstock: Deutsche Syntax – Lehr und Übungsbuch. Wirkendes Wort 8, 120–122.

Fouvry, Frederik and Meurers, Detmar. 2000. Towards a platform for linearization grammars. In Erhard W. Hinrichs, Detmar Meurers and Shuly Wintner (eds.), Proceedings of the Workshop on Linguistic Theory and Grammar Implementation, pages 153–168, Birmingham, UK: ESSLLI 2000.

Frank, Anette. 1994. Verb Second by Lexical Rule or by Underspecification. Arbeitspapiere des SFB 340 No. 43, IBM Deutschland GmbH, Heidelberg, ftp://ftp.ims.uni-stuttgart.de/pub/papers/anette/v2-usp.ps.gz. 08.20.2002.

Fries, Norbert. 1988. Über das Null-Topik im Deutschen. Forschungsprogramm Sprache und Pragmatik 3, Germanistisches Institut der Universität Lund, Lund.

Gazdar, Gerald, Klein, Evan, Pullum, Geoffrey K. and Sag, Ivan. 1985. Generalized Phrase Structure Grammar. Cambridge, Massachusetts: Harvard University Press.

Geißler, Stefan. 1994. Lexikalische Regeln in der IBM-Basisgrammatik. Verbmobil Report 20, IBM Informationssysteme GmbH, Institut für Logik und Linguistik, Heidelberg.

Görz, Günther (ed.). 1992. Konvens 92. 1. Konferenz „Verarbeitung natürlicher Sprache“. Nürnberg 7.–9. Oktober 1992, Informatik aktuell, Berlin, Heidelberg, New York, Springer-Verlag.

Götz, Thilo and Penn, Gerald. 1997. A Proposed Linear Specification Language. Arbeitspapiere des Sonderforschungsbereichs 340 134, SFB 340, Universität Tübingen, Tübingen, http://www.sfs.uni-tuebingen.de/sfb/reports/berichte/134/134abs.html.

Grewendorf, Günther. 1985. Anaphren bei Objekt-Koreferenz im Deutschen. Ein Problem für die Rektions-Bindungs-Theorie. In Werner Abraham (ed.), Erklärende Syntax des Deutschen, Studien zur deutschen Grammatik, No. 25, pages 137–171, Tübingen: Gunter Narr Verlag.

Grewendorf, Günther. 1988. Aspekte der deutschen Syntax. Eine Rektions-Bindungs-Analyse. Studien zur deutschen Grammatik, No. 33, Tübingen:

Gunter Narr Verlag.

Grosz, Barbara, Jones, Karen Sparck and Webber, Bonnie Lynn (eds.). 1986. Readings in Natural Language Processing. Los Altos, California: Morgan Kaufmann.

Gunji, Takao. 1986. Subcategorization and Word Order. In William J. Poser (ed.), Papers from the Second International Workshop on Japanese Syntax, pages 1–21, Stanford: CSLI Publications.

Haider, Hubert. 1990. Pro-bleme? In Fanselow and Felix (1990), pages 121–143.

Haider, Hubert. 1996. Downright Down to the Right. In On Extraction and Extraposition in German, Linguistik Aktuell / Linguistics Today, No. 11, pages 245–271, Amsterdam: Benjamins.

Heck, Fabian. 2000. Tiefenoptimierung: Deutsche Wortstellung als wettbewerbsgesteuerte Basisgenerierung. Linguistische Berichte 184, 441–468.

Hepple, Mark. 1994. Discontinuity and the Lambek Calculus. In Proceedings of the 15th Conference on Computational Linguistics (COLING-94), Kyoto, ftp://ftp.dcs.shef.ac.uk/home/hepple/papers/coling94.ps.

Hinrichs, Erhard, Bartels, Julia, Kawata, Yasuhiro, Kordoni, Valia and Telljohann, Heike. 2000. The Tübingen Treebanks for Spoken German, English, and Japanese. In Wolfgang Wahlster (ed.), Verbmobil: Foundations of Speech-to-Speech Translation, Artificial Intelligence, pages 552–576, Berlin: Springer.

Hinrichs, Erhard W. and Nakazawa, Tsuneko. 1989. Subcategorization and VP Structure in German. In Aspects of German VP Structure, SfS-Report-01-93, Eberhard-Karls-Universität Tübingen.

Hoberg, Ursula. 1981. Die Wortstellung in der geschriebenen deutschen Gegenwartssprache. Heutiges Deutsch. Linguistische Grundlagen. Forschungen des Instituts für deutsche Sprache, No. 10, München: Max Huber Verlag.

Höhle, Tilman N. 1982. Explikation für „normale Betonung“ und „normale Wortstellung“. In Werner Abraham (ed.), Satzglieder im Deutschen – Vorschläge zur syntaktischen, semantischen und pragmatischen Fundierung, Studien zur deutschen Grammatik, No. 15, pages 75–153, Tübingen: Gunter Narr Verlag.

Höhle, Tilman N. 1986. Der Begriff „Mittelfeld“, Anmerkungen über die Theorie der topologischen Felder. In Walter Weiss, Herbert Ernst Wiegand and Marga Reis (eds.), Akten des VII. Kongresses der Internationalen Vereinigung für germanische Sprach- und Literaturwissenschaft. Göttingen 1985. Band 3. Textlinguistik contra Stilistik? – Wortschatz und Wörterbuch – Grammatische oder pragmatische Organisation von Rede? volume 4 of Kontroversen, alte und neue, pages 329–340, Tübingen: Max Niemeyer Verlag.

Huck, Geoffrey. 1985. Exclusivity and discontinuity in phrase structure grammar. In West Coast Conference on Formal Linguistics (WCCFL), volume 4, pages 92–98, Stanford University, CSLI Publications.

Huck, Geoffrey and Ojeda, Almerindo (eds.). 1987. Discontinuous Constituency. Syntax and Semantics, No. 20, New York, et al.: Academic Press.

Huynh, Dung T. 1983. Commutative Grammars: The Complexity of Uniform Word Problems. Information and Control 57(1), 21–39.

Jackendoff, Ray S. 1977. $\overline{\mathrm{X}}$ Syntax: A Study of Phrase Structure. Cambridge: Massachusetts, London: England: The MIT Press.

Jacobs, Joachim. 1986. The Syntax of Focus and Adverbials in German. In Werner Abraham and S. de Meij (eds.), Topic, Focus, and Configurationality.

Papers from the 6th Groningen Grammar Talks, Groningen, 1984, Linguistik Aktuell / Linguistics Today, No. 4, pages 103–127, Amsterdam, Philadelphia: John Benjamins Publishing Company.

Jacobson, Pauline. 1987. Phrase Structure, Grammatical Relations, and Discontinuous Constituents. In Geoffrey J. Huck and Almerindo E. Ojeda (eds.), Discontinuous Constituency, volume 20 of Syntax and Semantics, pages 27–69, New York: Academic Press.

Johnson, Mark. 1985. Parsing with Discontinuous Constituents. In William C. Mann (ed.), Proceedings of the Twenty-Third Annual Meeting of the Association for Computational Linguistics, pages 127–132, Association for Computational Linguistics, Chicago, IL.

Johnson, Mark and Kay, Martin. 1992. Parsing and Empty Nodes. Arbeitspapiere des SFB 340 No. 25, IBM Deutschland GmbH, Heidelberg.

Kasper, Robert T. 1994. Adjuncts in the Mittelfeld. In Nerbonne et. al. (1994), pages 39–70.

Kathol, Andreas. 1995. Linearization-Based German Syntax. Ph. D.thesis, Ohio State University.

Kathol, Andreas. 1998. Constituency and Linearization of Verbal Complexes. In Erhard W. Hinrichs, Andreas Kathol and Tsuneko Nakazawa (eds.), Complex Predicates in Nonderivational Syntax, volume 30 of Syntax and Semantics, pages 221–270, San Diego: Academic Press.

Kathol, Andreas. 2000. Linear Syntax. New York, Oxford: Oxford University Press.

Kathol, Andreas and Pollard, Carl J. 1995. Extraposition via Complex Domain Formation. In Proceedings of the Thrirty-Third Annual Meeting of the ACL, Association for Computational Linguistics, Boston, http://linguistics.berkeley.edu/~kathol/Papers/ACL95.ps.gz. 06.29.99.

Kay, Martin. 1990. Head-Driven Parsing. In Masaru Tomita (ed.), Current Issues in Parsing Technology, Dordrecht: Kluwer Academic Publishers, previously published in the proceedings of the International Workshop on Parsing Technologies, 1989.

Keenan, Edward L. and Comrie, B. 1977. Noun Phrase Accessibility and Universal Grammar. Linguistic Inquiry 8, 63–99.

Kiss, Tibor. 1993. Infinite Komplementation – Neue Studien zum deutschen Verbum infinitum. Arbeiten des SFB 282 No. 42, Bergische Universität Gesamthochschule Wuppertal.

Kiss, Tibor. 1995. Infinite Komplementation. Neue Studien zum deutschen Verbum infinitum. Linguistische Arbeiten, No. 333, Tübingen: Max Niemeyer Verlag.

Kiss, Tibor. 2001. Configurational and Relational Scope Determination in German. In Meurers and Kiss (2001), pages 131–166.

Kiss, Tibor and Wesche, Birgit. 1991. Verb Order and Head Movement. In Otthein Herzog and Claus-Rainer Rollinger (eds.), Text Understanding in LILOG, Lecture Notes in Artificial Intelligence, No. 546, pages 216–242, Berlin Heidelberg New York: Springer-Verlag.

Klein, Wolfgang. 1985. Ellipse, Fokusgliederung und thematischer Stand. In Reinhard Meyer-Hermann and Hannes Rieser (eds.), Ellipsen und fragmentarische Ausdrücke, pages 1–24, Tübingen: Max Niemeyer Verlag.

Kolb, Hans-Peter and Thiersch, Craig L. 1991. Levels and Empty Categories in a Principles and Parameters Based Approach to Parsing. In Hubert Haider and Klaus Netter (eds.), Representation and Derivation in the Theory of Grammar, Studies in Natural Language and Linguistic Theory, No. 22, Dordrecht/Boston/London: Kluwer Academic Publishers.

Koronai, András and Pullum, Geoffrey K. 1990. The X-bar Theory of Phrase Structure. Language 66(1), 24–50.

Kroch, Anthony S. and Joshi, Aravind K. 1987. Analyzing Extraposition in a Tree Adjoining Grammar. In Huck and Ojeda (1987).

McCawley, James D. 1982. Parentheticals and discontinuous constituent structure. Linguistic Inquiry 13(1), 91–106.

Meurers, W. Detmar, Penn, Gerald and Richter, Frank. 2002. A Web-based Instructional Platform for Constraint-Based Grammar Formalisms and Parsing. In Dragomir Radev and Chris Brew (eds.), Effective Tools and Methodologies for Teaching NLP and CL, proceedings of the Workshop held at 40th Annual Meeting of the Association for Computational Linguistics. Philadelphia, PA.

Meurers, Walt Detmar. 1994. On Implementing an HPSG theory. In Erhard W. Hinrichs, Walt Detmar Meurers and Tsuneko Nakazawa (eds.), Partial-VP and Split-NP Topicalization in German – An HPSG Analysis and its Implementation, Arbeitspapiere des SFB 340, No. 58, Eberhard-Karls-Universität Tübingen, http://ling.osu.edu/˜dm/on-implementing.html. 06.12.96.

Meurers, Walt Detmar. 1999. Raising Spirits (and Assigning Them Case). Groninger Arbeiten zur Germanistischen Linguistik (GAGL) 43, 173–226, http://ling.osu.edu/˜dm/papers/gagl99.html. 04.18.2000.

Meurers, Walt Detmar. 2000. Lexical Generalizations in the Syntax of German Non-Finite Constructions. Arbeitspapiere des SFB 340 No. 145, Eberhard-Karls-Universität, Tübingen, http://www.ling.ohio-state.edu/˜dm/papers/diss.html. 08.19.2002.

Meurers, Walt Detmar and Kiss, Tibor (eds.). 2001. Constraint-Based Approaches to Germanic Syntax. Studies in Constraint-Based Lexicalism, No. 7, Stanford: CSLI Publications, http://cslipublications.stanford.edu/site/1575863049.html. 08.18.2002.

Morrill, Glynn V. 1995. Discontinuity in categorial grammar. Linguistics and Philosophy 18, 175–219.

Müller, Stefan. 1995. Scrambling in German – Extraction into the *Mittelfeld*. In Benjamin K. T'sou and Tom Bong Yeung Lai (eds.), Proceedings of the tenth Pacific Asia Conference on Language, Information and Computation, pages 79–83, City University of Hong Kong, http://www.dfki.de/˜stefan/Pub/scrambling.html. 28.03.2004.

Müller, Stefan. 1996. The Babel-System—An HPSG Prolog Implementation. In Proceedings of the Fourth International Conference on the Practical Application of Prolog, pages 263–277, London, http://www.dfki.de/˜stefan/Pub/babel.html. 28.03.2004.

Müller, Stefan. 1997. Yet another Paper about Partial Verb Phrase Fronting in German. Research Report RR-97-07, Deutsches Forschungszentrum für Künstliche Intelligenz, Saarbrücken, a shorter version appeared in *Proceedings of COLING 96*, pages 800–805. http://www.dfki.de/˜stefan/Pub/pvp.html. 28.03.2004.

Müller, Stefan. 1999a. Deutsche Syntax deklarativ. Head-Driven Phrase Structure Grammar für das Deutsche. Linguistische Arbeiten, No. 394, Tübingen: Max Niemeyer Verlag, http://www.dfki.de/˜stefan/Pub/hpsg.html. 28.03.2004.

Müller, Stefan. 1999b. An HPSG-Analysis for Free Relative Clauses in German. Grammars 2(1), 53–105, http://www.dfki.de/˜stefan/Pub/freeRel.html. 28.03.2004.

Müller, Stefan. 1999. Restricting Discontinuity. Verbmobil Report 237, DFKI, Saarbrücken, also published in the Proceedings of GLDV 99 (Frankfurt/Main). Available from http://www.dfki.de/˜stefan/Pub/e_restricting.html.

Müller, Stefan. 2000a. Complex Predicates: Verbal Complexes, Resultative Constructions, and Particle Verbs in German. Habilitationsschrift, Universität des Saarlandes, Saarbrücken, http://www.dfki.de/˜stefan/Pub/complex.html. 28.03.2004.

Müller, Stefan. 2000b. Continuous or Discontinuous Constituents? In Erhard W. Hinrichs, Walt Detmar Meurers and Shuly Wintner (eds.), Proceedings of the ESSLLI-2000 Workshop on Linguistic Theory and Grammar Implementation, pages 133–152, Birmingham, UK, August 14–18, http://www.dfki.de/~stefan/Pub/discont.html. 28.03.2004.

Müller, Stefan. 2001. An HPSG Analysis of German Depictive Secondary Predicates. In Proceedings of Formal Grammar 2001 / MOL 7, Electronic Notes in Theoretical Computer Science, No. 53, Helsinki, http://www.dfki.de/~stefan/Pub/depiktive.html. 28.03.2004.

Müller, Stefan. 2002. Complex Predicates: Verbal Complexes, Resultative Constructions, and Particle Verbs in German. Studies in Constraint-Based Lexicalism, No. 13, Stanford: CSLI Publications, http://www.dfki.de/~stefan/Pub/complex.html. 28.03.2004.

Müller, Stefan. To Appear a. Continuous or Discontinuous Constituents? A Comparison between Syntactic Analyses for Constituent Order and Their Processing Systems. Research on Language and Computation, Special Issue on Linguistic Theory and Grammar Implementation 1(4), http://www.dfki.de/~stefan/Pub/discont.html. 28.03.2004.

Müller, Stefan. To Appear b. Zur Analyse der scheinbar mehrfachen Vorfeldbesetzung. Linguistische Berichte http://www.dfki.de/~stefan/Pub/mehr-vf-lb.html. 28.03.2004.

Müller, Stefan and Kasper, Walter. 2000. HPSG Analysis of German. In Wahlster (2000), pages 238–253.

Nerbonne, John. 1994. Partial Verb Phrases and Spurious Ambiguities. In Nerbonne et. al. (1994), pages 109–150.

Nerbonne, John, Netter, Klaus and Pollard, Carl J. (eds.). 1994. German in Head-Driven Phrase Structure Grammar. CSLI Lecture Notes, No. 46, Stanford: CSLI Publications.

Netter, Klaus. 1992. On Non-Head Non-Movement. An HPSG Treatment of Finite Verb Position in German. In Görz (1992), pages 218–227.

Ojeda, Almerindo. 1987. Discontinuity, multidominances and unbounded dependency in Generalized Phrase Structure Grammar. In Huck and Ojeda (1987).

Oliva, Karel. 1992. The Proper Treatment of Word Order in HPSG. In ICCL (ed.), 14th International Conference on Computational Linguistics (COLING '92), August 23–28, pages 184–190, Nantes, France: ACL – Association for Computational Linguistics.

Paritong, Maike. 1992. Constituent Coordination in HPSG. In Görz (1992), pages 228–237.

Penn, Gerald. 1999. Linearization and WH-Extraction: Evidence from Serbo-Croatian. In Robert D. Borsley and Adam Przepiórkowski (eds.), Slavic in Head-Driven Phrase Structure Grammar, Studies in Constraint-Based Lexicalism, No. 2, pages 149–182, Stanford: CSLI Publications.

Pittner, Karin. 1995. Regeln für die Bildung von freien Relativsäzen. Eine Antwort an Oddleif Leirbukt. Deutsch als Fremdsprache 32(4), 195–200.

Plátek, Martin, Holan, Tomáš, Kuboň, Vladimir and Oliva, Karel. 2001. Word-Order Relaxations and Restrictions within a Dependency Grammar. In G. Satta (ed.), Proceedings of the Seventh International Workshop on Parsing Technologies (IWPT), pages 237–240, Beijing: Tsinghua University Press.

Pollard, Carl J. 1996. On Head Non-Movement. In Bunt and van Horck (1996), pages 279–305, veröffentlichte Version eines Ms. von 1990.

Pollard, Carl J., Kasper, Robert T. and Levine, Robert D. 1992. Linearization Grammar. Research Proposal to the National Science Foundation.

Pollard, Carl J., Kasper, Robert T. and Levine, Robert D. 1994. Studies in Constituent Ordering: Toward a Theory of Linearization in Head-Driven Phrase

Structure Grammar, research Proposal to the National Science Foundation. ftp://csli-ftp.stanford.edu/linguistics/sag/linearization-prop.ps.gz. 08.20.2002.

Pollard, Carl J. and Sag, Ivan A. 1987. Information-Based Syntax and Semantics. CSLI Lecture Notes, No. 13, Stanford: CSLI Publications.

Pollard, Carl J. and Sag, Ivan A. 1994. Head-Driven Phrase Structure Grammar. Studies in Contemporary Linguistics, Chicago, London: University of Chicago Press.

Pullum, Geoffrey K. 1977. Word Order Universals and Grammatical Relations. In Peter Cole and Jerrold M. Sadock (eds.), Grammatical Relations, volume 8 of Syntax and Semantics, pages 249–277, New York, San Francisco, London: Academic Press.

Rambow, Owen and Joshi, Aravind. 1994. A Formal Look at Dependency Grammars and Phrase-Structure Grammars, with Special Consideration of Word-Order Phenomena. In L. Wanner (ed.), Current Issues in Meaning-Text-Theory, London: Pinter, http://arxiv.org/abs/cmp-lg/9410007.

Ramsay, Allan M. 1999a. Direct Parsing with Discontinuous Phrases. Natural Language Engineering 5(3), 271–300.

Ramsay, Allan M. 1999b. Parsing with discontinuous phrases. Natural Language Engineering 5(3), 271–300.

Reape, Mike. 1991. Word Order Variation in Germanic and Parsing. DYANA Report Deliverable R1.1.C, University of Edinburgh.

Reape, Mike. 1992. A Formal Theory of Word Order: A Case Study in West Germanic. Ph. D.thesis, University of Edinburgh.

Reape, Mike. 1994. Domain Union and Word Order Variation in German. In Nerbonne et. al. (1994), pages 151–198.

Reape, Mike. 1996. Getting Things in Order. In Bunt and van Horck (1996), pages 209–253, published version of a Ms. dated January 1990.

Reis, Marga. 1974. Syntaktische Haupsatzprivilegien und das Problem der deutschen Wortstellung. Zeitschrift für germanistische Linguistik 2(3), 299–327.

Reis, Marga. 1980. On justifying Topological Frames: 'Positional Field' and the Order of Nonverbal Constituents in German. DRLAV: Revue de Linguistique 22/23, 59–85.

Richter, Frank and Sailer, Manfred. 2001. On the Left Periphery of German. In Meurers and Kiss (2001).

Rohrer, Christian. 1996. Fakultativ kohärente Infinitkonstruktionen im Deutschen und deren Behandlung in der Lexikalisch Funktionalen Grammatik. In Gisela Harras and Manfred Bierwisch (eds.), Wenn die Semantik arbeitet. Klaus Baumgärtner zum 65. Geburtstag, pages 89–108, Tübingen: Max Niemeyer Verlag.

Schiller, Anne, Teufel, Simone and Thielen, Christine. 1995. Guidlines für das Taggen deutscher Textcorpora mit STTS. Technical Report, IMS-CL, Univ. Stuttgart and SfS, Univ. Tübingen, http://www.cogsci.ed.ac.uk/˜simone/stts_guide.ps.gz.

Skut, Wojciech, Brants, Thorsten, Krenn, Brigitte and Uszkoreit, Hans. 1998. A Linguistically Interpreted Corpus of German Newspaper Text. In Proceedings of the ESSLLI Workshop on Recent Advances in Corpus Annotation, Saarbrücken, Germany, http://www.coli.uni-sb.de/˜thorsten/publications/Skut-ea-ESSLLI-Corpus98.ps.gz.

Skut, Wojciech, Krenn, Brigitte, Brants, Thorsten and Uszkoreit, Hans. 1997. An Annotation Scheme for Free Word Order Languages. In Proceedings of the 5th Conference on Applied Natural Language Processing (ANLP), Washington, D.C., http://www.coli.uni-sb.de/˜thorsten/publications/Skut-ea-ANLP97.ps.gz.

Stegmann, Rosmary, Telljohann, Heike and Hinrichs, Erhard W. 2000. Stylebook for the German Treebank in VERBMOBIL. Verbmobil-Report 239, Universität Tübingen, Tübingen, Germany, http://verbmobil.dfki.de/cgi-bin/verbmobil/htbin/decode.cgi/share/VM-depot/FTP-SERVER/vm-reports/report-239-00.ps.

Suhre, Oliver. 1999. Computational Aspects of a Grammar Formalism for Languages with Freer Word Order. Diplomarbeit, Department of Computer Science, University of Tübingen, published 2000 as Volume 154 in Arbeitspapiere des SFB 340, http://www.sfs.uni-tuebingen.de/sfb/reports/berichte/154/154abs.html.

Thielen, Christine and Schiller, Anne. 1996. Ein kleines und erweitertes Tagset fürs Deutsche. In Helmut Feldweg and Erhard W. Hinrichs (eds.), Lexikon und Text: wiederverwendbare Methoden und Ressourcen zur linguistischen Erschließung des Deutschen, volume 73 of Lexicographica: Series maior, pages 215–226, Tübingen: Max Niemeyer Verlag.

Thiersch, Craig L. 1978. Topics in German Syntax. Dissertation, M.I.T.

Uszkoreit, Hans. 1986. Linear Precedence in Discontinuous Constituents: Complex Fronting in German. Report No. CSLI-86-47, Center for the Study of Language and Information, Stanford.

Uszkoreit, Hans. 1987. Word Order and Constituent Structure in German. CSLI Lecture Notes, No. 8, Stanford: CSLI Publications.

Uszkoreit, Hans, Backofen, Rolf, Busemann, Stephan, Diagne, Abdel Kader, Hinkelman, Elizabeth A., Kasper, Walter, Kiefer, Bernd, Krieger, Hans-Ulrich, Netter, Klaus, Neumann, Günter, Oepen, Stephan and Spackman, Stephen P. 1994. DISCO—An HPSG-based NLP System and its Application for Appointment Scheduling. In COLING Staff (ed.), Proceedings of COLING 94, pages 436–440, Kyoto, Japan: ACL – Association for Computational Linguistics.

van Noord, Gertjan. 1997. An Efficient Implementation of the Head-Corner Parser. Computational Linguistics 23(3), 425–456.

van Noord, Gertjan, Bouma, Gosse, Koeling, Rob and Nederhof, Mark-Jan. 1999. Robust Grammatical Analysis for Spoken Dialogue Systems. Natural Language Engineering 1(5), 45–93, e-Print-Archive: http://xxx.lanl.gov/abs/cmp-lg/9906026. 07.02.99.

Wahlster, Wolfgang (ed.). 2000. Verbmobil: Foundations of Speech-to-Speech Translation. Artificial Intelligence, Berlin Heidelberg New York: Springer-Verlag.

Werner, Otmar. 1994. Was da sich ölles aahotmüßhör! ‚Was der sich alles hat anhören müssen!' Auxiliar-Inkorporation im Ostfränkisch-Thüringischen. In Heinrich Löffler, Karlheinz Jakob and Bernhard Kelle (eds.), Texttyp, Sprechergruppe, Kommunikationsbereich. Studien zur deutschen Sprache in Geschichte und Gegenwart. Festschrift für Hugo Steger zum 65. Geburtstag, pages 343–361, Berlin, New York: Walter de Gruyter.

Yatabe, Shûichi. 2001. The syntax and semantics of left-node raising in Japanese. In Dan Flickinger and Andreas Kathol (eds.), Proceedings of the HPSG-2000 Conference, University of California, Berkeley, pages 325–344, CSLI Publications, http://cslipublications.stanford.edu/HPSG/HPSG00/. 08.20.2002.