Computerlinguistische Modellierung der deutschen Satzstruktur

Arkadiusz Jasiński

20. September 2006

formale Sprachbeschreibung – Muss das sein?

computerlinguistische Motivation:

formal definierter Objektbereich / anwendungsorientierte CL

sprachwissenschaftliche Motivation:

präzise Modellierung der sprachlichen Kompetenz

"I have always understood a generative grammar to be nothing more than an **explicit** *grammar."* (Chomsky 1995:162)

formale Sprachen – ein Demonstrationsbeispiel

Annahme:
Eine Mini-Sprache L mit einem Vokabular Σ_L aus vier Elementen

$$\Sigma_L = \{ \quad \textit{Tobias}, \quad \textit{gähnt}, \quad \textit{kennt}, \quad \textit{und} \quad \}$$

Folglich:
Die (unendliche) Menge $\Sigma_L{}^+$ beinhaltet alle nicht-leeren Folgen von Elementen aus Σ_L

formale Sprachen – ein Demonstrationsbeispiel

Länge 1

Tobias.	gähnt.	kennt.	und.

Länge 2

Tobias Tobias.	Tobias gähnt.	Tobias kennt.	Tobias und.
gähnt Tobias.	gähnt gähnt.	gähnt kennt.	gähnt und.
kennt Tobias.	kennt gähnt.	kennt kennt.	kennt und.
und Tobias.	und gähnt.	und kennt.	und und.

Länge 3

Tobias Tobias Tobias.	Tobias Tobias gähnt.	Tobias Tobias kennt.	Tobias Tobias und.
Tobias gähnt Tobias.	Tobias gähnt gähnt.	Tobias gähnt kennt.	Tobias gähnt und.
Tobias kennt Tobias.	Tobias kennt gähnt.	Tobias kennt kennt.	Tobias kennt und.
Tobias und Tobias.	Tobias und gähnt.	Tobias und kennt.	Tobias und und.
gähnt Tobias Tobias.	gähnt Tobias gähnt.	gähnt Tobias kennt.	gähnt Tobias und.
gähnt gähnt Tobias.	gähnt gähnt gähnt.	gähnt gähnt kennt.	gähnt gähnt und.
gähnt kennt Tobias.	gähnt kennt gähnt.	gähnt kennt kennt.	gähnt kennt und.
gähnt und Tobias.	gähnt und gähnt.	gähnt und kennt.	gähnt und und.
kennt Tobias Tobias.	kennt Tobias gähnt.	kennt Tobias kennt.	kennt Tobias und.
kennt gähnt Tobias.	kennt gähnt gähnt.	kennt gähnt kennt.	kennt gähnt und.
kennt kennt Tobias.	kennt kennt gähnt.	kennt kennt kennt.	*usw.*

formale Sprachen – ein Demonstrationsbeispiel

Länge 1

Tobias.	gähnt.	kennt.	und.

Länge 2

Tobias Tobias.	**Tobias gähnt.**	Tobias kennt.	Tobias und.
gähnt Tobias.	gähnt gähnt.	gähnt kennt.	gähnt und.
kennt Tobias.	kennt gähnt.	kennt kennt.	kennt und.
und Tobias.	und gähnt.	und kennt.	und und.

Länge 3

Tobias Tobias Tobias.	Tobias Tobias gähnt.	Tobias Tobias kennt.	Tobias Tobias und.
Tobias gähnt Tobias.	Tobias gähnt gähnt.	Tobias gähnt kennt.	Tobias gähnt und.
Tobias kennt Tobias.	Tobias kennt gähnt.	Tobias kennt kennt.	Tobias kennt und.
Tobias und Tobias.	Tobias und gähnt.	Tobias und kennt.	Tobias und und.
gähnt Tobias Tobias.	gähnt Tobias gähnt.	gähnt Tobias kennt.	gähnt Tobias und.
gähnt gähnt Tobias.	gähnt gähnt gähnt.	gähnt gähnt kennt.	gähnt gähnt und.
gähnt kennt Tobias.	gähnt kennt gähnt.	gähnt kennt kennt.	gähnt kennt und.
gähnt und Tobias.	gähnt und gähnt.	gähnt und kennt.	gähnt und und.
kennt Tobias Tobias.	kennt Tobias gähnt.	kennt Tobias kennt.	kennt Tobias und.
kennt gähnt Tobias.	kennt gähnt gähnt.	kennt gähnt kennt.	kennt gähnt und.
kennt kennt Tobias.	kennt kennt gähnt.	kennt kennt kennt.	*usw.*

pragmatische Fragestellungen $\longrightarrow$ Performanzmodell

Länge 1

Tobias. gähnt. und.

Länge 2

Tobias gähnt.

gähnt Tobias.

Länge 3

Tobias kennt Tobias.

– *Wie heißt du?*
– *Tobias.*

– *Was macht der Tobias gerade?*
– *Gähnt.*

kennt Tobias Tobias.

– *Schau mal, der Tobias gähnt!*
– *Und?*

formale Sprachen – ein Demonstrationsbeispiel

Länge 2

Tobias gähnt.

gähnt Tobias.

Länge 3

Tobias kennt Tobias.

kennt Tobias Tobias.

$L = \{$ **Tobias gähnt.**
Gähnt Tobias?
Tobias kennt Tobias.
Kennt Tobias Tobias?
...
Kennt Tobias Tobias und Tobias ...?
...
Tobias gähnt und Tobias kennt und Tobias ...
... $\}$

formale Grammatik

$$G = \langle V_N,\ V_T,\ R,\ S \rangle,$$

worin

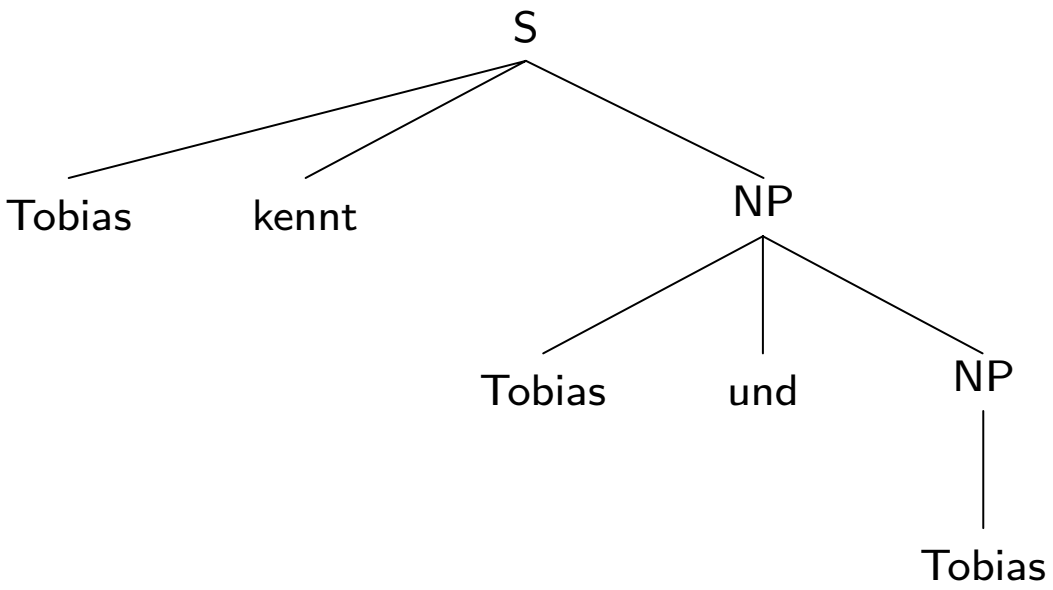
V_N – eine endliche Menge von Nichtterminalsymbolen,

V_T – eine endliche Menge von Terminalsymbolen,

R – eine endliche Menge von Regeln der Form $\alpha \longrightarrow \beta$, wobei α und $\beta \in (V_N \cup V_T)^*$

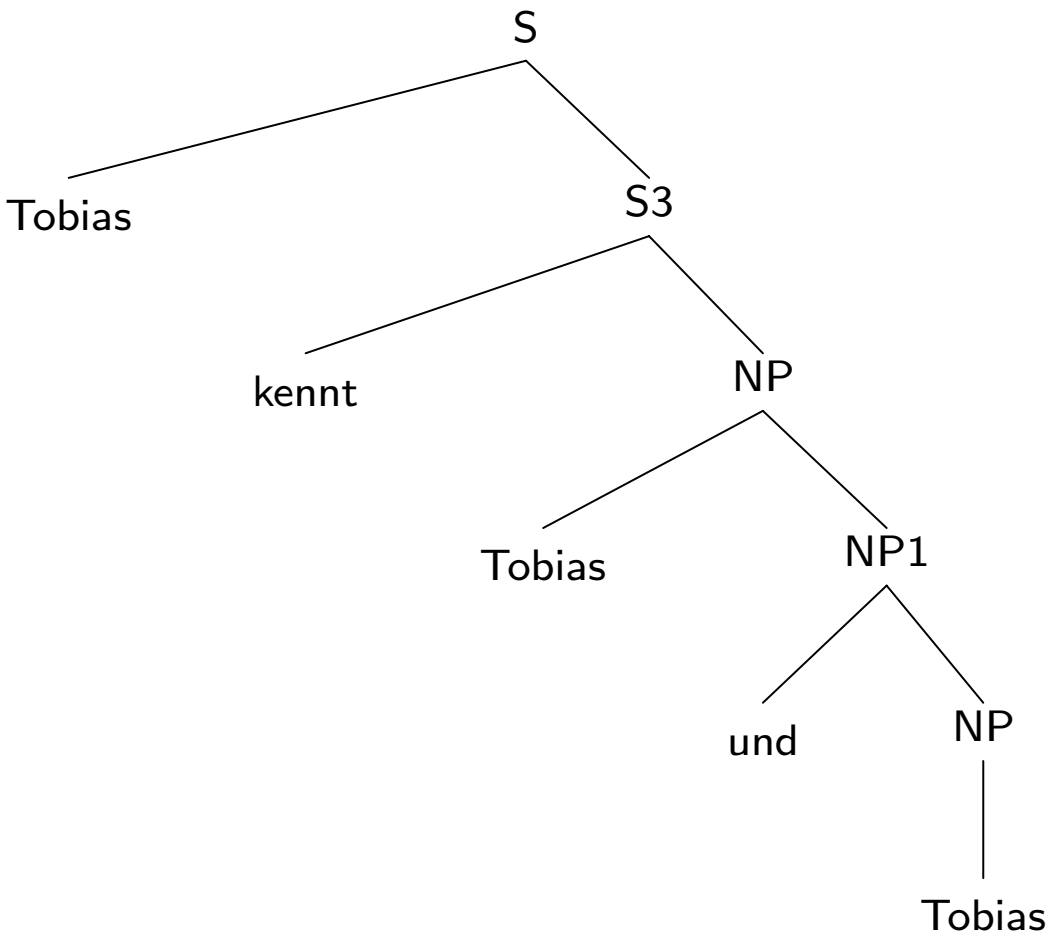
S – das Startsymbol, $S \in V_N$.

kontextfreie Grammatik ...



$$
\begin{aligned}
&G = \langle \\
&\{\ S,\ NP\ \}, \\
&\{\ \mathit{Tobias}, \mathit{gähnt}, \mathit{kennt}, \mathit{und}\ \}, \\
&\{\ S \rightarrow \mathit{Tobias}\ \mathit{gähnt}, \\
&\quad S \rightarrow \mathit{gähnt}\ \mathit{Tobias}, \\
&\quad S \rightarrow \mathit{Tobias}\ \mathit{kennt}\ NP, \\
&\quad S \rightarrow \mathit{kennt}\ \mathit{Tobias}\ NP, \\
&\quad S \rightarrow S\ \mathit{und}\ S, \\
&\quad NP \rightarrow \mathit{Tobias}\ \mathit{und}\ NP, \\
&\quad NP \rightarrow \mathit{Tobias}\ \}, \\
&S\ \rangle
\end{aligned}
$$

reguläre Grammatik ...



$$
\begin{aligned}
S &\rightarrow \textit{Tobias}\ S1,\\
S1 &\rightarrow \textit{gähnt},\\
S1 &\rightarrow \textit{gähnt}\ S4,\\
S1 &\rightarrow \textit{kennt}\ NP,\\
S &\rightarrow \textit{gähnt}\ S2,\\
S2 &\rightarrow \textit{Tobias},\\
S2 &\rightarrow \textit{Tobias}\ S4,\\
S &\rightarrow \textit{kennt}\ S3,\\
S3 &\rightarrow \textit{Tobias}\ NP,\\
NP &\rightarrow \textit{Tobias},\\
NP &\rightarrow \textit{Tobias}\ S4,\\
NP &\rightarrow \textit{Tobias}\ NP1,\\
NP1 &\rightarrow \textit{und}\ NP,\\
S4 &\rightarrow \textit{und}\ S
\end{aligned}
$$

Chomsky's Diktum 1

"[. . .] if you invent a computer language, it doesn't really matter which rules you pick to characterize its expressions; it's the **expressions** *that are the language, not the specific computational system that characterizes them.*

That's not the way natural language works. In natural language there is something in the head, which IS the computational system. The **generative system** *is something real, as real as the liver; the utterances generated are like an epiphenomenon. This is the opposite point of view."*

(Chomsky 1999:8)

Anforderungen an ein Grammatikmodell

Beobachtungsadäquatheit

Korrekte Erfassung der sprachlichen Daten

Beschreibungsadäquatheit

Korrekte Erfassung der Regularitäten eines Sprachsystems

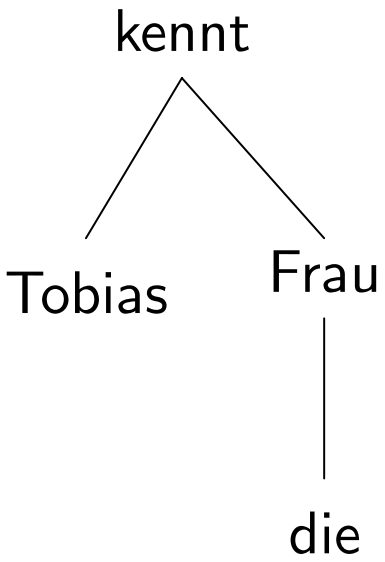
Erklärungsadäquatheit

Hypothese über die universelle Sprachausstattung

dependenzorientierte Ansätze ...

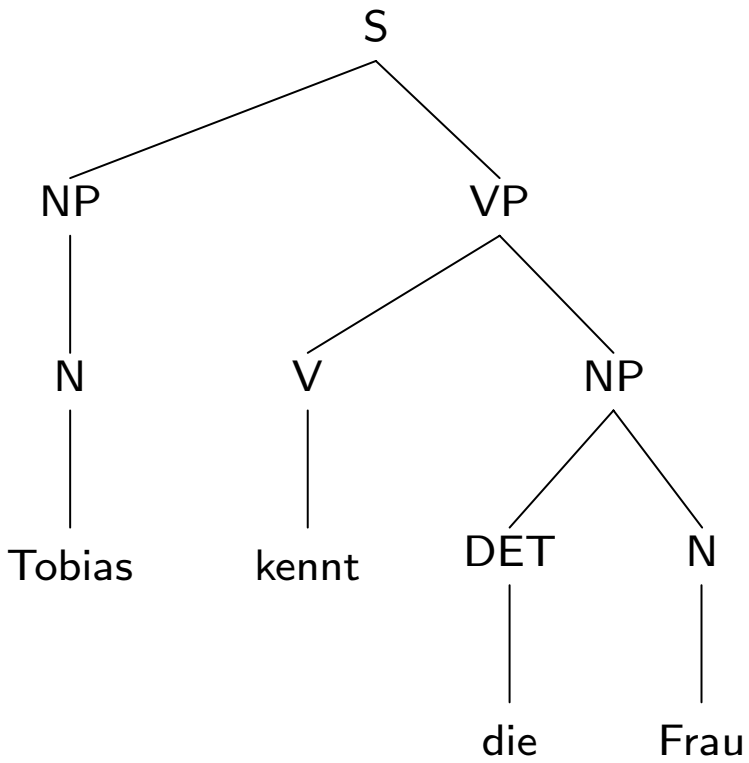
Relationen zwischen Wörtern
(beginnend mit Tesni‘ere:1980)

Tobias kennt die Frau

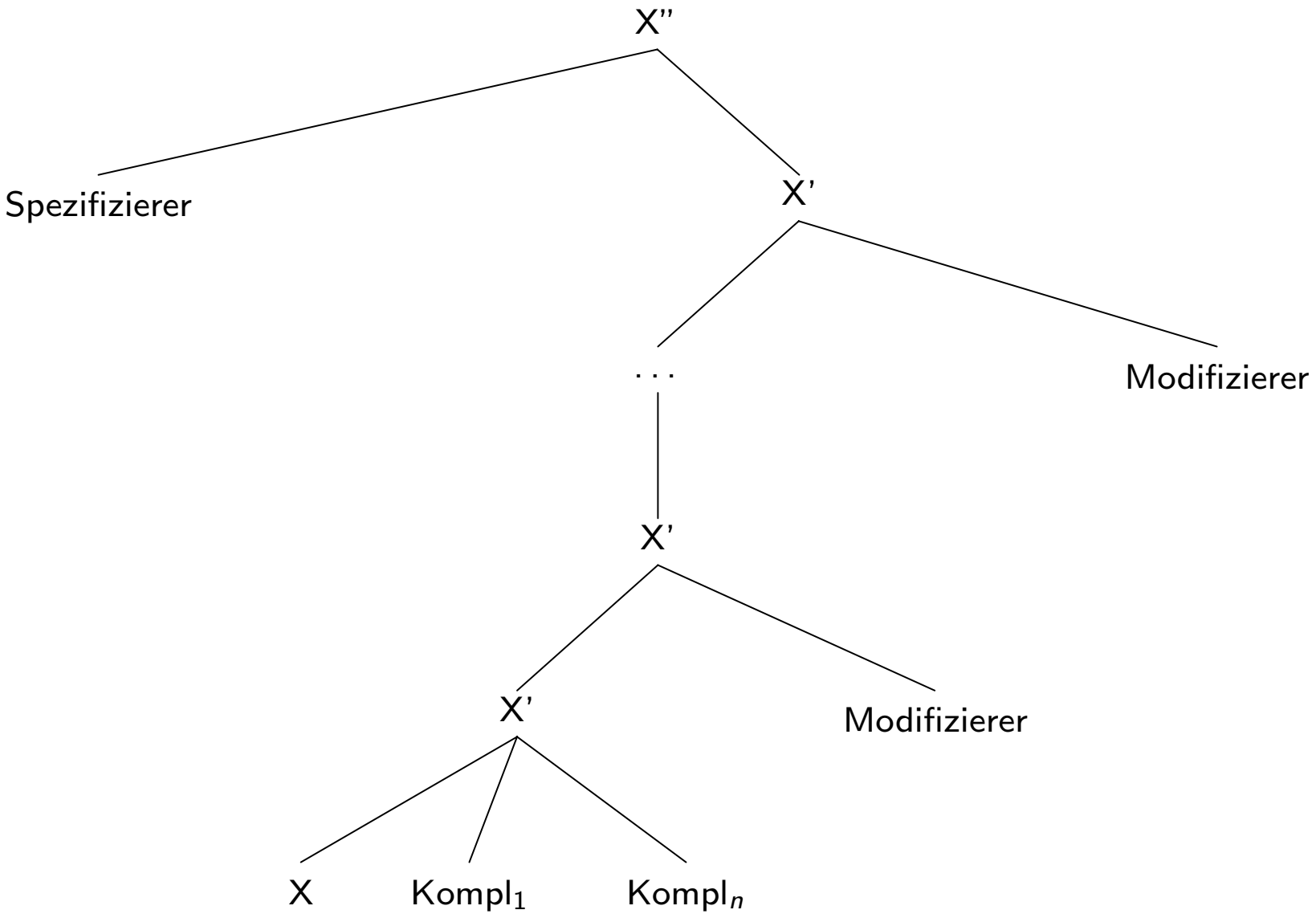


konstituentenorientierte Ansätze ...

Relationen zwischen Konstituenten
(beginnend mit Chomsky:1957)



X-Bar-Theorie (Jackendoff, 1977)



Chomsky's Diktum 2

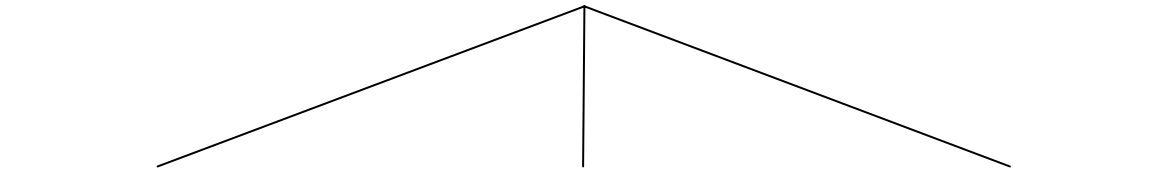
"For formal languages, there is no "right" grammar; it's arbitrary, you pick any one you like. So by analogy, in language the linguist can pick any grammar, depending on one or another concern or interest; the only thing that is real is the **utterances**.

That's a false analogy to start with; human languages are biological objects. What is real – what is in the brain – is a **particular procedure** *for characterizing information about sound, meaning and structural organization of linguistic expressions."*

(Chomsky 1999:16)

Chomsky's Diktum 2

Sprache $L(G)$ als Menge von Ausdrücken



---

die (mentale) **Grammatik**

natürliche Sprache (auch Menge von Äußerungen?)

Grammatikformalismen für das Kompetenzmodell

moderne Unifikationsgrammatiken

- Generalisierte Phrasenstrukturgrammatik (GPSG) (Gazdar, Klein, Pullum und Sag, 1985)
- Lexikalisch Funktionale Grammatik (LFG) (Bresnan, 1982a, 2001; Berman und Frank, 1996; Berman, 2003)
- Head-Driven Phrase Structure Grammar (HPSG) (Pollard und Sag, 1987, 1994; Muller, 1999a, 2002)

didaktische Motivation ...

Warum formal:

ein besseres "Gefühl" für sprachliches Regelwissen

durch Modellierung der zugrundeliegenden Strukturen für Sprachausschnitte des Deutschen

klares Verständnis der Zusammenhänge in der Satzstruktur

durch selbständiges Entwerfen und Testen des Computer-Codes

Behandlung mehr komplexer Fragestellungen

die sonst wegen der zu aufwendigen Mathematik vermieden werden müssten

WERKZEUGE – Prolog und DCGs ...

Prolog:

- deklarative Umgebung
- ermöglicht Abstrahieren von prozeduraler Aspekte
- "eingebauter" Top-Down-Parser mit Backtracking
- eigentlich DIE Programmiersprache für Fragestellungen der Linguistik und der KI

WERKZEUGE – Prolog und DCGs ...

DCG:

- von Stuart Shieber (Shieber 1986) als **Werkzeugformalismus** (*tool formalism*) charakterisiert
- ermöglicht Abstrahieren von programminterner Aspekte
- ermöglicht direkte Modellierung von Phrasenstruktursyntaxen

```
np  -->  det,  n.
```

- ermöglicht Erweiterung v. PSGn durch (komplexe) Merkmale

```
np(Num, Kas)  -->  det(Num, Kas, Gen),  n(Num, Kas, Gen).
```

Modellentwicklung ...

... Generalisierung über Wortarten, Kategorien, lineare Abfolge und Hierarchiebeziehungen

$$
\begin{array}{lll|lll}
n & \rightarrow & \textit{Tobias} & S & \rightarrow & NP\ VP \\
v_{intr} & \rightarrow & \textit{gähnt} & S & \rightarrow & VP\ NP \\
v_{tr} & \rightarrow & \textit{kennt} & NP & \rightarrow & n \\
\textit{conj} & \rightarrow & \textit{und} & VP & \rightarrow & v_{intr} \\
 & & & VP & \rightarrow & v_{tr}\ NP \\
 & & & & & \\
 & & & S & \rightarrow & S\ \textit{conj}\ S \\
 & & & NP & \rightarrow & NP\ \textit{conj}\ NP
\end{array}
$$

Merkmalsstrukturen

Modellierung komplexer Kategorien durch eine Menge von Merkmalsspezifikationen
(Paare grammatischer **Merkmale** und zugehöriger **Werte**)

$$S = \begin{bmatrix} \textsc{Merkmal}_1 & \mathit{Wert}_1 \\ & \dots \\ \textsc{Merkmal}_n & \mathit{Wert}_n \end{bmatrix}$$

---

z.B für Wortform *Hundes*: $N_{sg/mask/gen} = \begin{bmatrix} \textsc{cat} & N \\ \textsc{kasus} & gen \\ \textsc{num} & sg \\ \textsc{genus} & mask \end{bmatrix}$

Merkmalsstrukturen

Werte der Merkmale:

- **atomar**

oder

- **komplex**

Hundes

$$\begin{bmatrix} \text{CAT} & N \\ \text{AGR} & \begin{bmatrix} \text{KASUS} & gen \\ \text{NUM} & sg \\ \text{GENUS} & mask \\ \text{PERS} & 3 \end{bmatrix} \end{bmatrix}$$

Merkmalsstrukturen

Werte der Merkmale:

- **atomar**

oder

- **komplex**

$$Hund$$

$$\begin{bmatrix} \text{CAT} & N \\ \text{AGR} & \begin{bmatrix} \text{KASUS} & nom \vee dat \vee acc \\ \text{NUM} & sg \\ \text{GENUS} & mask \\ \text{PERS} & 3 \end{bmatrix} \end{bmatrix}$$

Merkmalsstrukturen

Unterspezifikation:

In einer Merkmalsstruktur können Merkmale, deren Werte noch nicht bekannt sind, unspezifiziert belassen werden

Spezifikation erfolgt nach Auswertung weiterer Information (Kontext)

$$Katze$$

$$\begin{bmatrix} \text{CAT} & N \\ \text{AGR} & \begin{bmatrix} \text{NUM} & sg \\ \text{GENUS} & fem \end{bmatrix} \end{bmatrix}$$

... kein Info über Kasus (hier egal, wird vom DET bestimmt)
... Person (3) wird vom V in die komplette NP durchgereicht

Merkmalsstrukturen

Unifikation:

Lücken in partiellen Beschreibungen werden im Verlauf der Unifikation gefüllt

Katze

$$\begin{bmatrix} \text{CAT} & \textbf{N} \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \textit{sg} \\ \text{GENUS} & \textit{fem} \end{bmatrix} \end{bmatrix}$$

die

$$\begin{bmatrix} \text{CAT} & \textbf{DET} \\ \text{AGR} & \begin{bmatrix} \text{KASUS} & \textit{nom} \\ \text{GENUS} & \textit{fem} \\ \text{NUM} & \textit{sg} \end{bmatrix} \end{bmatrix}$$

Reihenfolge der Merkmalsstrukturen, die unifiziert werden, spielt keine Rolle!

Wie geht das?

Unifikation:

Lücken in partiellen Beschreibungen werden im Verlauf der Unifikation gefüllt

$$Katze$$

$$\begin{bmatrix} \text{CAT} & \textbf{N} \\ \text{AGR} & \begin{bmatrix} \text{NUM} & \textit{sg} \\ \text{GENUS} & \textit{fem} \end{bmatrix} \end{bmatrix}$$

$$\textbf{NP} \longrightarrow \begin{bmatrix} \text{CAT} & \textbf{DET} \\ \text{AGR} & \begin{bmatrix} \text{KASUS} & \alpha \\ \text{NUM} & \beta \\ \text{GENUS} & \gamma \end{bmatrix} \end{bmatrix} \quad \begin{bmatrix} \text{CAT} & \textbf{N} \\ \text{AGR} & \begin{bmatrix} \text{KASUS} & \alpha \\ \text{NUM} & \beta \\ \text{GENUS} & \gamma \end{bmatrix} \end{bmatrix}$$

Von einfachen zu komplexen Kategorien ...

$$\textbf{NP} \rightarrow \textbf{DET} \quad \textbf{N}$$

$$\begin{bmatrix} \text{CAT} & \textbf{NP} \\ \text{AGR} & \begin{bmatrix} \text{KASUS} & \alpha \\ \text{NUM} & \beta \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \text{CAT} & \textbf{DET} \\ \text{AGR} & \begin{bmatrix} \text{KASUS} & \alpha \\ \text{NUM} & \beta \\ \text{GENUS} & \gamma \end{bmatrix} \end{bmatrix} \quad \begin{bmatrix} \text{CAT} & \textbf{N} \\ \text{AGR} & \begin{bmatrix} \text{KASUS} & \alpha \\ \text{NUM} & \beta \\ \text{GENUS} & \gamma \end{bmatrix} \end{bmatrix}$$

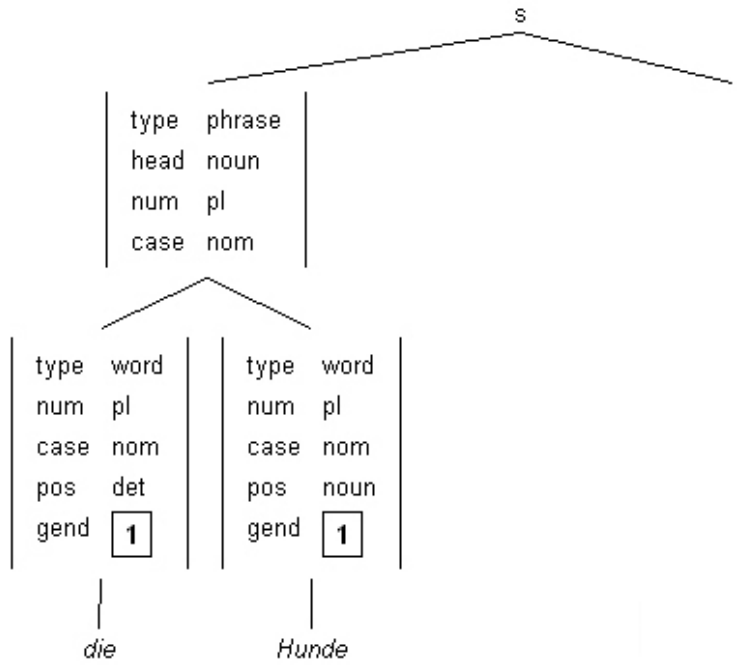
mit $\alpha \in \{nom, gen, dat, akk\}$, $\beta \in \{sg, pl\}$ **und** $\gamma \in \{mask, fem, neut\}$

$$\begin{bmatrix} \text{CAT} & \textbf{NP} \\ \text{AGR} & \begin{bmatrix} \text{KASUS} & nom \\ \text{NUM} & sg \end{bmatrix} \end{bmatrix} \rightarrow \begin{bmatrix} \text{CAT} & \textbf{DET} \\ \text{AGR} & \begin{bmatrix} \text{KASUS} & nom \\ \text{NUM} & sg \\ \text{GENUS} & fem \end{bmatrix} \end{bmatrix} \quad \begin{bmatrix} \text{CAT} & \textbf{N} \\ \text{AGR} & \begin{bmatrix} \text{KASUS} & nom \\ \text{NUM} & sg \\ \text{GENUS} & fem \end{bmatrix} \end{bmatrix}$$

$$die \qquad Katze$$

Beispiel 1: sysana

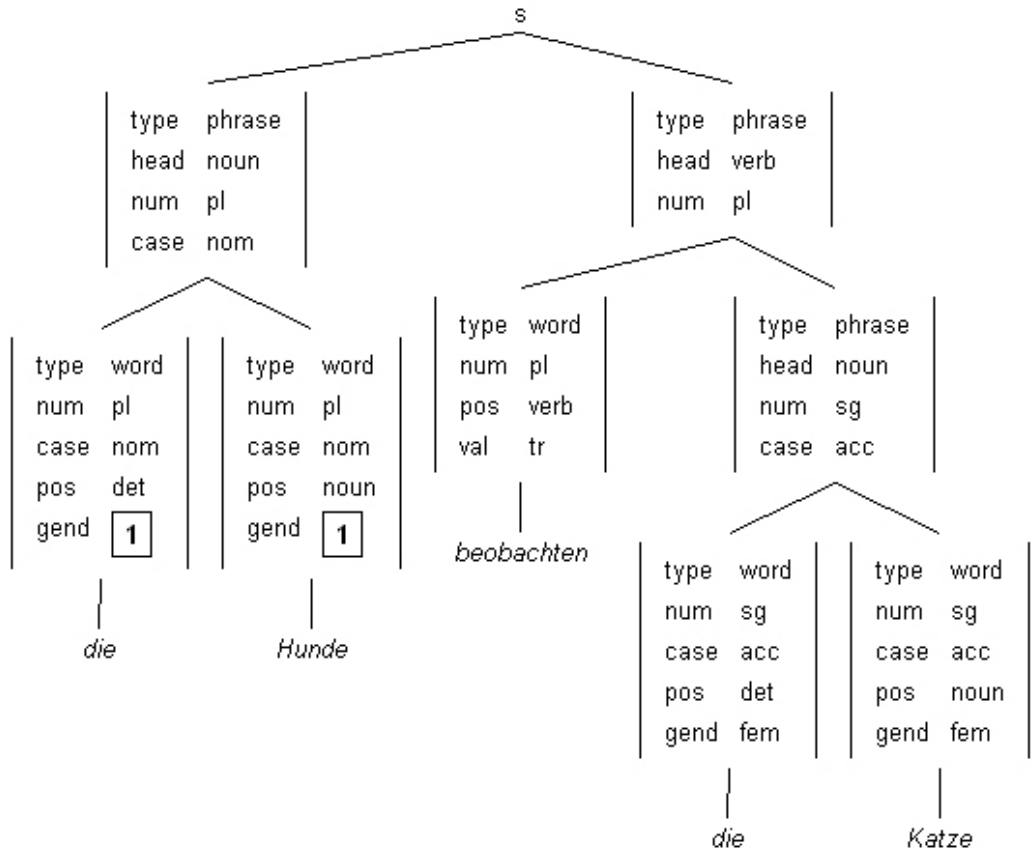
```
% Regel fuer Nominalphrase:

type:phrase
..head:noun
..num:NUMERUS
..case:KASUS
                    -->   type:word
                          ..pos:det
                          ..num:NUMERUS
                          ..case:KASUS
                          ..gend:GENUS
                         ,
                          type:word
                          ..pos:noun
                          ..num:NUMERUS
                          ..case:KASUS
                          ..gend:GENUS
                         .
```




```
% lexikalische Regeln:

type:word                          type:word
..pos:det                          ..pos:noun
..num:pl                           ..num:pl
..case:nom    --> "die".           ..case:nom    --> "Hunde".
```

Beispiel 1: sysana

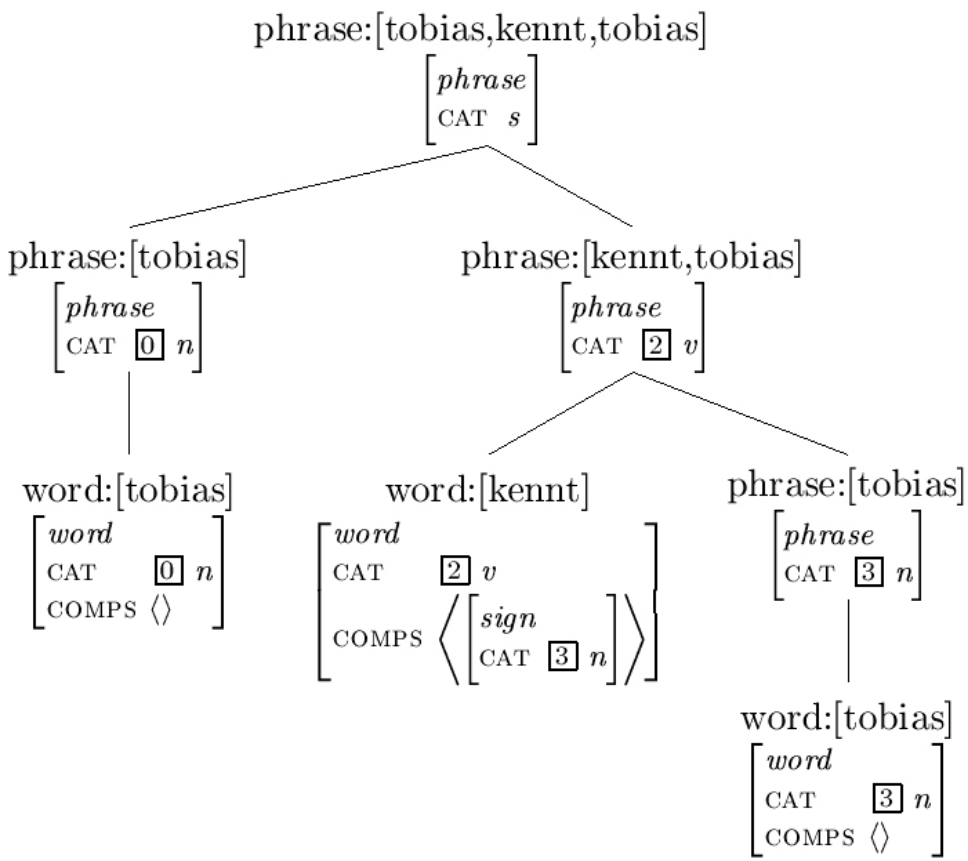
```
% Regel fuer Nominalphrase:

type:phrase
..head:noun
..num:NUMERUS
..case:KASUS
                    -->   type:word
                          ..pos:det
                          ..num:NUMERUS
                          ..case:KASUS
                          ..gend:GENUS
                         ,
                          type:word
                          ..pos:noun
                          ..num:NUMERUS
                          ..case:KASUS
                          ..gend:GENUS
                         .
```




```
% lexikalische Regeln:

type:word                          type:word
..pos:det                          ..pos:noun
..num:pl                           ..num:pl
..case:nom    --> "die".           ..case:nom    --> "Hunde".
```

Beispiel 2: trale

```
s_rule ## (phrase,
          dtrs:[D1,D2],
          cat:s)    ===>
                    cat> (D1,phrase,
                          cat:n),
                    cat> (D2,phrase,
                          cat:v).
xp_rule1 ## (phrase,
          dtrs:[D1],
          cat:Cat) ===>
                    cat> (D1,word,
                          cat:Cat,
                          comps:[]).
xp_rule2 ## (phrase,
          dtrs:[D1,D2],
          cat:Cat) ===>
                    cat> (D1,word,
                          cat:Cat,
                          comps:[(cat:Compl)]),
                    cat> (D2,phrase,
                          cat:Compl).
```



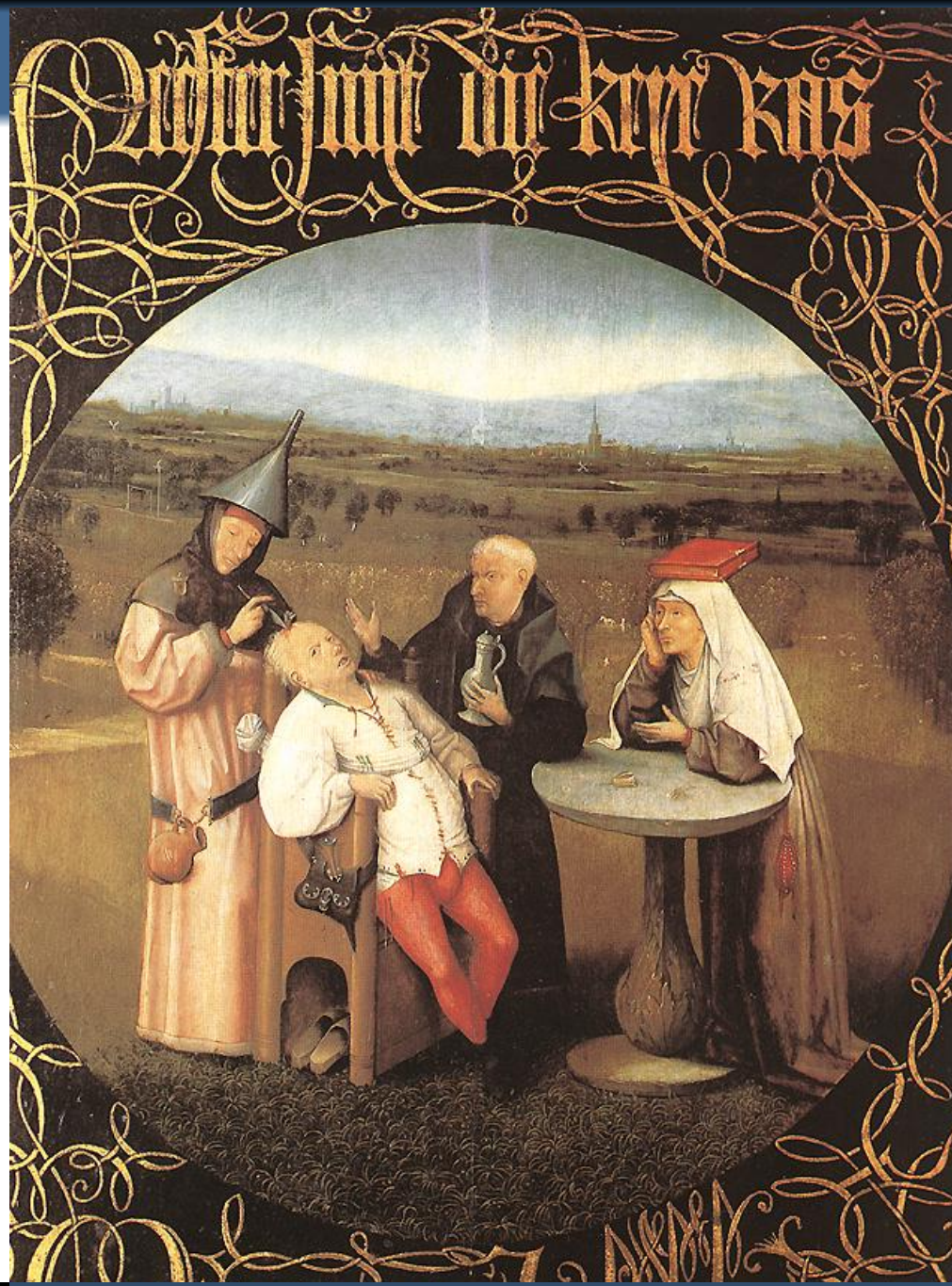

```
% lexikalische Regeln:

tobias --> (word,                kennt --> (word,
           cat:n,                          cat:v,
           comps:[]).                      comps:[(cat:n)]).
```

Lehr-/Lernprozess:

Vermittlung von Wissen an Lernende?

Lehr-/Lernprozess:

Vermittlung von Wissen an Lernende?



Das Steinschneiden
Hieronymus Bosch
15. Jahrhundert

Lehr-/Lernprozess:

Vermittlung von Wissen an Lernende?

oder

gemeinsame **Konstruktion von Wissen**

Student

21. Jahrhundert