Formale Methoden III - Tutorium

Daniel Jettka

22.05.06

Material zu FM3:

(Folien von Christian Ebert, FM3 im SS2005)

http://www.uni-bielefeld.de/lili/personen/cebert/teaching/05fm3/folien.pdf

Inhaltsverzeichnis

1. Getypte AWMn

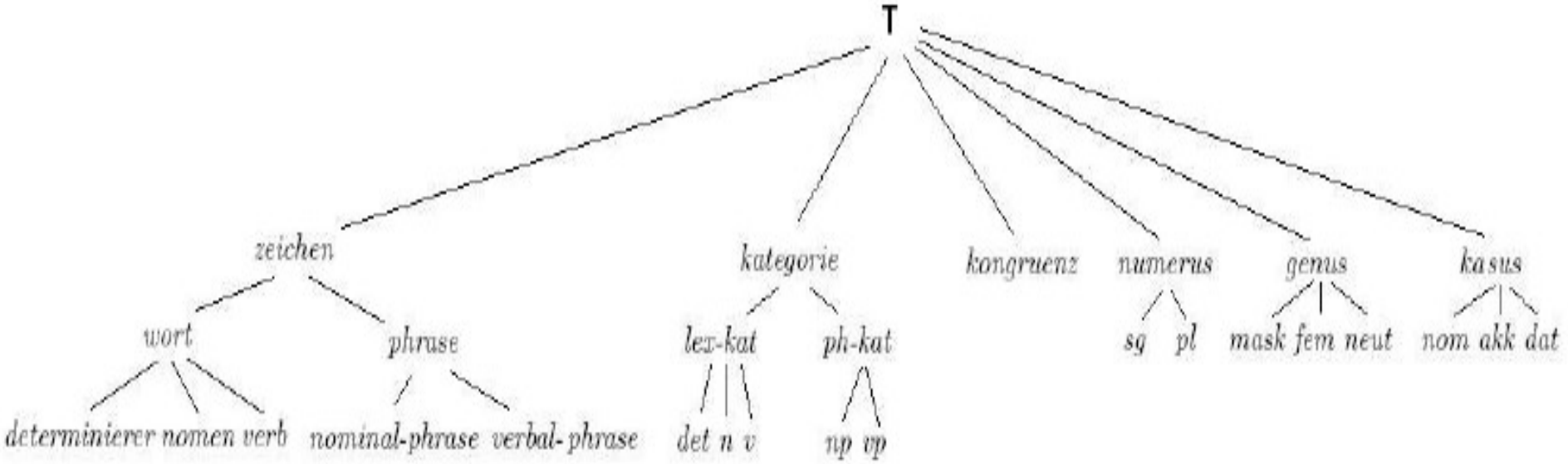
Wozu Typen?

$$\begin{bmatrix} \text{GENUS} & akk \\ \text{KAT} & sg \\ \text{NUMERUS} & \begin{bmatrix} \text{GENUS} & NP \\ \text{KASUS} & pl \end{bmatrix} \end{bmatrix}$$

→ Werte für Attribute einschränken

→ Vererbung (Vereinfachung des Lexikons)

Eine Typhierarchie:



Ange-messenheits-funktionen

app(*zeichen*, KAT) = *kategorie*
app(*nomen*, KAT) = *n*
app(*nominal-phrase*, KAT)= *np*
app(*phrase*, DTR1) = *zeichen*
app(*wort*, KAT) = *lex-kat*
app(*zeichen*, KGR) = *kongruenz*
app(*kongruenz*, GENUS) = *genus*

app(*determinierer*, KAT) = *det*
app(*verb*, KAT) = *v*
app(*verbal-phrase*, KAT) = *vp*
app(*phrase*, DTR2) = *zeichen*
app(*phrase*, KAT) = *ph-kat*
app(*kongruenz*, NUMERUS) = *numerus*
app(*kongruenz*, KASUS) = *kasus*

Fragen:

(a) Wie sähe eine angemessene getypte AWM nach der gegebenen Typhierarchie aus?

(b) Wie sähe eine Merkmalsstruktur auf Grundlage der Typhierarchie aus?

2. LKB

Lexical Knowledge Builder (LKB)

im WWW: http://wiki.delph-in.net/moin/LkbTop

LKB-Download: http://lingo.stanford.edu/ftp/stable/lkb_windows.zip

LKB-Grammatiken: http://lingo.stanford.edu/ftp/stable/lkb_data.tgz

z.B.: ../data/itfs/g2agr

zum Einlesen in LKB „script“ auswählen

Typhierarchie der Grammatik g2agr in LKB:

types.tdl

```
;;; Types

syn-struc := *top* &
[ CATEG cat,
  NUMAGR agr ].

cat := *top*.

s := cat.

np := cat.

vp := cat.

det := cat.

n := cat.

agr := *top*.

sg := agr.

pl := agr.

phrase := syn-struc &
[ ARGS *list* ].

word := syn-struc &
[ ORTH string ].

sg-word := word &
[ NUMAGR sg ].

pl-word := word &
[ NUMAGR pl ].

;;; standard types

string := *top*.

*list* := *top*.

*ne-list* := *list* &
 [ FIRST *top*,
   REST *list* ].

*null* := *list*.
```

Lexikon der Grammatik g2agr in LKB:

lexicon.tdl

```
;;; Lexicon
this := sg-word &
[ ORTH "this",
  CATEG det ].

these := pl-word &
[ ORTH "these",
  CATEG det ].

the := word &
[ ORTH "the",
  CATEG det ].

sleep := pl-word &
[ ORTH "sleep",
  CATEG vp ].

sleeps := sg-word &
[ ORTH "sleeps",
  CATEG vp ].

dog := sg-word &
[ ORTH "dog",
  CATEG n ].

dogs := pl-word &
[ ORTH "dogs",
  CATEG n ].
```

Regeln der Grammatik g2agr in LKB:

rules.tdl

```
;;; Rules

s_rule := phrase &
[ CATEG s,
  NUMAGR #1,
  ARGS [ FIRST [ CATEG np,
                 NUMAGR #1 ],
         REST [ FIRST [ CATEG vp,
                        NUMAGR #1 ],
                REST *null* ]]] .

np_rule := phrase &
[ CATEG np,
  NUMAGR #1,
  ARGS [ FIRST [ CATEG det,
                 NUMAGR #1 ],
         REST [ FIRST [ CATEG n,
                        NUMAGR #1 ],
                REST *null* ]]] .
```