



# Künstliche Intelligenz Sprachverarbeitung mit Prolog

Stephan Schwiebert

WS 2009/2010

Sprachliche Informationsverarbeitung

Institut für Linguistik

Universität zu Köln



# Parsing mit Prolog



# Grammatik

- Wortartenklassifikation, u.a.
  - Nomen (N)
  - Verb (V)
  - Adjektiv (A)
  - Präposition (P)
- Klassifikation von Phrasen, u.a.
  - Nominalphrasen (NP)
  - Verbalphrasen (VP)
  - Adjektivphrasen (AP)
  - Präpositionalphrasen (PP)



# Grammatik

- Verschiedene Ersetzungsregeln können zur Produktion von Sätzen verwendet werden, z.B.

- S → NP VP

- NP → N

- VP → V NP

- VP → V

- N → maus

- N → katze

- V → frisst

- det → die



# Phrasenstrukturregeln

- Links steht genau ein Kategorialsymbol (kontextfreie Grammatik)
- Rechts steht mindestens ein kategorielles bzw. lexikalisches Symbol
- Rekursion ist möglich.



# Syntax-Recognizer in Prolog

- Wie lässt sich ein Syntax-Recognizer programmieren, der die Grammatikalität von Sätzen beurteilen kann?
- Wie lassen sich PS-Regeln in Prolog umsetzen?  
[die, katze, frisst, die, maus]



# PS-Regeln in Prolog

- `n([katze]).`
- `n([maus]).`
- `det([die]).`
  
- `np(NP) :- append(DET, N, NP),  
det(DET),  
n(N).`



# PS-Regeln in Prolog

- `vp(VP) :- append(V, NP, VP),  
v(V),  
np(NP).`
- `vp(VP) :- v(VP).`
- `S(S) :- append(NP, VP, S),  
np(NP),  
vp(VP).`





# Definite Clause Grammar



# Definite Clause Grammar

- DCG-Regeln können genutzt werden, um eine PSG „anwenderfreundlich“ zu modellieren.

$s \rightarrow np, vp.$

$np \rightarrow n.$

$np \rightarrow det, n.$

$vp \rightarrow v, np.$

$vp \rightarrow v.$

$n \rightarrow [katze].$

$n \rightarrow [hund].$



# Definite Clause Grammar

- Intern werden die Regeln in Prolog-Prädikate konvertiert:

$s \rightarrow np, vp.$

$s(A,C) :- np(A,B), vp(B,C).$

- Entsprechend werden sie aufgerufen wie folgt:  
 $s([die,katze,jagt,die,maus],[ ]).$   
 $s([X],[ ]).$



# Definite Clause Grammar

- Weiterhin notwendige bzw. wünschenswert:
  - Berücksichtigung von Kongruenz
  - Syntaktische Struktur
  - Semantische Struktur



# Definite Clause Grammar

- Wünschenswert: Implementation eines Parsers in Prolog, um auf die syntaktische Struktur eines Satzes zugreifen zu können.
- DCGs können um weitere Argumente erweitert werden:

$s(s(NP,VP)) \rightarrow np(NP), vp(VP).$



# Definite Clause Grammar

- Implementation eines Parsers in Prolog:

`s(s(NP,VP)) → np(NP), vp(VP).`

`np(np(DET,N)) → det(DET), n(N).`

...

`n(n(katze)) → [katze].`

`n(n(maus)) → [maus].`

- Aufruf des build-in-Prädikats `listing(s)` zeigt die interne Struktur der Regeln:

`s(s(A, C), B, E) :-`

`np(A, B, D),`

`vp(C, D, E).`



# Definite Clause Grammar

- Problem: Eine Erweiterung um die Regeln

det  $\rightarrow$  [der].

n  $\rightarrow$  [hund].

führt zur ungewollten Erkennung fehlerhafter Sätze:

\*die schlage jagt der katze

\*die hund jagt der schlange

- Kongruenz bzgl. Genus, Numerus und Kasus muss überprüft werden können.



# Definite Clause Grammar

- Naiver Ansatz:

Hinzufügen spezialisierter Regeln:

$np \rightarrow detM, nM.$

$np \rightarrow detF, nF.$

$detM \rightarrow [der].$

$detF \rightarrow [die].$

$nM \rightarrow [hund].$

$nF \rightarrow [katze].$





# Definite Clause Grammar

- Besserer Ansatz:

Erweiterung um zusätzliche Parameter:

s → np(nom,\_), vp.  
np(K,G) → det(K,G), n(K,G).  
vp → v, np(akk,\_).

n(nom,f) → [katze].

n(akk,f) → [katze].

n(nom,m) → [hund].

usw.



# Definite Clause Grammar

- Kombination von Prolog-Code und DCG:

```
n(Kasus, Genus, Lexem) --> [Lexem],  
    {lex(Lexem, n, Kasus, Genus)}.
```

```
det(Kasus, Genus, Lexem) --> [Lexem],  
    {lex(Lexem, det, Kasus, Genus)}.
```

```
lex(katze, n, nom, f).
```

```
lex(katze, n, akk, f).
```

```
lex(schlange, n, nom, f).
```



# Definite Clause Grammar

- Verbindung zwischen Parser und Prolog-Logik:  
Extrahieren von Subjekt, Verb und Objekt, Aufruf eines Prädikats der Art  
  
Verb(Subjekt,Objekt).



# Definite Clause Grammar

- Erweitern Sie die DCG aus Beispiel 2 um Regeln, durch die auch Sätze wie  
„die katze jagt die schlange und die maus“  
verarbeitet werden können.
- Welches Problem taucht auf?



# Definite Clause Grammar: Fazit

- Für einfache Aufgaben sinnvoll
  - (größtenteils) kontextfreie Sprachen
  - beschränktes Vokabular (Computerspiele)
- Für komplexere Projekte nicht einsetzbar
  - Linksrekursion
  - Ineffizientes Backtracking



# Hausaufgaben

- Erweitern Sie das Beispielprogramm psg5 so, dass Nominalphrasen Adjektive enthalten können.
  - „Die große Schlange“
  - „Die schnelle Maus“  
usw.
- Erweitern Sie es anschließend um unbestimmte Artikel („Ein“, „Eine“...) und sorgen sie dafür, dass Adjektive korrekt dekliniert werden
  - „Ein großer Hund“
  - „Der große Hund“  
usw.



# Hausaufgaben

Erweitern Sie das Beispielprogramm so, dass auch Sätze wie

- Die Katze jagt.
- Die Katze jagt die Maus.
- Die Katze schläft.
- Die Maus gähnt.

generiert werden, aber nicht

- \*Die Katze schläft den Hund.
- \*Die Katze gähnt die Maus.