Übungen 10: DCG

Programmiertechniken in der Computerlinguistik I · Wintersemester 2004/2005

Um keine Zeit mit Tippen und Tippfehlerkorrektur zu verschwenden, findest du die Programmtexte dieser Übung unter http://www.cl.unizh.ch/siclemat/lehre/ws0405/pcl1/uebung10.txt

1. Parsen und Beweisen

Gegeben sei folgende DCG (mit exakt dieser Klauselreihenfolge):

```
np --> det, n.                    det --> [the].
vp --> v.                         n --> [cat].
vp --> v, np.                     n --> [dog].
                                  v --> [sees].
                                  v --> [sings].
```

a) Welche einzelnen Schritte nimmt Prolog vor bis zum Beweis der Anfrage?

```
?- vp([sees,the,dog], []).
```

Zeichne den entsprechenden Such-/Beweisbaum.

b) Notiere die Ableitungsschritte der Satzformen, die der Prolog-Parsing-Strategie entsprechen, in der ⇒–Notation. Betrachte VP als Startsymbol!

2. Das reguläre Gelächter

Schreibe eine Lachgrammatik der Sprache L⊆ {hi,ha,ho}* in DCG, mit der alle Gelächter folgender Art beschrieben werden:

a) zuerst kommen beliebig viele his
b) dann kommen beliebig viele has
c) dann kommen beliebig viele hos

Die Grammatik sollte dann etwa ein Liste wie

`[hi,hi,ha,ha,ha,ho,ho,ho,ho]` oder `[ha,ho]` oder `[]`

als zulässiges Gelächter erkennen und ein syntaktisch falsches Lachen wie

`[ho,ho,hi,hi]`

zurückweisen.

Tipp: Denke daran, dass ε die Folge von 0 Elementen von Terminalsymbolen ist.

3. Kontextfreie Grammatik erstellen

a) Schreibe eine Grammatik für einfache deutsche Sätze. Benutze folgende Symbole:

Symbol	**Bedeutung**	**Beispiel(e)**
S	Satz	die Katze spielt mit dem gelben Ball
NP	Nominal-Phrase	die Katze · dem gelben Ball · die Katze, die spielt · die Katze mit dem Ball
VP	Verbal-Phrase	spielt mit dem Ball · singt
PP	Präpositional-Phrase	mit dem Ball
RelOpt	Optionaler Relativsatz	die singt · der rollt
N	Nomen	Ball · Katze
V	Verb	spielt · singt · rollt
P	Präposition	mit · unter
A	Adjektiv	gelben · rote · blaue
Det	Artikel (*Determiner*)	der · die · dem
RelPron	Relativpronomen	der · die

b) Eine Grammatik heisst *übergenerierend,* wenn zu ihrer Sprache auch Sätze gehören, die nicht Teil der modellierten natürlichen Sprache sind. Umgekehrt heisst eine Grammatik *untergenerierend,* wenn sie bestimmte Sätze nicht akzeptiert, auch wenn diese Bestandteil der natürlichen Sprache wären, die mit der Grammatik modelliert wird.
Notiere dir mindestens zwei Wortketten, welche nach obiger Grammatik syntaktisch korrekt wären, aber trotzdem nicht Teil der deutschen Sprache sind (Übergenerierung).
Notiere dir mindestens zwei Sätze des Deutschen aus den Wörtern unserer Grammatik, welche durch einfache Modifikation der syntaktischen Regeln unserer Grammatik zugelassen werden könnten (Untergenerierung).

c) Überprüfe für einige Beispielsätze, ob sie der Grammatik aus (a) entsprechen, indem du die Grammatik als DCG formulierst und den in Prolog eingebauten DCG-Parser benutzt. Teste deine Bespiele für Über- und Untergenerierung aus (b) darauf, ob der Prolog-DCG-Parser deine Meinung teilt.
Schreib-Tipp: Prolog unterstützt die übliche abkürzende Schreibweise für Regeln mit gleicher RHS:

```
det --> [der] | [die] | [dem].
```

d) Freiwillig: Wieviele Sätze lässt deine Grammatik zu? Definiere ein Prädikat show_phrases/2, mit dem du sämtliche Phrasen der Grammatik bis zu einer bestimmten Länge als Wort-Listen auf den Bildschirm schreiben kannst. Jede Phrase soll auf eine eigene Zeile kommen. Beispiel:

```
?- show_phrases(np, 3).
[die,Katze]
…
[der,Ball]
…
[die,gelben,Katze]
…
```