Aufgabenblatt 7
Computerlinguistische Anwendungen

Universität München, CIS, SS 2011

Hans Leiß

Abgabetermin: Mi, 6.7. 2011

Aufgabe 7.1 (Generierung von natürlichsprachlichen Ausdrücken)

Nach dem Laden mit `?- [grammatik,astronomie].` können wir (außer mit unserem `parse/0`) auch mit

```
?- np([Def,Pers,Gen],[Num,Kas],Baum,+Atomliste,[]).
```

Prolog auffordern, zu einer *gegebenen* Atomliste eine Analyse als Nominalphrase zu suchen.

Wir wollen aber auch testen können, welche Nominalphrasen die Grammatik *erzeugen* kann. Da die Grammatik unendlich viele Nominalphrasen erkennt, kann man nicht alle erzeugen lassen.

a) Welche Regeln der Grammatik führen dazu, daß es unendlich viele Nominalphrasen gibt? (Genauer: was ist die kleinste Menge von Regeln, die man weglassen muß, damit die Grammatik nur endlich viele Nominalphrasen erkennt?) (3 Punkte)

b) Wie muß man `np/5` aufrufen, damit Prolog eine Atomliste ausgibt, die eine definite Nominalphrase im Nominativ aus zwei Wörtern ist? (2 Punkte)

   Wenn man das testet, erhält man nach einer Reihe von Lösungen eine Fehlermeldung. Welche Grammatikregel verursacht diese Fehlermeldung? (1 Punkt)

c) Kommentiere diese Grammatikregel nach b) aus und lade die Grammtik neu.

   Schreibe ein Prolog-Programm `erzeuge_np/1`, das zu einer Zahl $N > 0$ alle Nominalphrasen (im Nominativ) der Länge N erzeugt, indem es

   (i) zuerst eine Atomliste der Länge N erzeugt, die eine Nominalphrase bildet,
   (ii) dann diese Atome mit Leerzeichen zu einem langen Atom verbindet,
   (iii) dieses Atom in eine neue Zeile schreibt,
   (iv) und das solange wiederholt, bis alle Nominalphrasen der Länge N ausgegeben sind.

   Teste das Programm für kleine N ($N = 1, 2, 3, 4$). (6 Punkte)

   Hinweis: Mache es erst für festes N, sagen wir $N = 3$. Mit `length(-Liste,+N)` bekommt man eine Liste von `N` unbelegten Variablen.

d) Schreibe ein Programm, das alle Aussagesätze (= definite Sätze in Verbzweitstellung) mit einem Vollverb und Satzlänge 5 erzeugt. (6 Punkte)

Hinweis: der Unterschied zu c) ist, daß nicht mehr alle Satzregeln durchlaufen werden dürfen. Wie sieht die Struktur der gesuchten Sätze aus, und wie kann man Prolog dazu bringen, nur Sätze mit dieser Struktur zu erzeugen?

Aufgabe 7.2 (Prolog-Nachhilfe) Schreibe ein Prolog-Prädikat, `gute_Teilterme(+Term,-Teilterme)`, das zu einem Term die Liste aller *guten* Teilterme ausgibt. Ein Teilterm heiße *gut*, wenn er keine Variable und keine Zahl enthält. Wie lautet Ihr Programm und was ergibt es bei folgendem Aufruf?

⟨ *Testaufruf* ⟩≡

```
gute_Teilterme(f(X,g(10,h(a,b)),k(23),k(a)),Teilterme).
```

Hinweis: Verwende einen „Akkumulator“, um beim Durchlaufen des Terms die guten Teilterme aufzusammeln.

Bemerkung Die Tutorin hat mir gerade geschrieben, daß sie die Aufgabe 5 mit Ihnen am Mittwoch in der Fragestunde besprochen hat. In der Tafelübung gestern haben Sie so getan, als sei Ihnen völlig unklar, wie man die Aufgabe 5 löst, sodaß ich alles ausführlich vorgemacht und erklärt habe, obwohl wir im Stoff schon ein gutes Stück hinter dem Plan sind.

Ich fühle mich belogen und bespreche in Zukunft in der Tafelübung erstmal keine weiteren Aufgaben! Nicht nur ich, sondern auch diejenigen, die meinen, sie müssten alles doppelt erklärt bekommen, haben auch eine Verantwortung den Stundenten gegenüber, die sich anstrengen und etwas Neues lernen wollen!

Nachtrag: „belogen“ ist vielleicht zu hart gesagt; aber es hätte ja mindestens jemand sagen können, daß die Aufgabe schon von der Tutorin besprochen wurde, aber immer noch ein bestimmter Punkt unklar ist. Dann hätten wir uns diesen Punkt beschränken können und nicht die ganze Stunde verwenden müssen.