Aufgabenblatt 4
Computerlinguistische Anwendungen

Universität München, CIS, SS 2012

Hans Leiß

Abgabetermin: Mi, 24.5.2012

Aufgabe 4.1 Die Grammatik in `Beispiele/programmierer.pl` berücksichtigt die im Deutschen notwendige Kongruenz morphologischer Merkmale nicht.

Dann hatten wir (vgl.Vorlesungsfolien) diese Grammatik um einige Merkmale erweitert:

⟨*Beispiel/dcg.merkmale.pl*⟩≡

```
% ----------Beispiele/dcg.merkmale.pl-------------------------
% Nur Pers = 3, daher ignoriert.

startsymbol(s([_Temp])).  % für später

s([Temp]) -->
    np([_Gen],[Num,nom]), vp([Temp,Num]).
np([Gen],[Num,Kas]) -->
    det([Gen,Num,Kas]), n([Gen],[Num,Kas]).
vp([Temp,Num]) -->
    ( v([Temp,Num])
    ; v([Temp,Num]), np([_Gen],[_Num,akk])
    ).

det([Gen,sg,Kas]) -->
    ( [der], { Gen = mask, Kas = nom }
    ; [das], { Gen = neut, (Kas = nom; Kas = akk) }
    ).
n([mask],[Num,Kas]) -->
    ['Programmierer'], { (Num = sg ; Num = pl),
                         (Kas = nom; Kas = akk) }.
n([neut],[sg,Kas]) -->
    ['Programm'], { (Kas = nom; Kas = akk) }.

v([praes,sg]) --> [steht].
v([praet,sg]) --> [schrieb].
```

```
% ------------------------------------------------------------
```

a) Ergänze eine Regel für die Vollform `die` des Artikels im Plural, die alle korrekten Genus- und Kasusmöglichkeiten erfaßt, und eine Regel für die Verbform `schrieben` (ohne Personen-Merkmal). (3 Punkte)

b) Ergänze die Kategorie `v` um ein Merkmal zur Unterscheidung von Verbarten (durch die Angabe der Komplementrahmen `[nom]` bei intransitiven bzw. `[nom,akk]` bei transitiven Verben). Passe die `v`-Regeln und die `vp`-Regel so an, daß intransitive Verben nur ohne Objekt-`np` und transitive nur mit Akkusativ-Objekt verwendbar sind (3 Punkte)

c) Geben Sie die Prolog-Klausel an, in die die angepasste `vp`-Regel beim Einlesen der Datei (nach Laden von `Parser/term_expansion.pl`) übersetzt wird. (3 Punkte)

d) Geben Sie die Aufrufe an, die zum Parsen der Eingaben

```
die Programmierer schrieben das Programm.
das Programm schrieben die Programmierer.
```

nötig sind, und die von Prolog erzeugten Syntaxbäume (als Prolog-Terme). (4 Punkte)

Aufgabe 4.2 In der (obigen) Grammatik `Beispiele/dcg.merkmale.pl` wird die Kongruenz von Artiken und Nomen in Genus, Numerus und Kasus in den Nominalphrasen durch gleiche Merkmalvariablen sichergestellt, ebenso die Kongruenz von Subjektnominalphrase und Verbalphrase im Numerus in Sätzen.

Angenommen, wir wollen eine Syntaxanalyse, die Kongruenzfehler erkennt, aber tolieriert. Wenn z.B. in einer Nominalphrase Artikel und Nomen verschiedenen Kasus haben, sollte bei der entstandenen NP der Kasus einen Fehlerwert `error` erhalten. Beim Satz (wo kein Numerus an der Wurzel vorgesehen ist) sollte nur eine Warnung ausgegeben werden, wenn die Nominalphrase und die Verbalphrase verschiedene Numeruswerte haben.

(a) Ändern Sie die beiden Grammatikregeln für Sätze und Nominalphrasen so, daß (der Regelkopf und) die Konstituenten jeweils unterschiedliche Variablen (z.B. `GenNP`, `GenDet`, `GenN`) haben, und fügen Sie eine Nebenbedingung `{...}` ein, die bei Inkongruenz der Konstituenten in einem Merkmal dem Kopf den Merkmalwert `error` gibt bzw. beim Numerus des Satzes die Warnung ausgibt.

Ein Ausdruck sollte also so behandelt werden, daß zuerst seine Konstituenten unabhängig von einander analysiert werden und dabei pro Lesart gewisse Merkmalwerte erhalten, und daß anschließend durch die Nebenbedingung `{ kongruenz(?Merkmalvariable,...) }` die Kongruenz festgestellt oder ein Fehlerwert bzw. eine Warnung erzeugt wird. (Fehler, die nur durch "unpassende" Wahl lexikalischer Lesarten entstehen –wenn es also auch eine "passende" Wahl gibt– brauchen Sie nicht zu unterdrücken.) (4 Punkte)

(b) Laden Sie `Parser/term_expansion.pl` und `Parser/showTree.pl` und geben Sie den von `showTree` in Textform formatierten Syntaxbaum einer Nominalphrase mit einem Kongruenzfehler an. (3 Punkte)