

Lösungen zum Aufgabenblatt 4

Syntax natürlicher Sprachen

Universität München, CIS, WS 2016/17

Hans Leiß

Abgabetermin: Mi, 30.11.2016

Aufgabe 4.1

Unter den Sätzen `beispielsätze.txt` zum ersten Aufgabenblatt finden sich die folgenden:

Derartige Fälle sprengten den Rahmen der geltenden Finanzverfassung.
Deren technische Sicherung basiere auf gesetzlichen Vorgaben.
Des Autors Vorliebe gilt den weiblichen Figuren.
Der Mann hat Freude an seinem Job.

Nur für manche der darin auftretenden Wörter gibt es Lexikoneinträge in `lexikon_np.pl` (d.h. lexikalische DCG-Regeln wie `art([sw,sg],[fem,nom]) --> [die].`). Lege eine zweite Lexikondatei `lexikon-beispielsätze.pl` mit Einträgen für die in `lexikon_np.pl` fehlenden Wörter an, bis sich die Nominalphrasen in den obigen Sätzen mit

```
?- ['np.folien.pl', 'lexikon-beispielsätze.pl'].
```

syntaktisch analysieren lassen (wenn `DCGparser/parser.pl` geladen ist).

Lege eine Datei `beispielnominalphrasen.txt` aller Nominalphrasen der obigen Beispiele an und lasse sie mit `?- parse('beispielnominalphrasen.txt').` analysieren.

Sie können eine Ausgabedatei mit den Syntaxbäumen erzeugen durch:

```
?- ['np.folien.pl', 'lexikon-beispielsätze.pl'].  
?- tell('beispielnominalphrasen.ana'), parse('beispielnominalphrasen.txt'), told.
```

Dadurch wird eine Ausgabedatei `'beispielnominalphrasen.ana'` angelegt (oder überschrieben!) und die Ausgaben des Parsers dorthin geschrieben. Durch `told.` wird die Ausgabe wieder an den Bildschirm umgelenkt.

Bem. Eine Regel mit Genitivattribut als Determinator hat die Grammatik absichtlich nicht, vgl. Folie 116 zu linksrekursiven Regeln. Daher wird `des Autors Vorliebe` nicht erkannt. Aus dem gleichen Grund gibt es keine Regel für koordinierte Nominalphrasen wie `der Job und die Freude`. (Man könnte aber eine Regel für koordinierte Nominalphrasen einbauen, in der die Artmerkmale der linken `np`-Konstituente *echte* Genus- und Numeruswerte sind und bei der entstehenden Nominalphrase die Pseudowerte - verlangt werden.)

Lösung von Aufgabe 4.1 Lexikoneinträge, die die (einige der) Adjektive und Nomen der Beispielsätze enthalten, sind, geordnet nach den Wortarten des bisherigen Lexikons `lexikon_np.pl`:

```
% Lexikon zu Aufgabe 4.1 (lexikon-beispielsaetze.pl)
```

```
% (definites) Personalpronomen:
```

```
% Eigennamen pn([Gen],[Kas]).
```

```
% Zaehlbare Nomen (count nouns):
```

```
cn([fem],[sg,nom]) --> ['Finanzverfassung'].
cn([fem],[sg,gen]) --> ['Finanzverfassung'].
cn([fem],[sg,dat]) --> ['Finanzverfassung'].
cn([fem],[sg,akk]) --> ['Finanzverfassung'].
```

```
cn([fem],[pl,nom]) --> ['Finanzverfassungen'].
cn([fem],[pl,gen]) --> ['Finanzverfassungen'].
cn([fem],[pl,dat]) --> ['Finanzverfassungen'].
cn([fem],[pl,akk]) --> ['Finanzverfassungen'].
```

```
cn([fem],[sg,nom]) --> ['Sicherheit'].
cn([fem],[sg,gen]) --> ['Sicherheit'].
cn([fem],[sg,dat]) --> ['Sicherheit'].
cn([fem],[sg,akk]) --> ['Sicherheit'].
```

```
cn([fem],[pl,nom]) --> ['Sicherungen'].
cn([fem],[pl,gen]) --> ['Sicherungen'].
cn([fem],[pl,dat]) --> ['Sicherungen'].
cn([fem],[pl,akk]) --> ['Sicherungen'].
```

```
cn([fem],[sg,nom]) --> ['Vorgabe'].
cn([fem],[sg,gen]) --> ['Vorgabe'].
cn([fem],[sg,dat]) --> ['Vorgabe'].
cn([fem],[sg,akk]) --> ['Vorgabe'].
```

```
cn([fem],[pl,nom]) --> ['Vorgaben'].
cn([fem],[pl,gen]) --> ['Vorgaben'].
cn([fem],[pl,dat]) --> ['Vorgaben'].
cn([fem],[pl,akk]) --> ['Vorgaben'].
```

```
cn([fem],[sg,nom]) --> ['Vorliebe'].
cn([fem],[sg,gen]) --> ['Vorliebe'].
cn([fem],[sg,dat]) --> ['Vorliebe'].
cn([fem],[sg,akk]) --> ['Vorliebe'].
```

```
cn([fem],[pl,nom]) --> ['Vorlieben'].
cn([fem],[pl,gen]) --> ['Vorlieben'].
```

cn([fem],[pl,dat]) --> ['Vorlieben'].
 cn([fem],[pl,akk]) --> ['Vorlieben'].

cn([fem],[sg,nom]) --> ['Figur'].
 cn([fem],[sg,gen]) --> ['Figur'].
 cn([fem],[sg,dat]) --> ['Figur'].
 cn([fem],[sg,akk]) --> ['Figur'].

cn([fem],[pl,nom]) --> ['Figuren'].
 cn([fem],[pl,gen]) --> ['Figuren'].
 cn([fem],[pl,dat]) --> ['Figuren'].
 cn([fem],[pl,akk]) --> ['Figuren'].

cn([mask],[sg,nom]) --> ['Rahmen'].
 cn([mask],[sg,gen]) --> ['Rahmens'].
 cn([mask],[sg,dat]) --> ['Rahmen'].
 cn([mask],[sg,akk]) --> ['Rahmen'].

cn([mask],[pl,nom]) --> ['Rahmen'].
 cn([mask],[pl,gen]) --> ['Rahmen'].
 cn([mask],[pl,dat]) --> ['Rahmen'].
 cn([mask],[pl,akk]) --> ['Rahmen'].

cn([mask],[sg,nom]) --> ['Fall'].
 cn([mask],[sg,gen]) --> ['Falles'].
 cn([mask],[sg,dat]) --> ['Fall'] ; ['Falle'].
 cn([mask],[sg,akk]) --> ['Fall'].

cn([mask],[pl,nom]) --> ['Fälle'].
 cn([mask],[pl,gen]) --> ['Fälle'].
 cn([mask],[pl,dat]) --> ['Fällen'].
 cn([mask],[pl,akk]) --> ['Fälle'].

cn([mask],[sg,nom]) --> ['Autor'].
 cn([mask],[sg,gen]) --> ['Autors'].
 cn([mask],[sg,dat]) --> ['Autor'].
 cn([mask],[sg,akk]) --> ['Autor'].

cn([mask],[pl,nom]) --> ['Autoren'].
 cn([mask],[pl,gen]) --> ['Autoren'].
 cn([mask],[pl,dat]) --> ['Autoren'].
 cn([mask],[pl,akk]) --> ['Autoren'].

cn([mask],[sg,nom]) --> ['Job'].
 cn([mask],[sg,gen]) --> ['Jobs'].
 cn([mask],[sg,dat]) --> ['Job'].
 cn([mask],[sg,akk]) --> ['Job'].

```

cn([mask],[pl,nom]) --> ['Jobs'].
cn([mask],[pl,gen]) --> ['Jobs'].
cn([mask],[pl,dat]) --> ['Jobs'].
cn([mask],[pl,akk]) --> ['Jobs'].

% Massennomen: mn([Gen],[sg,Kas]).

% Nominalisierte Adjektive na([Kompl,Gen],[Num,Kas])

% Nominalisierte Verben nv([Kompl,Gen],[Num,Kas])

% Nomina mit Argument: n([Kompl,Gen],[Num,Kas])

n([[an,dat],fem],[sg,nom]) --> ['Freude'].
n([[an,dat],fem],[sg,gen]) --> ['Freude'].
n([[an,dat],fem],[sg,dat]) --> ['Freude'].
n([[an,dat],fem],[sg,akk]) --> ['Freude'].

n([[an,akk],mask],[sg,nom]) --> ['Glaube'].
n([[an,akk],mask],[sg,gen]) --> ['Glaube'].
n([[an,akk],mask],[sg,dat]) --> ['Glaube'].
n([[an,akk],mask],[sg,akk]) --> ['Glaube'].

praeposition(an,dat).

% Determinatoren: Art, Qnt, Poss, (NPgen), Pron

% Artikel art([ADeklination,Numerus],[Genus,Kasus]) spaeter:Definitheit

% Quantoren qnt([ADeklination,Numerus],[Genus,Kasus]) spaeter: Definitheit

% Possessivpronomen: poss([Pers,Gen,Num],[Gen,Num,Kas]) (attributiv)

poss([3,mask,sg],[fem,sg,nom]) --> [seine].
poss([3,mask,sg],[fem,sg,gen]) --> [seiner].
poss([3,mask,sg],[fem,sg,dat]) --> [seiner].
poss([3,mask,sg],[fem,sg,akk]) --> [seine].

poss([3,mask,sg],[mask,sg,nom]) --> [sein].
poss([3,mask,sg],[mask,sg,gen]) --> [seines].
poss([3,mask,sg],[mask,sg,dat]) --> [seinem].
poss([3,mask,sg],[mask,sg,akk]) --> [seinen].

poss([3,mask,sg],[neut,sg,nom]) --> [sein].
poss([3,mask,sg],[neut,sg,gen]) --> [seines].
poss([3,mask,sg],[neut,sg,dat]) --> [seinem].
poss([3,mask,sg],[neut,sg,akk]) --> [sein].

```

```

poss([3,-,sg],[-,pl,nom]) --> [seine].
poss([3,-,sg],[-,pl,gen]) --> [seiner].
poss([3,-,sg],[-,pl,dat]) --> [seinen].
poss([3,-,sg],[-,pl,akk]) --> [seine].

% possessiv verwendetes Demonstrativpronomen:
% = vorangestelltes DemPron.gen (???)
% Demonstrativpronomen:
% mask der, dessen, dem, den
% fem die, deren/derer, der, die
% neut das, dessen, dem, den
% pl die, deren/derer, denen, die

% ad-hoc:
% mask|neut: dessen (Hund|Katze|Tier)(e)
% fem deren (Hund|Katze|Tier)(e)
% pl deren (Hund|Katze|Tier)(e)

poss([3,mask,sg],[fem,sg,nom]) --> [dessen].
poss([3,mask,sg],[fem,sg,gen]) --> [dessen].
poss([3,mask,sg],[fem,sg,dat]) --> [dessen].
poss([3,mask,sg],[fem,sg,akk]) --> [dessen].

poss([3,mask,sg],[mask,sg,nom]) --> [dessen].
poss([3,mask,sg],[mask,sg,gen]) --> [dessen].
poss([3,mask,sg],[mask,sg,dat]) --> [dessen].
poss([3,mask,sg],[mask,sg,akk]) --> [dessen].

poss([3,mask,sg],[neut,sg,nom]) --> [dessen].
poss([3,mask,sg],[neut,sg,gen]) --> [dessen].
poss([3,mask,sg],[neut,sg,dat]) --> [dessen].
poss([3,mask,sg],[neut,sg,akk]) --> [dessen].

poss([3,fem,sg],[fem,sg,nom]) --> [deren].
poss([3,fem,sg],[fem,sg,gen]) --> [deren].
poss([3,fem,sg],[fem,sg,dat]) --> [deren].
poss([3,fem,sg],[fem,sg,akk]) --> [deren].

poss([3,fem,sg],[mask,sg,nom]) --> [deren].
poss([3,fem,sg],[mask,sg,gen]) --> [deren].
poss([3,fem,sg],[mask,sg,dat]) --> [deren].
poss([3,fem,sg],[mask,sg,akk]) --> [deren].

poss([3,fem,sg],[neut,sg,nom]) --> [deren].
poss([3,fem,sg],[neut,sg,gen]) --> [deren].
poss([3,fem,sg],[neut,sg,dat]) --> [deren].
poss([3,fem,sg],[neut,sg,akk]) --> [deren].

```

```

poss([3,-,pl],[-,pl,nom]) --> [deren].
poss([3,-,pl],[-,pl,gen]) --> [derer]. % deren?
poss([3,-,pl],[-,pl,dat]) --> [deren].
poss([3,-,pl],[-,pl,akk]) --> [deren].

```

```

% Adjektive a([Kompl],[Dekl,Gen,Num,Kas])

```

```

a([[]],[st,mask,sg,nom]) --> [derartiger].
a([[]],[st,mask,sg,gen]) --> [derartigen].
a([[]],[st,mask,sg,dat]) --> [derartigem].
a([[]],[st,mask,sg,akk]) --> [derartigen].

```

```

a([[]],[st,fem,sg,nom]) --> [derartige].
a([[]],[st,fem,sg,gen]) --> [derartiger].
a([[]],[st,fem,sg,dat]) --> [derartiger].
a([[]],[st,fem,sg,akk]) --> [derartige].

```

```

a([[]],[st,neut,sg,nom]) --> [derartiges].
a([[]],[st,neut,sg,gen]) --> [derartigen].
a([[]],[st,neut,sg,dat]) --> [derartigem].
a([[]],[st,neut,sg,akk]) --> [derartiges].

```

```

a([[]],[st,-,pl,nom]) --> [derartige].
a([[]],[st,-,pl,gen]) --> [derartiger].
a([[]],[st,-,pl,dat]) --> [derartigen].
a([[]],[st,-,pl,akk]) --> [derartige].

```

```

a([[]],[sw,mask,sg,nom]) --> [derartige].
a([[]],[sw,mask,sg,gen]) --> [derartigen].
a([[]],[sw,mask,sg,dat]) --> [derartigen].
a([[]],[sw,mask,sg,akk]) --> [derartigen].

```

```

a([[]],[sw,fem,sg,nom]) --> [derartige].
a([[]],[sw,fem,sg,gen]) --> [derartigen].
a([[]],[sw,fem,sg,dat]) --> [derartigen].
a([[]],[sw,fem,sg,akk]) --> [derartige].

```

```

a([[]],[sw,neut,sg,nom]) --> [derartige].
a([[]],[sw,neut,sg,gen]) --> [derartigen].
a([[]],[sw,neut,sg,dat]) --> [derartigen].
a([[]],[sw,neut,sg,akk]) --> [derartige].

```

```

a([[]],[sw,-,pl,nom]) --> [derartigen].
a([[]],[sw,-,pl,gen]) --> [derartigen].
a([[]],[sw,-,pl,dat]) --> [derartigen].
a([[]],[sw,-,pl,akk]) --> [derartigen].

```

a([], [ge,mask,sg,nom]) --> [derartiger].
a([], [ge,mask,sg,gen]) --> [derartigen].
a([], [ge,mask,sg,dat]) --> [derartigen].
a([], [ge,mask,sg,akk]) --> [derartigen].

a([], [ge,fem,sg,nom]) --> [derartige].
a([], [ge,fem,sg,gen]) --> [derartigen].
a([], [ge,fem,sg,dat]) --> [derartigen].
a([], [ge,fem,sg,akk]) --> [derartige].

a([], [ge,neut,sg,nom]) --> [derartiges].
a([], [ge,neut,sg,gen]) --> [derartigen].
a([], [ge,neut,sg,dat]) --> [derartigen].
a([], [ge,neut,sg,akk]) --> [derartiges].

a([], [ge,-,pl,nom]) --> [derartigen].
a([], [ge,-,pl,gen]) --> [derartigen].
a([], [ge,-,pl,dat]) --> [derartigen].
a([], [ge,-,pl,akk]) --> [derartigen].

a([], [st,mask,sg,nom]) --> [geltender].
a([], [st,mask,sg,gen]) --> [geltenden].
a([], [st,mask,sg,dat]) --> [geltendem].
a([], [st,mask,sg,akk]) --> [geltenden].

a([], [st,fem,sg,nom]) --> [geltende].
a([], [st,fem,sg,gen]) --> [geltender].
a([], [st,fem,sg,dat]) --> [geltender].
a([], [st,fem,sg,akk]) --> [geltende].

a([], [st,neut,sg,nom]) --> [geltendes].
a([], [st,neut,sg,gen]) --> [geltenden].
a([], [st,neut,sg,dat]) --> [geltendem].
a([], [st,neut,sg,akk]) --> [geltendes].

a([], [st,-,pl,nom]) --> [geltende].
a([], [st,-,pl,gen]) --> [geltender].
a([], [st,-,pl,dat]) --> [geltenden].
a([], [st,-,pl,akk]) --> [geltende].

a([], [sw,mask,sg,nom]) --> [geltende].
a([], [sw,mask,sg,gen]) --> [geltenden].
a([], [sw,mask,sg,dat]) --> [geltenden].
a([], [sw,mask,sg,akk]) --> [geltenden].

a([], [sw,fem,sg,nom]) --> [geltende].
a([], [sw,fem,sg,gen]) --> [geltenden].

a([], [sw, fem, sg, dat]) --> [geltenden].
a([], [sw, fem, sg, akk]) --> [geltende].

a([], [sw, neut, sg, nom]) --> [geltende].
a([], [sw, neut, sg, gen]) --> [geltenden].
a([], [sw, neut, sg, dat]) --> [geltenden].
a([], [sw, neut, sg, akk]) --> [geltende].

a([], [sw, -, pl, nom]) --> [geltenden].
a([], [sw, -, pl, gen]) --> [geltenden].
a([], [sw, -, pl, dat]) --> [geltenden].
a([], [sw, -, pl, akk]) --> [geltenden].

a([], [ge, mask, sg, nom]) --> [geltender].
a([], [ge, mask, sg, gen]) --> [geltenden].
a([], [ge, mask, sg, dat]) --> [geltenden].
a([], [ge, mask, sg, akk]) --> [geltenden].

a([], [ge, fem, sg, nom]) --> [geltende].
a([], [ge, fem, sg, gen]) --> [geltenden].
a([], [ge, fem, sg, dat]) --> [geltenden].
a([], [ge, fem, sg, akk]) --> [geltende].

a([], [ge, neut, sg, nom]) --> [geltendes].
a([], [ge, neut, sg, gen]) --> [geltenden].
a([], [ge, neut, sg, dat]) --> [geltenden].
a([], [ge, neut, sg, akk]) --> [geltendes].

a([], [ge, -, pl, nom]) --> [geltenden].
a([], [ge, -, pl, gen]) --> [geltenden].
a([], [ge, -, pl, dat]) --> [geltenden].
a([], [ge, -, pl, akk]) --> [geltenden].

a([], [st, mask, sg, nom]) --> [technischer].
a([], [st, mask, sg, gen]) --> [technischen].
a([], [st, mask, sg, dat]) --> [technischem].
a([], [st, mask, sg, akk]) --> [technischen].

a([], [st, fem, sg, nom]) --> [technische].
a([], [st, fem, sg, gen]) --> [technischer].
a([], [st, fem, sg, dat]) --> [technischer].
a([], [st, fem, sg, akk]) --> [technische].

a([], [st, neut, sg, nom]) --> [technisches].
a([], [st, neut, sg, gen]) --> [technischen].
a([], [st, neut, sg, dat]) --> [technischem].
a([], [st, neut, sg, akk]) --> [technisches].

a([[]],[st,-,pl,nom]) --> [technische].
a([[]],[st,-,pl,gen]) --> [technischer].
a([[]],[st,-,pl,dat]) --> [technischen].
a([[]],[st,-,pl,akk]) --> [technische].

a([[]],[sw,mask,sg,nom]) --> [technische].
a([[]],[sw,mask,sg,gen]) --> [technischen].
a([[]],[sw,mask,sg,dat]) --> [technischen].
a([[]],[sw,mask,sg,akk]) --> [technischen].

a([[]],[sw,fem,sg,nom]) --> [technische].
a([[]],[sw,fem,sg,gen]) --> [technischen].
a([[]],[sw,fem,sg,dat]) --> [technischen].
a([[]],[sw,fem,sg,akk]) --> [technische].

a([[]],[sw,neut,sg,nom]) --> [technische].
a([[]],[sw,neut,sg,gen]) --> [technischen].
a([[]],[sw,neut,sg,dat]) --> [technischen].
a([[]],[sw,neut,sg,akk]) --> [technische].

a([[]],[sw,-,pl,nom]) --> [technischen].
a([[]],[sw,-,pl,gen]) --> [technischen].
a([[]],[sw,-,pl,dat]) --> [technischen].
a([[]],[sw,-,pl,akk]) --> [technischen].

a([[]],[ge,mask,sg,nom]) --> [technischer].
a([[]],[ge,mask,sg,gen]) --> [technischen].
a([[]],[ge,mask,sg,dat]) --> [technischen].
a([[]],[ge,mask,sg,akk]) --> [technischen].

a([[]],[ge,fem,sg,nom]) --> [technische].
a([[]],[ge,fem,sg,gen]) --> [technischen].
a([[]],[ge,fem,sg,dat]) --> [technischen].
a([[]],[ge,fem,sg,akk]) --> [technische].

a([[]],[ge,neut,sg,nom]) --> [technisches].
a([[]],[ge,neut,sg,gen]) --> [technischen].
a([[]],[ge,neut,sg,dat]) --> [technischen].
a([[]],[ge,neut,sg,akk]) --> [technisches].

a([[]],[ge,-,pl,nom]) --> [technischen].
a([[]],[ge,-,pl,gen]) --> [technischen].
a([[]],[ge,-,pl,dat]) --> [technischen].
a([[]],[ge,-,pl,akk]) --> [technischen].

a([[]],[st,mask,sg,nom]) --> [gesetzlicher].

a([[]], [st,mask,sg,gen]) --> [gesetzlichen].
a([[]], [st,mask,sg,dat]) --> [gesetzlichem].
a([[]], [st,mask,sg,akk]) --> [gesetzlichen].

a([[]], [st,fem,sg,nom]) --> [gesetzliche].
a([[]], [st,fem,sg,gen]) --> [gesetzlicher].
a([[]], [st,fem,sg,dat]) --> [gesetzlicher].
a([[]], [st,fem,sg,akk]) --> [gesetzliche].

a([[]], [st,neut,sg,nom]) --> [gesetzliches].
a([[]], [st,neut,sg,gen]) --> [gesetzlichen].
a([[]], [st,neut,sg,dat]) --> [gesetzlichem].
a([[]], [st,neut,sg,akk]) --> [gesetzliches].

a([[]], [st,-,pl,nom]) --> [gesetzliche].
a([[]], [st,-,pl,gen]) --> [gesetzlicher].
a([[]], [st,-,pl,dat]) --> [gesetzlichen].
a([[]], [st,-,pl,akk]) --> [gesetzliche].

a([[]], [sw,mask,sg,nom]) --> [gesetzliche].
a([[]], [sw,mask,sg,gen]) --> [gesetzlichen].
a([[]], [sw,mask,sg,dat]) --> [gesetzlichen].
a([[]], [sw,mask,sg,akk]) --> [gesetzlichen].

a([[]], [sw,fem,sg,nom]) --> [gesetzliche].
a([[]], [sw,fem,sg,gen]) --> [gesetzlichen].
a([[]], [sw,fem,sg,dat]) --> [gesetzlichen].
a([[]], [sw,fem,sg,akk]) --> [gesetzliche].

a([[]], [sw,neut,sg,nom]) --> [gesetzliche].
a([[]], [sw,neut,sg,gen]) --> [gesetzlichen].
a([[]], [sw,neut,sg,dat]) --> [gesetzlichen].
a([[]], [sw,neut,sg,akk]) --> [gesetzliche].

a([[]], [sw,-,pl,nom]) --> [gesetzlichen].
a([[]], [sw,-,pl,gen]) --> [gesetzlichen].
a([[]], [sw,-,pl,dat]) --> [gesetzlichen].
a([[]], [sw,-,pl,akk]) --> [gesetzlichen].

a([[]], [ge,mask,sg,nom]) --> [gesetzlicher].
a([[]], [ge,mask,sg,gen]) --> [gesetzlichen].
a([[]], [ge,mask,sg,dat]) --> [gesetzlichen].
a([[]], [ge,mask,sg,akk]) --> [gesetzlichen].

a([[]], [ge,fem,sg,nom]) --> [gesetzliche].
a([[]], [ge,fem,sg,gen]) --> [gesetzlichen].
a([[]], [ge,fem,sg,dat]) --> [gesetzlichen].
a([[]], [ge,fem,sg,akk]) --> [gesetzliche].

a([[]], [ge, neut, sg, nom]) --> [gesetzliches].
a([[]], [ge, neut, sg, gen]) --> [gesetzlichen].
a([[]], [ge, neut, sg, dat]) --> [gesetzlichen].
a([[]], [ge, neut, sg, akk]) --> [gesetzliches].

a([[]], [ge, -, pl, nom]) --> [gesetzlichen].
a([[]], [ge, -, pl, gen]) --> [gesetzlichen].
a([[]], [ge, -, pl, dat]) --> [gesetzlichen].
a([[]], [ge, -, pl, akk]) --> [gesetzlichen].

a([[]], [st, mask, sg, nom]) --> [weiblicher].
a([[]], [st, mask, sg, gen]) --> [weiblichen].
a([[]], [st, mask, sg, dat]) --> [weiblichem].
a([[]], [st, mask, sg, akk]) --> [weiblichen].

a([[]], [st, fem, sg, nom]) --> [weibliche].
a([[]], [st, fem, sg, gen]) --> [weiblicher].
a([[]], [st, fem, sg, dat]) --> [weiblicher].
a([[]], [st, fem, sg, akk]) --> [weibliche].

a([[]], [st, neut, sg, nom]) --> [weibliches].
a([[]], [st, neut, sg, gen]) --> [weiblichen].
a([[]], [st, neut, sg, dat]) --> [weiblichem].
a([[]], [st, neut, sg, akk]) --> [weibliches].

a([[]], [st, -, pl, nom]) --> [weibliche].
a([[]], [st, -, pl, gen]) --> [weiblicher].
a([[]], [st, -, pl, dat]) --> [weiblichen].
a([[]], [st, -, pl, akk]) --> [weibliche].

a([[]], [sw, mask, sg, nom]) --> [weibliche].
a([[]], [sw, mask, sg, gen]) --> [weiblichen].
a([[]], [sw, mask, sg, dat]) --> [weiblichen].
a([[]], [sw, mask, sg, akk]) --> [weiblichen].

a([[]], [sw, fem, sg, nom]) --> [weibliche].
a([[]], [sw, fem, sg, gen]) --> [weiblichen].
a([[]], [sw, fem, sg, dat]) --> [weiblichen].
a([[]], [sw, fem, sg, akk]) --> [weibliche].

a([[]], [sw, neut, sg, nom]) --> [weibliche].
a([[]], [sw, neut, sg, gen]) --> [weiblichen].
a([[]], [sw, neut, sg, dat]) --> [weiblichen].
a([[]], [sw, neut, sg, akk]) --> [weibliche].

a([[]], [sw, -, pl, nom]) --> [weiblichen].

```

a([[]],[sw,-,pl,gen]) --> [weiblichen].
a([[]],[sw,-,pl,dat]) --> [weiblichen].
a([[]],[sw,-,pl,akk]) --> [weiblichen].

a([[]],[ge,mask,sg,nom]) --> [weiblicher].
a([[]],[ge,mask,sg,gen]) --> [weiblichen].
a([[]],[ge,mask,sg,dat]) --> [weiblichen].
a([[]],[ge,mask,sg,akk]) --> [weiblichen].

a([[]],[ge,fem,sg,nom]) --> [weibliche].
a([[]],[ge,fem,sg,gen]) --> [weiblichen].
a([[]],[ge,fem,sg,dat]) --> [weiblichen].
a([[]],[ge,fem,sg,akk]) --> [weibliche].

a([[]],[ge,neut,sg,nom]) --> [weibliches].
a([[]],[ge,neut,sg,gen]) --> [weiblichen].
a([[]],[ge,neut,sg,dat]) --> [weiblichen].
a([[]],[ge,neut,sg,akk]) --> [weibliches].

a([[]],[ge,-,pl,nom]) --> [weiblichen].
a([[]],[ge,-,pl,gen]) --> [weiblichen].
a([[]],[ge,-,pl,dat]) --> [weiblichen].
a([[]],[ge,-,pl,akk]) --> [weiblichen].

```

Bei Nomen, die ein präpositionales Objekt erlauben, sollten die Lexikoneinträge im Komplementrahmen in $n([\text{Kompl}, \text{Gen}], [\text{Num}, \text{Kas}])$ nicht nur die Präposition angeben, sondern auch den von ihr verlangten Kasus der Objekt-Nominalphrase; daher sollte der Lexikoneintrag

```
n([[an,dat],fem],[sg,nom]) --> ['Freude'].
```

lauten. Dann können wir bei nominalisierten Verben „an mit Akkusativ“ verlangen, etwa in

```
n([[an,akk],fem],[sg,nom]) --> ['Glaube'].
```

und gleichzeitig „an mit Dativ“ beim Parsen ablehnen.

Da bei der Syntaxanalyse eines erweiterten Nomens ohne Adjektiv nie für das Merkmal `Dekl` ein Wert gefunden wird, hatte ich in der Regel ein

```

n1([Gen],[Dekl,Num,Kas]) -->
  { Dekl=(-) },
  ...

```

eingebaut; das gibt weniger Analysen, als wenn man hier jeden der drei Merkmalwerte `st`, `sw`, `ge` für `Dekl` erlaubt. Bei dem Zusammenbauen der $n1([\text{Gen}], [-, \text{Num}, \text{Kas}])$ mit einem Artikel muß man dann darauf achten, daß die Deklination nicht beim regierenden Artikel abgelesen wird.

Bem. Wenn man die Nominalphrase mit vorangestelltem Genitiv statt eines Artikels erlauben will, muß man dafür sorgen, daß die Nominalphrase nicht selber im Genitiv steht, da sonst eine `np.gen` links in einer Regel für `np.gen` auftritt und dann der Prolog-Parser divergiert.

```
% NP mit vorangestellter Genitiv-NP statt Determinator
np([3,Gen,Num],[Kas]) -->
  { member(Kas,[nom,dat,akk]) }, % gen ausschliessen wg. Linksrekursion
  np([Per1,Gen1,Num1],[gen]),
  { Dekl = ge },
  n1([Gen],[Dekl,Num,Kas]).
```

Hier ist die Ausgabe, allerdings nachdem Regeln aus `np.folien.pl` geändert und vielleicht nicht genau auf den Stand der Folien zurückgebracht wurden: (einige scheitern, weil noch Lexikoneinträge fehlen)

```
?- ['np.folien.pl','lexikon-beispielsaetze.pl'].
% tokenizer.mini.pl compiled into tokenizer 0.00 sec, 9,984 bytes
% addDifflists compiled into addDifflists 0.00 sec, 1,444 bytes
% addTree compiled into addTree 0.00 sec, 2,328 bytes
% term_expansion.pl compiled 0.01 sec, 4,612 bytes
% showTree.pl compiled 0.00 sec, 2,772 bytes
% dot.syntaxbaum.pl compiled into display_tree 0.00 sec, 6,940 bytes
% DCGparser/parser.pl compiled 0.01 sec, 25,528 bytes
% lexikon_np compiled 0.06 sec, 177,272 bytes
% np.folien.pl compiled 0.07 sec, 218,960 bytes
% lexikon-beispielsaetze.pl compiled 0.04 sec, 101,896 bytes
true.
```

```
?- parse('beispielnominalphrasen.txt').
```

```
[derartige, 'Fälle']
```

```
np([3, mask, pl], [akk])
  n1([mask], [st, pl, akk])
  ap([], [pos, st, -, pl, akk])
  a([], [st, -, pl, akk]) derartige
  n1([mask], [-, pl, akk])
  n([], mask, [pl, akk])
  cn([mask], [pl, akk]) 'Fälle'
np([3, mask, pl], [nom])
  n1([mask], [st, pl, nom])
  ap([], [pos, st, -, pl, nom])
  a([], [st, -, pl, nom]) derartige
  n1([mask], [-, pl, nom])
  n([], mask, [pl, nom])
  cn([mask], [pl, nom]) 'Fälle'
```

```
[den, 'Rahmen']
```

```
np([3, mask, pl], [dat])
  det([3, pl, sw], [-, dat])
  art([sw, pl], [-, dat]) den
```

```

n1([mask], [-, pl, dat])
  n([], mask, [pl, dat])
    cn([mask], [pl, dat]) 'Rahmen'
np([3, mask, sg], [akk])
  det([3, sg, sw], [mask, akk])
  art([sw, sg], [mask, akk]) den
n1([mask], [-, sg, akk])
  n([], mask, [sg, akk])
    cn([mask], [sg, akk]) 'Rahmen'

```

[der, geltenden, 'Finanzverfassung']

```

np([3, fem, sg], [dat])
  det([3, sg, sw], [fem, dat])
  art([sw, sg], [fem, dat]) der
n1([fem], [sw, sg, dat])
  ap([], [pos, sw, fem, sg, dat])
  a([], [sw, fem, sg, dat]) geltenden
n1([fem], [-, sg, dat])
  n([], fem, [sg, dat])
    cn([fem], [sg, dat]) 'Finanzverfassung'
np([3, fem, sg], [gen])
  det([3, sg, sw], [fem, gen])
  art([sw, sg], [fem, gen]) der
n1([fem], [sw, sg, gen])
  ap([], [pos, sw, fem, sg, gen])
  a([], [sw, fem, sg, gen]) geltenden
n1([fem], [-, sg, gen])
  n([], fem, [sg, gen])
    cn([fem], [sg, gen]) 'Finanzverfassung'

```

[den, 'Rahmen', der, geltenden, 'Finanzverfassung']

```

np([3, mask, pl], [dat])
  det([3, pl, sw], [-, dat])
  art([sw, pl], [-, dat]) den
n1([mask], [-, pl, dat])
  n([], mask, [pl, dat])
    cn([mask], [pl, dat]) 'Rahmen'
np([3, fem, sg], [gen])
  det([3, sg, sw], [fem, gen])
  art([sw, sg], [fem, gen]) der
n1([fem], [sw, sg, gen])
  ap([], [pos, sw, fem, sg, gen])
  a([], [sw, fem, sg, gen]) geltenden
n1([fem], [-, sg, gen])
  n([], fem, [sg, gen])
    cn([fem], [sg, gen]) 'Finanzverfassung'

```

```

np([3, mask, sg], [akk])
  det([3, sg, sw], [mask, akk])
    art([sw, sg], [mask, akk]) den
  n1([mask], [-, sg, akk])
    n([], mask, [sg, akk])
      cn([mask], [sg, akk]) 'Rahmen'
  np([3, fem, sg], [gen])
    det([3, sg, sw], [fem, gen])
      art([sw, sg], [fem, gen]) der
    n1([fem], [sw, sg, gen])
      ap([], [pos, sw, fem, sg, gen])
        a([], [sw, fem, sg, gen]) geltenden
      n1([fem], [-, sg, gen])
        n([], fem, [sg, gen])
          cn([fem], [sg, gen]) 'Finanzverfassung'

```

[die, 'Sicherung']

```

np([3, fem, sg], [akk])
  det([3, sg, sw], [fem, akk])
    art([sw, sg], [fem, akk]) die
  n1([fem], [-, sg, akk])
    n([], fem, [sg, akk])
      cn([fem], [sg, akk]) 'Sicherung'
  np([3, fem, sg], [nom])
    det([3, sg, sw], [fem, nom])
      art([sw, sg], [fem, nom]) die
    n1([fem], [-, sg, nom])
      n([], fem, [sg, nom])
        cn([fem], [sg, nom]) 'Sicherung'

```

[deren, 'Sicherung']

```

np([3, fem, sg], [akk])
  det([3, sg, ge], [fem, akk])
    poss([3, fem, sg], [fem, sg, akk]) deren
  n1([fem], [-, sg, akk])
    n([], fem, [sg, akk])
      cn([fem], [sg, akk]) 'Sicherung'
  np([3, fem, sg], [dat])
    det([3, sg, ge], [fem, dat])
      poss([3, fem, sg], [fem, sg, dat]) deren
    n1([fem], [-, sg, dat])
      n([], fem, [sg, dat])
        cn([fem], [sg, dat]) 'Sicherung'
  np([3, fem, sg], [gen])
    det([3, sg, ge], [fem, gen])
      poss([3, fem, sg], [fem, sg, gen]) deren

```

```

n1([fem], [-, sg, gen])
  n([], fem), [sg, gen])
    cn([fem], [sg, gen]) 'Sicherung'
np([3, fem, sg], [nom])
  det([3, sg, ge], [fem, nom])
    poss([3, fem, sg], [fem, sg, nom]) deren
  n1([fem], [-, sg, nom])
    n([], fem), [sg, nom])
      cn([fem], [sg, nom]) 'Sicherung'

```

[deren, technische, 'Sicherung']

```

np([3, fem, sg], [akk])
  det([3, sg, ge], [fem, akk])
    poss([3, fem, sg], [fem, sg, akk]) deren
  n1([fem], [ge, sg, akk])
    ap([], [pos, ge, fem, sg, akk])
      a([], [ge, fem, sg, akk]) technische
    n1([fem], [-, sg, akk])
      n([], fem), [sg, akk])
        cn([fem], [sg, akk]) 'Sicherung'
np([3, fem, sg], [nom])
  det([3, sg, ge], [fem, nom])
    poss([3, fem, sg], [fem, sg, nom]) deren
  n1([fem], [ge, sg, nom])
    ap([], [pos, ge, fem, sg, nom])
      a([], [ge, fem, sg, nom]) technische
    n1([fem], [-, sg, nom])
      n([], fem), [sg, nom])
        cn([fem], [sg, nom]) 'Sicherung'

```

[gesetzlichen, 'Vorgaben']

```

np([3, fem, pl], [dat])
  n1([fem], [st, pl, dat])
    ap([], [pos, st, -, pl, dat])
      a([], [st, -, pl, dat]) gesetzlichen
    n1([fem], [-, pl, dat])
      n([], fem), [pl, dat])
        cn([fem], [pl, dat]) 'Vorgaben'

```

[des, 'Autors']

```

np([3, mask, sg], [gen])
  det([3, sg, sw], [mask, gen])
    art([sw, sg], [mask, gen]) des
  n1([mask], [-, sg, gen])
    n([], mask), [sg, gen])

```


cn([mask], [sg, gen]) 'Autors'

[des, 'Autors', 'Vorliebe']

* keine Analysen gefunden *

[den, weiblichen, 'Figuren']

np([3, fem, pl], [dat])
det([3, pl, sw], [-, dat])
art([sw, pl], [-, dat]) den
n1([fem], [sw, pl, dat])
ap([], [pos, sw, -, pl, dat])
a([], [sw, -, pl, dat]) weiblichen
n1([fem], [-, pl, dat])
n([], fem], [pl, dat])
cn([fem], [pl, dat]) 'Figuren'

[der, 'Mann']

np([3, mask, sg], [nom])
det([3, sg, sw], [mask, nom])
art([sw, sg], [mask, nom]) der
n1([mask], [-, sg, nom])
n([], mask], [sg, nom])
cn([mask], [sg, nom]) 'Mann'

[seinem, 'Job']

np([3, mask, sg], [dat])
det([3, sg, ge], [mask, dat])
poss([3, mask, sg], [mask, sg, dat]) seinem
n1([mask], [-, sg, dat])
n([], mask], [sg, dat])
cn([mask], [sg, dat]) 'Job'

['Freude']

np([3, fem, sg], [akk])
n1([fem], [-, sg, akk])
n([[an, dat], fem], [sg, akk]) 'Freude'
np([3, fem, sg], [dat])
n1([fem], [-, sg, dat])
n([[an, dat], fem], [sg, dat]) 'Freude'
np([3, fem, sg], [nom])
n1([fem], [-, sg, nom])
n([[an, dat], fem], [sg, nom]) 'Freude'

['Freude', an, seinem, 'Job']

```
np([3, fem, sg], [akk])
  n1([fem], [-, sg, akk])
    n([[an, dat], fem], [sg, akk]) 'Freude'
    an
    np([3, mask, sg], [dat])
      det([3, sg, ge], [mask, dat])
        poss([3, mask, sg], [mask, sg, dat]) seinem
      n1([mask], [-, sg, dat])
        n([], mask, [sg, dat])
          cn([mask], [sg, dat]) 'Job'
np([3, fem, sg], [dat])
  n1([fem], [-, sg, dat])
    n([[an, dat], fem], [sg, dat]) 'Freude'
    an
    np([3, mask, sg], [dat])
      det([3, sg, ge], [mask, dat])
        poss([3, mask, sg], [mask, sg, dat]) seinem
      n1([mask], [-, sg, dat])
        n([], mask, [sg, dat])
          cn([mask], [sg, dat]) 'Job'
np([3, fem, sg], [nom])
  n1([fem], [-, sg, nom])
    n([[an, dat], fem], [sg, nom]) 'Freude'
    an
    np([3, mask, sg], [dat])
      det([3, sg, ge], [mask, dat])
        poss([3, mask, sg], [mask, sg, dat]) seinem
      n1([mask], [-, sg, dat])
        n([], mask, [sg, dat])
          cn([mask], [sg, dat]) 'Job'
```

[des, 'Autors', 'Freude', an, seinem, 'Job']

* keine Analysen gefunden *

[die, 'Freude', an, seinem, 'Job']

```
np([3, fem, sg], [akk])
  det([3, sg, sw], [fem, akk])
    art([sw, sg], [fem, akk]) die
  n1([fem], [-, sg, akk])
    n([[an, dat], fem], [sg, akk]) 'Freude'
    an
    np([3, mask, sg], [dat])
      det([3, sg, ge], [mask, dat])
        poss([3, mask, sg], [mask, sg, dat]) seinem
```

```

        n1([mask], [-, sg, dat])
          n([], mask, [sg, dat])
            cn([mask], [sg, dat]) 'Job'
np([3, fem, sg], [nom])
  det([3, sg, sw], [fem, nom])
    art([sw, sg], [fem, nom]) die
  n1([fem], [-, sg, nom])
    n([[an, dat], fem], [sg, nom]) 'Freude'
  an
  np([3, mask, sg], [dat])
    det([3, sg, ge], [mask, dat])
      poss([3, mask, sg], [mask, sg, dat]) seinem
    n1([mask], [-, sg, dat])
      n([], mask, [sg, dat])
        cn([mask], [sg, dat]) 'Job'

```

[der, 'Glaube', an, seinem, 'Job']

* keine Analysen gefunden *

[die, 'Freude', an, seinen, 'Job']

* keine Analysen gefunden *

[der, 'Glaube', an, seinen, 'Job']

```

np([3, mask, sg], [nom])
  det([3, sg, sw], [mask, nom])
    art([sw, sg], [mask, nom]) der
  n1([mask], [-, sg, nom])
    n([[an, akk], mask], [sg, nom]) 'Glaube'
  an
  np([3, mask, sg], [akk])
    det([3, sg, ge], [mask, akk])
      poss([3, mask, sg], [mask, sg, akk]) seinen
    n1([mask], [-, sg, akk])
      n([], mask, [sg, akk])
        cn([mask], [sg, akk]) 'Job'

```

Fertig.

true.

?-

Beachte, daß manche Eingaben nur erweiterte Nomen n1 sind und nicht *als* np erkannt werden, weil die Regel np --> n1 eine Beschränkung auf Numerus p1 hat.