

Syntax natürlicher Sprachen

WS 2016/17
Erläuterungen zur implementierten Grammatik

Hans Leiß
leiss@cis.uni-muenchen.de
Universität München
Centrum für Informations- und Sprachverarbeitung

8. März 2017

1 Kategorien

Alle Wort- und Ausdruckskategorien werden durch *Artmerkmale* in Unterarten eingeteilt und durch *Flexions-* oder *Formmerkmale* nach ihren, i.a. von der Unterart abhängigen, Wort- oder Ausdrucksformen unterschieden.

Zu einer Kategorie *Kat* werden im unteren Index die Artmerkmale, im oberen index die Formmerkmale notiert: Kat_{Art}^{Form} .

In der Implementation als Hornformelgrammatik (DCG) in Prolog hat die Kategorie zwei Parameterlisten, eine Liste $[a_1, \dots, a_n]$ von Artmerkmalen und eine Liste $[f_1, \dots, f_k]$ von Formmerkmalen, $cat([a_1, \dots, a_n], [f_1, \dots, f_k])$. Die Merkmale (bzw. Merkmalwerte) sind in bestimmter Reihenfolge angegeben, wodurch wir uns eine Angabe der Merkmalsdimensionen sparen, z.B. $n([fem], [sg, dat])$ statt $n([genus = fem], [numerus = sg, kasus = dat])$.

Abkürzungen der Merkmalbereiche und -Werte

Neben den hier genannten Merkmalwerten wird der Wert - verwendet, der für „unbestimmter Wert“ des Merkmals steht. Das kann z.B. das Genus beim Nomen und der Nominalphrase *die Leute* sein, aber auch Merkmale wie Numerus und Tempus bei koordinierten Ausdrücken, deren Teilausdrücke bei diesem Merkmal verschiedene Werte aufweisen, z.B. *ein Erwachsener und mehrere Kinder*, oder *sie kamen gestern und werden bis morgen hier bleiben*.

Kasus = $\{nom, gen, dat, akk\}$, mit Variable *Kas*.

Numerus = $\{sg, pl\}$, mit Variable *Num*.

Genus = $\{mask, fem, neut\}$, mit Variable *Gen*.

Adjektivdeklination = $\{st, sw, ge\}$ mit den Werten *stark*, *schwach*, *gemischt* und der Variable *ADekl* oder kurz *Dekl*.

Person = $\{1, 2, 3\}$ für *erste*, *zweite*, *dritte* Person, und Variable *Pers*.

Komplementrahmen , mit Variable $Kompl$, hat Listen aus Kasus, Präpositionen und Satzarten und -formen als Werte, z.B. $[an, akk]$ für Präpositionalobjekte mit der Präposition $an : P_{[akk]}$, die ein Nominaphrasenobjekt im Akkusativ erfordern. (Die Grammatikregeln lesen die Kasus von Objekten am Komplementrahmen ab und streuen die Präposition als Terminalsymbol zwischen die Konstituentenkategorien.)

Genauere Werte: siehe Verben, Nomina und Adjektive.

Verbtempus = $\{praes, praet\}$ für Präsens und Präteritum, mit Variable $VTemp$ oder auch $Temp$.

Modus = $\{ind, konj\}$ für Indikativ und Konjunktiv, mit Variable Mod .

Tempus = $\{praes, praet, perf, plqm, fut1, fut2\}$ für Präsens, Präteritum, Perfekt, Plusquamperfekt, Futur-1 und Futur-2, mit Variable $Temp$.

Verbstellung = $\{ve, vz, vl\}$ für *Verberst-*, *Verbweit-* und *Verbletzstellung*, mit Variable Vst , einem Formmerkmal bei einfachen Sätzen.

Definitheit = $\{def, qu, rel\}$ für *definite*/bestimmte, *interrogative*, und *relativ(ierend)e* Unterarten von Ausdrücken, z.B. Pronomen, Nominalphrasen oder Sätzen. Relativsätze unterteilen wir genauer mit Werten $rel(Gen, Num)$ nach dem Genus und Numerus des Bezugsnomens.

Adverbialdimension = $\{temp, mod, kaus, inst\}$ für *temporale*, *modale*, *kausale*, *instrumentale* (und ggf. *finale*, *durative* o.ä.) Arten von Umstandsbestimmungen, mit Variable Dim (o. Art).

1.1 Wortarten

Nomen $N_{Kompl, Gen}^{Num, Kas}$ werden nach ihrem Komplementrahmen und dem Genus in Arten eingeteilt und variieren in Numerus und Kasus. Für Nomen, die nur Singularformen oder nur Pluralformen haben, z.B. *das Geschehene*, *die Leute*, geben wir im Lexikon einfach nur diese Formen an. Bei Nomen, die nur Pluralformen haben, vergeben wir evtl. ad hoc einen echten Genuswert; besser wäre, den Wert (-) für undefiniertes Genus zu vergeben.

Wir verwenden die Kategorie N als Vereinigung von Unterkategorien, also eigentlich als eine Ausdrucksart; aber alle Nomen, die einen Komplementrahmen haben, werden als $N_{Kompl, Gen}^{Num, Kas}$ im Lexikon aufgeführt, auch wenn es eigentlich Gemein- oder Massennamen sind, bei deren Kategorien wir keinen Komplementrahmen vorgesehen haben, weil der meistens leer wäre.

Eigennamen PN_{Gen}^{Kas} variieren nur im Kasus.¹ Als Nomen erhalten sie den Numerus Singular und einen leeren Komplementrahmen, die Kategorie $N_{[], Gen}^{sg, Kas}$.

Gemeinnamen oder *zählbare Nomen*, $CN_{Gen}^{Num, Kas}$, variieren in Numerus und Kasus; bei den Ausnahmen, die nur in einem Numerus vorliegen, werden die anderen Formen einfach im Lexikon nicht aufgeführt.² Als Nomen erhalten sie einen leeren Komplementrahmen.

Massennamen $MN_{Gen}^{sg, Kas}$ variieren nur im Kasus und haben nur Singularformen; als Nomen erhalten sie einen leeren Komplementrahmen.

Nominalisierte Adjektive und Partizipien $NA_{Kompl, Gen}^{Num, Kas}$ variieren im Numerus (*der/die/das Gute*, *die Guten*) und Kasus; sie erben ihren Komplementrahmen von dem Adjektiv,

¹Eigennamen sind hier aufgeführt, da man sie durch Adjektive modifizieren kann und z.B. *die kleine Maria* : NP bekommt. Dann sollte aber die Regel $NP \rightarrow PN$ fehlen, weil sie –irrtümlich?– über $N' \rightarrow PN$ ableitbar ist.

²Der Nachteil ist, daß man nie weiß, ob Formen im Lexikon absichtlich fehlen oder ob nur das Lexikon noch unvollständig ist.

von dem sie abgeleitet wurden, z.B. *Neid auf* von *neidig auf* (wobei die Ableitung hier vielleicht eher umgekehrt ist).

Partizipien sind hier grob subsumiert; man muß aber genauer zwischen aktivischem Partizip (*das ein Bild malende Kind*) und passivischem Partizip (*das von einem Kind gemalte Bild*) unterscheiden und (wenn man Semantik betreiben will) den Komplementrahmen aus dem Komplementrahmen des zugrundeliegenden Verbs berechnen, was i.a. kein direktes Erben ist (*das Malen des Kindes*, *das Gemälde des Kindes*).

Nominalisierte Verben $NV_{Kompl, Gen}^{Num, Kas}$ variieren in Numerus und Kasus und erben ihren Komplementrahmen von demjenigen Verb, von dem sie abgeleitet sind, z.B. *Hoffnung*: $NA_{[auf, akk], fem}^{sg, Kas}$ von *hoffen*: $V_{[nom, auf, akk]}^{[inf]}$.

Im Komplementrahmen werden die Präposition und der Kasus von Präpositionalobjekten angegeben, z.B. *Angst (vor)*: $N_{[vor, dat], fem}^{sg, nom}$. In den Grammatikregeln wird eine Nebenbedingung angegeben, die zwischen Präpositionen und Kasus unterscheidet, und die Präposition wird als Terminalsymbol vor die Kategorie der Objekt nominalphrase gestellt.

Artikel betrachtet man normalerweise als in Genus, Numerus und Kasus variierend und teilt sie nach Definitheit (in bestimmte und unbestimmte) ein, also sollte $Art_{Def}^{Gen, Num, Kas}$ die Kategorie sein.

Stattdessen verwenden wir die Kategorie $Art_{ADekl, Num}^{Gen, Kas}$, betrachten die Artikel also als im Genus und Kasus variierend. Statt der Unterscheidung nach Definitheit geben wir unter den Artmerkmalen an, wie der Artikel die Adjektivdeklinationsart registriert: der bestimmte Artikel erfordert die schwache (*sw*), der unbestimmte die gemischte (*ge*) Adjektivdeklinationsart. Außerdem sehen wir den Numerus als inhärent und nicht variierbar an, weil der unbestimmte Artikel *ein*, *eine*, *ein* keinen Plural hat.³ Da wir beim bestimmten Artikel aber Singular- und Pluralformen haben, müssen wir diese dann auf zwei Arten $Art_{ADekl, sg}^{Gen, Kas}$ und $Art_{ADekl, pl}^{-, Kas}$ verteilen. Hierbei besagt der Genuswert -, daß kein echtes Genus ablesbar ist.⁴

Quantoren $Qnt_{ADekl, Num}^{Gen, Kas}$ werden nach der Adjektivdeklinationsart, die sie regieren, und dem Numerus eingeteilt, da nicht jeder Quantor in beiden Numeri auftritt: *jeder*: $Qnt_{sw, sg}^{mask, nom}$ hat keinen Plural, *vielen*: $Qnt_{sw, pl}^{-, nom}$ keinen Singular.

Bem. Ganz so einfach ist es auch wieder nicht: im Singular kann man *viel* mit Massennomen kombinieren, *viel Freude*, *viel Mut*, *viel Pech*, im Plural mit zählbaren Nomen: *vielen Frauen*, *vielen Männern*, *vielen Kindern*, aber nicht umgekehrt. Das ginge mit zwei Quantoren: *viel*: $Qnt_{st, sg}^{-, Kas}$, *vielen*: $Qnt_{ge, pl}^{-, Kas}$. (Hierüber müßte man genauer nachdenken.)

Adjektive $A_{Kompl}^{ADekl, Gen, Num, Kas}$ variieren in Adjektivdeklinationsart, Genus, Numerus und Kasus (und der nicht berücksichtigten Komparation) und werden nach ihren Komplementrahmen unterteilt. Es werden nur Präpositionalobjekte behandelt, z.B. *neugierig*: $A_{[auf, akk]}^{Form}$.

Adjektive mit Objekt nominalphrasen wie (*einer Sache*) *überdrüssig*: $A_{[gen]}^{ADekl, Gen, Num, Kas}$ werden nicht implementiert, da es zu Linksrekursionen (s.u.) führen würde.

Personalpronomen $Pron^{Pers, Gen, Num, Kas}$ variieren in Person, Genus, Numerus und Kasus. Hiermit sind nur definite Personalpronomen gemeint.

³M.M. ist *einige* nicht der Plural von *ein*, da *einige* einen eigenen Singular hat: *einiges Pech*.

⁴Da Genusunterschiede nur im Singular auftreten, könnte man es noch genauer mit zwei Kategorien $Art_{ADekl}^{Gen, Kas}$ für die Singular- und $Art_{ADekl, pl}^{Kas}$ für die Pluralartikel machen; es erleichtert aber das Formulieren der Grammatikregel, wenn man die Werte für *Gen*, *Num*, *Kas* an der Kategorie direkt ablesen und entsprechende Variable damit belegen kann.

Bem. Andere Pronomenarten wie Interrogativpronomen (*wer, was*) oder Indefinitpronomen (*man, jemand, etwas, niemand, nichts*), existieren nur in der dritten Person Singular und variieren nicht in allen genannten Dimensionen (z.B. Genus und Kasus bei *man?*), weshalb man bei ihnen andere Merkmale braucht. Für Relativsätze brauchen wir aber noch:

Relativpronomen $Pron_{rel}^{Pers, Gen, Num, Kas}$ variieren in Person, Genus, Numerus und Kasus. Unser Lexikon enthält nur die Form der dritten Person; in der ersten und zweiten Person sind die Formen im Nominativ aus zwei Wörtern zusammengesetzt: *der/die ich, der/die du, die wir, die ihr*, z.B. in *Wir, die wir dies unser Schicksal erkannten* (G.Seferis).

Verben $V_{Kompl}^{Pers, Num, VTemp, Mod}$ variieren in Person, Numerus, Tempus und Modus, wobei wir im Tempus bei Verben nur die Werte $VTemp = \{praes, praet\}$ vorsehen, während bei Sätzen weitere Tempuswerte wie Futur, Perfekt, Plusquamperfekt nötig sind. Als Artmerkmale berücksichtigen wir in *Kompl* gewisse Komplementrahmen wie $[nom, akk]$ für transitive Verben, $[nom]$ für intransitive Verben mit Nominalphrasensubjekt, $[nom, an, akk]$ für Verben mit Präpositionalobjekt *an jemanden/etwas*. Statt des Komplementrahmens sind auch die Werte *modal* für Modalverben, *sein, haben, werden, bleiben* für die entsprechenden Hilfsverben vorgesehen.

Adverbien Adv_{Dim} haben keine Formvarianten. Als Artmerkmal benutzen wir die (eher semantisch motivierte) Adverbialdimension. Für eine größere Grammatik bräuchte man auch die Definitheit, also $Adv_{Def, Dim}$, aber da wir nur definite Adverbien wie *hier* : $Adv_{def, lok}$ verwenden, aber keine interrogativen wie *wo* : $Adv_{qu, lok}$ in *Wo liegt das Buch?* und keine relativierenden wie *wo* : $Adv_{rel, lok}$ in *dort, wo der Pfeffer wächst*, lassen wir das Definitheitsmerkmal weg.

2 Nominalphrasengrammatik

Die Grammatik darf keine linksrekursive Regel enthalten, d.h. keine Regel der Form $Cat \rightarrow Cat \dots$, deren Ausdrücke mit einem Teilausdruck derselben Kategorie beginnt. Der Grund hierfür ist, daß solche Regeln bei der automatischen Übersetzung in Prologprädikate zu einer Definition wie $cat(X, Y) :- cat(X', Y'), \dots$ führen, die zu einer endlosen Beweissuche führen: es wird immer wieder $cat(X, Y)$ aufgerufen, ohne daß ein Wort der Eingabe gelesen bzw. „verbraucht“ wird. Allgemeiner darf mit der Grammatik auch keine Ableitung $A \Rightarrow^+ A\alpha$ in mehreren Schritten möglich sein, da das genauso zur Divergenz des (top-down-) Parsers führen würde.

Die Grammatikregeln erfordern neben den Wortarten eine Reihe von Ausdruckskategorien. Wir unterscheiden hier

Nominalphrasen $NP_{Pers, Gen, Num}^{Kas}$ variieren im Allgemeinen nur im Kasus. In Person und Numerus variieren zwar ein paar spezielle Nominalphrasen, z.B. die Personalpronomen, aber andere, z.B. die mit bestimmtem Artikel und Nomen aufgebauten, existieren nur in dritter Person.

Wir unterteilen die Nominalphrasen nach Person, Genus, und Numerus. Man könnte und sollte sie weiter nach *Definitheit* einteilen und zwischen *definiten, indefiniten, interrogativen* und *relativierenden* Nominalphrasen unterscheiden, also eine Kategorie $NP_{Def, Pers, Gen, Num}^{Kas}$ benutzen; da wir hier keine interrogativen und relativierenden Nominalphrasen betrachten, lassen wir das Definitheitsmerkmal weg.

erweiterte Nomen $N_{Gen}^{ADekl, Num, Kas}$ sollen Nomen mit ihren Objekten und Adjektiv- oder Genitivattributen sein; sie variieren in der Adjektivdeklinaton, Numerus und Kasus, haben aber ein (vom Nomen stammendes) festes Genus.

Determinatoren $Det_{Pers,Num,ADekl}^{Gen,Kas}$ teilen wir nach Person, Numerus und Adjektivdeklinatation ein, da man auch Personalpronomen der ersten und zweiten Person als Determinatoren verwenden kann; auch wollen wir die Quantoren, die i.a. nicht im Numerus variieren, zu den Determinatoren zählen⁵. Die Einteilung nach Adjektivdeklinatation dient dazu, die Abhängigkeit von Adjektivendungen vom Determinator festzulegen.

Adjektivphrasen $AP^{ADekl,Gen,Num,Kas}$ variieren in Adjektivdeklinatation, Genus, Numerus und Kasus (und der nicht berücksichtigten Komparation); man könnte sie in einfache und koordinierte einteilen.

2.1 Grammatikregeln

Die Grammatikregeln sind hier i.a. *schematische Regeln* oder *Regelschemata*: in einer Regel wie

$$NP_{Pers,Gen,Num}^{Kas} \rightarrow Pron^{Pers,Gen,Num,Kas}$$

treten Merkmalvariable statt bestimmter Merkmalwerte auf, z.B. *Kas* statt *nom, gen, dat, akk*. Das Regelschema steht also für alle die konkreten Regeln, die durch Belegen der Merkmalvariable mit den dafür erlaubten Merkmalwerten entstehen. Das obige Regelschema repräsentiert also grob $|Person| \cdot |Genus| \cdot |Numerus| \cdot |Kasus| = 3 \cdot 3 \cdot 2 \cdot 4 = 72$ Grammatikregeln; genau genommen, sind es nicht einmal halb so viele (Singular+Plural = $|\{1, 2, mask, fem, neut\}| \cdot 4 + |Person| \cdot 4 = 8 \cdot 4 = 32$), weil es in der ersten und zweiten Person und im Plural im Deutschen keine Genusunterschiede gibt.

Die Merkmalvariablen der Regelschemata werden durch die Benutzung lexikalischer Regeln wie $Pron^{1,-.pl,nom} \rightarrow wir$, in denen nur Merkmalwerte auftreten, durch Werte belegt. Wie im Beispiel das Genus in Kombination mit dem Plural können Merkmale durch den Wert - belegt sein, wenn die Unterscheidung nach den abstrakten Merkmalwerten nur zu unnötigen Lesartenvarianten führte.

Weiter ist zu beachten, daß in manchen Regeln die Kategorie des konstruierten Ausdrucks (der linken Regelseite) Merkmale tragen kann, die bei keiner der Konstituentenkategorien (der rechten Regelseite) vorkommen; in diesen Fällen muß man durch eine Nebenbedingung die Merkmalvariable belegen, wie die Person und den Numerus beim Regelschema

$$NP_{Pers,Gen,Num}^{Kas} \rightarrow PN_{Gen}^{Kas} \{Pers = 3, Num = sg\}.$$

Man könnte stattdessen auch die entsprechenden Merkmalwerte einsetzen, wie in

$$NP_{3,Gen,sg}^{Kas} \rightarrow PN_{Gen}^{Kas}.$$

Die Regelschemata sind hier oft so implementiert, daß auf der rechten Regelseite verschiedene Möglichkeiten erlaubt werden, wodurch quasi mehrere Regeln zusammengefaßt werden (wie $A \rightarrow \alpha_1$ bis $A \rightarrow \alpha_n$ in $A \rightarrow \alpha_1 | \dots | \alpha_n$)

Zum ersten Regelschema

$\langle \text{Grammatikregeln aus np.folien.pl} \rangle \equiv$

```
np( [Pers, Gen, Num], [Kas] ) -->
  ( pron( [], [Pers, Gen, Num, Kas] )           % er, sie, es
  ; pn( [Gen], [Kas] ), { Pers=3, Num=sg } % Hans, Maria
  ; n1( [Gen], [Dekl, Num, Kas] ),             % alte Menschen
    { Pers=3, (Num=pl; Num=sg),               % ohne Adj nicht
```

⁵Determinatoren sind hier also keine Wortart, sondern eine Art unterschiedlich gebildeter Ausdrücke, die hauptsächlich nach ihrer *Funktion* zusammengefaßt werden.

```

        (Dekl=st ; (Dekl=(-), Kas\=gen)) } % im pl.gen
; det([Pers,Num,Dekl],[GenD,Kas]), % der alte Hund
  { GenD=(-) ; GenD=Gen }, % ein alter Hund
n1([Gen],[Dekl1,Num,Kas]), % ich alter Hund
  { Dekl1=(-) ; Dekl1=Dekl } % ggf. kein Adj in N1
).

```

erläutern wir noch die beiden letzten Fälle:

a) wenn ein erweitertes Nomen ohne Artikel oder Quantor als Nominalphrase benutzt wird,

```

<Spezialfall np --> n1>≡
np([Pers,Gen,Num],[Kas]) -->
  n1([Gen],[Dekl,Num,Kas]), % alte Menschen
  { Pers=3, % (Land und Leute):Num=(-) ?
    (Dekl=st ; (Dekl=(-), Kas\=gen)) } % ohne Adj nicht im pl.gen

```

dritter Person entsteht. Am Wert der Adjektivdeklinaton sieht man⁶, ob im erweiterten Nomen ein Adjektivattribut auftritt (ADekl=st) oder nicht (ADekl=(-)); außerdem muß ein Adjektiv im erweiterten Nomen *stark* dekliniert sein, *kleines Kind*, und wenn kein Adjektiv auftritt, dann soll die Nominalphrase nicht im Genitiv verwendet sein: Genitivattribute wie in *die Angst kleiner Kinder* oder *eine Frage großer Angst* sollen erlaubt, aber solche ohne Adjektiv wie in *die Angst Kinder* oder *eine Frage Angst* ausgeschlossen werden.

b) Wenn ein Determinator mit einem erweiterten Nomen verbunden wird,

```

<Spezialfall np --> det, n1>≡
np([Pers,Gen,Num],[Kas]) -->
  det([Pers,Num,Dekl],[GenD,Kas]), % der alte Hund
  { GenD=(-) ; GenD=Gen }, % ein alter Hund
  n1([Gen],[Dekl1,Num,Kas]), % ich alter Hund
  { Dekl1=(-) ; Dekl1=Dekl } % ggf. kein Adj in N1

```

kann es sein, daß der Determinator keinen echten Genuswert hat, z.B. *alle* : $Qnt_{sw,pl}^{-,nom}$, und dann darf die Kongruenz mit dem Genus des Nomens, **GenD=Gen**, nicht verlangt werden. Analog darf man nur dann mit **Dekl1=Dekl** fordern, daß das regierende Merkmal **Dekl** ∈ {*st*, *sw*, *ge*} des Determinators das Formmerkmal **Dekl1** des erweiterten Nomens bestimmt, wenn im erweiterten Nomen ein Adjektivattribut (und damit nicht **Dekl=(-)**) auftritt.

```

<Regeln zur Erweiterung eines Nomens um Objekte und Attribute>≡
% N mit Komplement und Attribut: n1([Gen],[Adekl,Num,Kas])

n1([Gen],[Dekl,Num,Kas]) --> % Nomen+Komplement
  { Dekl = (-) }, % keine Adjektive vorhanden
n([Kompl,Gen],[Num,Kas]), % ggf. Kompl = []
( [] ; np([3,-,-],[gen]) ), % opt. gen.poss/subj
( [] % Komplement optional: Hoffnung (auf NP)
; { Kompl=[gen] }, [] % gen-Objekt statt gen.subj
; { Kompl=[Praep,KasP], % Angst vor den Hunden
  praeposition(Praep,KasP) },

```

⁶Man vergleiche das mit dem Merkmal *Figur* bei Dionysos Thrax, mit dem zwischen einfachen und zusammengesetzten Ausdrücken unterschieden wurde. Analog können wir am Wert **Num=(-)** oder **Pers=(-)** feststellen, daß eine Nominalphrase nicht als Subjekt brauchbar ist, z.B. *manche Erwachsene und jedes Kind* oder *du und er*.

```

    [Praep], np([_Pers,_Gen,_Num],[KasP])
; { Kompl=[dass] },
  ( s([dass],[v1,Mod]) % These, daß sie rund ist|sei
  ; s([def],[vz,konj]) % These, sie *ist|sei rund
  )
; { Kompl=[ob] },      % Untersuchung, ob sie lebt
  s([ob],[v1,ind])
; { Kompl=[qu] },      % Frage, wo Maria lebt
  s([qu],[v1,Mod])
; { Kompl=[infzu] },   % Absicht, zu rauchen
  s([infzu],[ ])
; s([rel],[Mod])      % Menschen, an die wir denken.
).

```

Hier wird nach dem unter den Artmerkmalen des Nomens angegebenen „Komplementrahmen“ *Kompl* unterschieden und jeweils gesagt, welche Ausdrücke das Nomen erweitern, genauer gesagt, auf das Nomen folgen. Grob gesagt, hat die Regel die Form $N' \rightarrow N_{Kompl} K_1 \dots K_n$, wobei die Ausdrucks-kategorien $K_1 \dots K_n$ aus *Kompl* hervorgehen. Ein Adjektivattribut soll erst später (vor dem Nomen) ergänzt werden, deshalb fordert die Nebenbedingung am Anfang, daß $ADek1 = (-)$.

Auf das Nomen selbst kann ein Genitivattribut folgen, das im Komplementrahmen nicht genannt ist, wie in *Stein der Weisen*, oder ein dort genanntes Genitiv-Objekt, wie in *Ermordung des Cäsars*; das wird durch die Alternative (`[] ; np([3,-,],[gen])`) ausgedrückt. Für die im Komplementrahmen des Nomens vorgesehenen Objekte wird durch (`[] ; ...`) gesagt, daß keines davon in der Nominalphrase vorkommen muß.

Die Alternative `; { Kompl=[gen] }, []` besagt, daß ein erwartetes Genitivobjekt wie bei *Ermordung* : $N_{[gen],fem}^{sg,nom}$ durch das vorher erlaubte `np([3,-,],[gen])` erkannt werden soll; das dient hauptsächlich dazu, eine auf das Nomen folgende Nominalphrase im Genitiv vorrangig als Genitivobjekt zu betrachten, vgl. *Stein der Weisen* mit *Ermordung der Weisen*.

Die Alternative

```

⟨Erweiterung eines Nomens um ein Präpositionalobjekt: n1 --> praep, np⟩≡
; { Kompl=[Praep,KasP], % Angst vor den Hunden
  praeposition(Praep,KasP) },
  [Praep], np([_Pers,_Gen,_Num],[KasP])

```

besagt, daß bei Nomen, die ein mit Präposition anzuschließendes Objekt erfordern, im Komplementrahmen `[Praep,KasP]` sowohl die Präposition als auch der von ihr regierte Kasus auftreten müssen, z.B. *Angst* : $N_{[vor,dat],fem}^{sg,nom}$. Durch Ausführen der Bedingung `praeposition(Praep,KasP)` bei der Syntaxanalyse wird überprüft, ob die beiden Werte im Komplementrahmen tatsächlich eine Präposition und ihr Kasus sind (was getrennt festgelegt werden muß). Falls das der Fall ist, sollen die genannte Präposition und eine Nominalphrase im genannten Kasus auf das Nomen folgen.

Entsprechend werden bei anderen Werten im Komplementrahmen andere Ausdrucksarten für die Objekte erwartet. Neben den Objekten, die die Form von Nominalphrasen haben, können Nomina auch Objekte haben, die die Form von Sätzen oder Infinitiven haben. Dazu dienen die Alternativen

```

⟨Erweiterung des Nomens um Objektsätze und -Infinitive⟩≡
; { Kompl=[dass] },
  ( s([dass],[v1,Mod]) % These, daß sie rund ist|sei
  ; s([def],[vz,konj]) % These, sie *ist|sei rund

```

```

)
; { Kompl=[ob] },      % Untersuchung, ob sie lebt
  s([ob],[v1,ind])
; { Kompl=[qu] },      % Frage, wo Maria lebt
  s([qu],[v1,Mod])
; { Kompl=[infzu] },   % Absicht, zu rauchen
  s([infzu],[ ])

```

Hier müssen die Merkmale der Satzkategorie noch geändert werden, wenn sich bei der Entwicklung der Satzregeln herausstellt, daß diese vorläufigen Merkmale ungeeignet sind.

Als letzte Alternative soll ein Relativsatz $s([\text{rel}], [\text{Mod}])$ erlaubt sein. Hierbei müssen aber Genus und Numerus des (Bezugs-)Nomens mit dem Genus und Numerus des Relativpronomens⁷ übereinstimmen. Dazu sollten die Relativsätze im Artmerkmal sagen, welches Genus und Numerus ihr Relativpronomen hat; d.h. statt $s([\text{rel}], [\text{Mod}])$ brauchen wir $s([\text{rel}(\text{Gen}, \text{Num})], [\text{Form}])$; da wir bei Sätzen nur *ein* Artmerkmal *Satzdefinitheit* vorgesehen haben, verwenden wir hier komplexe Werte wie $\text{rel}(\text{mask}, \text{sg})$. Die entsprechende Grammatikregel sollte dann lauten:

$$N_{\text{Gen}}^{\text{Dekl}, \text{Num}, \text{Kas}} \rightarrow N_{\text{Gen}}^{\text{Num}, \text{Kas}}, S_{\text{rel}(\text{Gen}, \text{Num})}^{\text{Form}} \{ \text{Dekl} = (-) \},$$

wobei noch zu überlegen ist, was die Formmerkmale von Relativsätzen sein sollen. Außerdem sollte es erlaubt sein, einen Relativsatz auch an Nomina mit Objekten anzuschließen, aber in diesem Fall stehen die Objekte *vor* dem Relativsatz: *(die) Hoffnung auf den Lottogewinn, die mich enttäuscht hat*. Die Regel muß daher korrigiert werden zu:

(Regeln zur Erweiterung eines Nomens um Gen-Attribut, Objekt, Relativsatz (korrigiert)) ≡
 % N mit Komplement und Attribut: $n1([\text{Gen}], [\text{Adekl}, \text{Num}, \text{Kas}])$

```

n1([Gen],[Dekl,Num,Kas]) --> % Nomen+Komplement
  { Dekl = (-) },           % keine Adjektive vorhanden
  n([Kompl,Gen],[Num,Kas]), % ggf. Kompl = []
  ( [] ; np([3,-,-],[gen]) ), % opt. gen.poss/subj
  ( [] % Komplement optional: Hoffnung (auf NP)
  ; { Kompl=[gen] }, []     % gen-Objekt statt gen.subj
  ; { Kompl=[Praep,KasP],   % Angst vor den Hunden
    praeposition(Praep,KasP) },
    [Praep], np([_Pers,_Gen,_Num],[KasP])
  ; { Kompl=[dass] },
    ( s([dass],[v1,Mod]) % These, daß sie rund ist|sei
    ; s([def],[vz,konj]) % These, sie *ist|sei rund
    )
  ; { Kompl=[ob] },        % Untersuchung, ob sie lebt
    s([ob],[v1,ind])
  ; { Kompl=[qu] },        % Frage, wo Maria lebt
    s([qu],[v1,Mod])
  ; { Kompl=[infzu] },     % Absicht, zu rauchen
    s([infzu],[ ])
  ),
  ( [] ; s([rel(Gen,Num)], [Mod]) % Menschen, an die wir denken.
  ).

```

⁷allgemeiner: der den Relativsatz einleitenden Nominal- oder Präpositionalphrase, wie *Autor, dessen Vorliebe ihnen galt* oder *Job, an dem er Freude hat*

Nachdem ein Nomen um sein Objekt oder um einen Relativsatz erweitert ist, erlauben wir eine zweite Erweiterung um ein vorangestelltes Adjektiv (mit seinen Objekten):

```

⟨Erweiterung eines erweiterten Nomens um ein Adjektiv-Attribut⟩≡
n1([Gen],[Dekl,Num,Kas]) -->
  { Komp = pos }, % TODO: Komp=komp, sup
  ap([], [Komp,Dekl,GenD,Num,Kas]), % alte Hunde, die ...
  { GenD=(-) ; GenD=Gen },
  n1([Gen],[Dekl1,Num,Kas]),
  { Dekl1 = Dekl ; Dekl1 = (-) }.

```

Wir erhalten dadurch für Nominalphrasen der Form Det AP N NP.gen NP RelS immer (und nur) die Klammerung

```
(Det ( (Objekt A):AP ((N NP.gen Objekt) RelS):N1 ):N1) : NP
```

Da bei den Adjektiven nur die positiven Formen im Lexikon aufgeführt sind, wird der Komparativ durch die Nebenbedingung { Komp = pos } auf den Positiv eingeschränkt; man könnte das auch durch Vorgeben des Werts in ap([], [pos,Dekl,GenD,Num,Kas]) tun. Da die Adjektivformen im Deutschen im Plural nicht vom Genus des Bezugsnomens abhängen, wird in den Lexikoneinträgen im Plural der Genuswert - verwendet. Daher muß die Rektion der Adjektivform durch das Genus des Nomens in { GenD=(-) ; GenD=Gen } diesen Sonderfall berücksichtigen. Da beim gesamten Ausdruck eine Erweiterung um eine Adjektivphrase vorliegt, muß das Merkmal Dekl des Ausdrucks mit dem des Adjektivs übereinstimmen; bei dem inneren Ausdruck muß entweder die gleiche Adjektivdeklination verwendet werden, wie in *ein kleiner dummer Junge*, oder er enthält kein Adjektiv, *ein dummer Junge, der sich etwas einbildet*.

Wenn man die NP- und AP-Regeln verallgemeinern will, ist eine gewisse Vorsicht nötig:

- Bei Adjektivphrasen werden keine Linksergänzungen von Adjektiven zugelassen, die mit Adjektiven beginnen können, also auch keine Adjektive, deren vorangestelltes Objekt eine Nominalphrase (ohne Artikel) sein könnte, wie *Bücher liebende Kinder*. Da könnte man mit genaueren Regeln mehr erlauben, indem man z.B. explizit die Kategorien alternativ erlaubt, die als linke Ecke auftreten können, ohne eine Linksrekursion einzuführen. In AP → NP^{Kas} A^{Form}_[Kas] wie *der Mühen überdrüssig* müßte man die Alternativen von NP^{Kas} aufzählen, die nicht mit einer AP links beginnen können. (Hierzu könnte man definite von indefiniten Nominalphrasen unterscheiden, um dann *großer Mühen* : NP^{gen}_{indef,3,fem,pl} *überdrüssig* : AP ausschließen zu können.)
- Ebenfalls wegen der Linksrekursion werden werden keine vorangestellten NP^{gen}-Attribute als Determinatoren in NP = DetN erlaubt. Wenn man es doch machen will, muß man (wenn man Prolog als Parser benutzt) verlangen, daß die entstehende Nominalphrase nicht im Genitiv steht, da man sonst eine Linksrekursion beim Parsen hat.

```

⟨NP-Regeln mit linker NP.gen als Det⟩≡
np([Pers,Gen,Num],[Kas]) -->
  { kasus(Kas), Kas\=gen }, % kasus(Kas) belegt Kas!
  np([Pers,Gen2,Num2],[gen]), % des Autors (np-pron nötig!)
  { Pers=3 },
  n1([Gen],[Dekl,Num,Kas]), % alter Hund | grosse Vorlieben
  { Dekl=(-) ; Dekl=st }. % ggf. kein Adj in N1

```

An dieser Regel sieht man auch, daß die Kategorie NP^{gen} zu grob ist: man müßte hier die Personalpronomen ausnehmen (dafür gibt es ja Possessivpronomen), aber auch Nomina ohne Artikel: *der Natur beruhigende Wirkung*, aber nicht *Natur beruhigende Wirkung*. Die Regel ist „übergenerierend“, d.h. läßt auch ungrammatische Ausdrücke zu.

Andererseits wäre eine Erweiterung der Nominalphrasengrammatik wünschenswert. Bei der NP -Kategorie wird die Einteilung in drei oder vier Arten, (in)definite, quantifizierte, interrogative und relativierende, hier nicht berücksichtigt; statt NP_{def}^{Form} wird NP^{Form} benutzt. Das ist eine wesentliche Einschränkung für die Grammatik von Sätzen: für Fragesätze außer Ja/Nein-Fragen, z.B. *welches Kind schläft?*, wäre eine Regel wie

$$S_{qu}^{Temp,Mod,vz} \rightarrow NP_{qu,Pers,Gen,Num}^{nom} V_{[nom]}^{Pers,Num,Temp,Mod}$$

nötig, bei der man sicherstellt, daß eine interrogative Nominalphrase verwendet wird. Entsprechendes gilt für Relativsätze und für die Unterteilung von Adverbialen nach Definitheit.

3 Satzgrammatik

Für einfache Sätze verwenden wir die Kategorie $S_{Def}^{Temp,Mod,Vst}$, wobei mit

$$Def = \{def, qu, rel(Gen, Num) \mid genus(Gen), numerus(Num)\}$$

zwischen Aussagesätzen, Fragesätzen, und Relativsätzen unterschieden wird. Die Form von (einfachen) Sätzen variiert nach Tempus, Modus und Verbstellung, $ve = verberst$, $vz = verbzweit$, $v1 = verbletzt$. Allerdings sind Relativsätze auf die Verbletztstellung beschränkt.

(Satzregeln für transitive Verben, Aufgabe 5.1)≡

% 5.1.b) einfache (unkoordinierte) Saetze (mit Verb an zweiter Stelle: vz)

startsymbol(s(_Art, _Form)).
startsymbol(np(_, _)).

s([def], [Temp, Mod, vz]) --> % def = Aussagesatz
np([Pers, _Gen, Num], [nom]),
v([[nom, Kas]], [Pers, Num, Mod, Temp]), % Verb mit Objekt-NP
{ Temp = praes ; Temp = praet },
np([_Pers2, _Gen2, _Num2], [Kas]).

s([def], [Temp, Mod, vz]) --> % def = Aussagesatz
np([Pers, _Gen, Num], [nom]),
v([[nom, Prep, Kas]], [Pers, Num, Mod, Temp]), % Verb mit Objekt-PP
{ Temp = praes ; Temp = praet },
{ praeposition(Prep, Kas) },
[Prep],
np([_Pers2, _Gen2, _Num2], [Kas]).

% Andere Verbstellungen:

s([def], [Temp, Mod, v1]) --> % untergeordneter Aussagesatz
np([Pers, _Gen, Num], [nom]),
np([_Pers2, _Gen2, _Num2], [Kas]),

```

    { Temp = praes ; Temp = praet },
    v([[nom,Kas]], [Pers,Num,Mod,Temp]).

s([def],[Temp,Mod,ve]) --> % Konditionalsatz (Verb an erster Stelle)
    { Temp = praes ; Temp = praet },
    v([[nom,Kas]], [Pers,Num,Mod,Temp]),
    np([Pers,_Gen,Num],[nom]),
    np([_Pers2,_Gen2,_Num2],[Kas]).

% (e) Koordination von Aussagesätzen mit Verhinderung der Linksrekursion

s([Def],[-,Mod,Vst]) --> % Linksrekursion durch Temp != Temp1 verhindern
    { Temp1 = praes ; Temp1 = praet },
    s([Def],[Temp1,Mod,Vst]),
    ( [und] ; [oder] ),
    { Temp2 = praes ; Temp2 = praet },
    s([Def],[Temp2,Mod,Vst]).

s([ob],[-,Mod,vl]) -->
    [ob], s([def],[_Tmp,Mod,vl]).

% (f) Relativsaetze: man braucht s([rel(Gen,Num)],Form) !

% Form anpassen: in (np -> det,n,srel) hat srel die Form [Mod]
s([rel],[Mod]) --> s([rel],[Temp,Mod,vl]),
    { Temp=praes ; Temp=praet ; Temp=(-) }.

s([rel],[Temp,Mod,vl]) --> % Relativsatz mit RelPron.nom
    pron([rel],[Pers,_Gen,Num,nom]),
    { Pers = 3 },
    np([_Pers2,_Gen2,_Num2],[Kas]),
    { Temp = praes ; Temp = praet },
    v([[nom,Kas]], [Pers,Num,Mod,Temp]).

s([rel],[Temp,Mod,vl]) --> % Relativsatz mit RelPron != nom
    { Temp = praes ; Temp = praet },
    pron([rel],[3,_Gen2,_Num2,Kas]),
    np([Pers,_Gen,Num],[nom]),
    v([[nom,Kas]], [Pers,Num,Mod,Temp]).

```

Diese Regeln sind vorläufig; in den einfachen Sätzen sollen noch Adverbiale erlaubt werden. Dazu geben wir zuerst Konstruktionen für Adverbiale an und anschließend eine überarbeitete Version der Satzregeln.⁸

⁸Es ist auch ungeschickt, die Nebenbedingung zum Tempus *vor* dem Verb anzugeben: falls das Verb ein anderes Tempus hat, müssen die Konstituenten zwischen der Nebenbedingung und dem Verb neu analysiert werden. Die Nebenbedingung ist nicht nötig, solange man keine Satztempora behandelt, die vom Verbtampus verschieden sind.

4 Adverbiale

Adverbiale oder Umstandsbestimmungen sind Satzglieder, die das Prädikat oder den ganzen restlichen Satz modifizieren, d.h. seine Bedeutung ändern. Da die Bedeutung in unterschiedlichen Dimensionen geändert werden kann, liegt es nahe, Adverbiale in verschiedene Arten einzuteilen, z.B. temporale, lokale, modale, instrumentale. Die Form der Adverbiale ist unveränderlich, weshalb wir als Kategorie $AdvP_{Dim}$ bzw. $advp([Dim], [])$ verwenden, mit *temp*, *lok*, *kaus*, *inst* und anderen Werten der Adverbialdimension (Vgl. `adverbiale.pl` zu Aufgabe 6.2).

Für eine größere Grammatik wäre es nötig, Adverbiale auch nach ihrer Definitheit zu unterscheiden, also $AdvP_{Def,Dim}$ zu verwenden, um definite von interrogativen und relativierenden Adverbialen zu trennen, z.B. *das Buch liegt (hier : $AdvP_{Def,lok}$), die Stelle, (wo : $AdvP_{rel,lok}$) das Buch liegt*, und *(wo : $AdvP_{qu,lok}$) liegt das Buch?*. Einfachheitshalber beschränken wir uns auf definite Adverbiale und lassen das Definitheitsmerkmal weg.

Im Deutschen haben wir drei Möglichkeiten, Adverbiale zu bilden: einfache Adverbiale aus Adverbien, z.B. *deswegen*, solche aus einer (bedeutungsvollen) Präposition mit Nominalphrase, z.B. *wegen des schönen Wetters*, und Adverbialsätze, also solche, die aus bestimmten Subjunktionen mit einem Satz in Verbletzstellung bestehen, z.B. *weil das Wetter schön war*.⁹

Adverbien sind etwa:

(Auszug aus `adverbiale.pl`) ≡

```
adv([temp], []) -->
    [morgens] ; [mittags] ; [abends] ; [nachts]
    ; [oft] ; [selten] ; [immer] ; [nie]
    ; [gestern] ; [heute] ; [morgen].
adv([lok], []) -->
    [hier] ; [dort].
adv([kaus], []) -->
    [deshalb] ; [daher].
adv([instr], []) -->
    [damit] ; [dadurch].
adv([dur], []) -->
    [kurz] ; [lange].
adv([kond], []) -->
    [gegebenenfalls].
adv([neg], []) -->
    [nicht].                                % eingeschränkte Stellungen?
```

Ob es sinnvoll ist, dem Negationsadverb eine eigene Unterart von Adverbien zuzuordnen, ist natürlich fraglich; es gibt auch “negierte” Adverbien anderer Dimensionen, wie *niemals*, *nirgendwo*; $adv([neg], [])$ dient hier nur dazu, einige Verwendungen der Negation durch Satzregeln mit Adverbialen zu erfassen.

Manche Adverbien können noch durch eine vorangestellte Partikel modifiziert werden, z.B. erlauben wir *sehr lange* : $AdvP_{dur}$.

Bei den Adverbialen, die aus einer Präposition mit Nominalphrase gebildet werden, soll die Art des Adverbials an der Präposition abgelesen werden; dazu verwenden wir eine Wortart $P_{Dim,Kas}$, oder $prep([Dim,Kas], [])$, mit der Dimension des erzeugten Adverbials und dem

⁹Daneben gibt es noch adverbial verwendete Nominalphrasen, z.B. *eines schönen Tages* : $AdvP_{temp}$. Wir lassen sie weg, da sie nur mit einer feinen Klassifizierung der Nomen erkennbar wären und die Syntaxanalyse erschweren.

regierenden Kasus als Artmerkmalen. Allerdings ist die Adverbialdimension der Präposition nicht eindeutig: z.B. *an diesem Morgen*: *AdvP_{temp}* und *an dieser Wand*: *AdvP_{lok}*. Ohne zusätzliche Informationen über die Nomen akzeptiert man hiermit allerdings viele unsinnige Adverbiale, etwa *auf der Idee*: *AdvP_{lok}*.

Bei den Adverbialsätzen bestimmt die einleitende Subjunktion die Art des Adverbials:

```
(einfache (unkoordinierte) Adverbiale)≡
advp([Art], []) -->
  ( ([ ] ; [sehr], { Art=dur } ),
    adv([Art], [])
  ; prep([Art,Kas], []),
    np([-,-,-], [Kas])
  ; subj([Art], []),
    s([def], [-,ind,v1])
  ).
advp([kond], []) -->
  s([def], [praes,ind,ve]). % , ( [ ] ; [dann] ).
```

Bei den Konditionalsätzen gibt es den Sonderfall, daß Sätze ohne einleitende Subjunktion als konditionale Adverbiale verwendbar sind, wenn das Verb am Anfang steht: *Kräht der Hahn auf dem Mist, bleibt das Wetter (so), wie es ist*. Allerdings sind solche Konditionalsätze nur am Satzanfang erlaubt, also nicht etwa *Maria arbeitet, kräht der Hahn auf dem Mist*.

```
(Beispiele für Präpositionen und Subjunktionen)≡
prep([lok, dat], []) --> [auf].
prep([dir, akk], []) --> [auf].
prep([lok, dat], []) --> [hintere].
prep([instr,dat], []) --> [mit].
prep([temp, dat], []) --> [nach].
prep([temp, gen], []) --> [während].
prep([kaus, dat], []) --> [wegen].
prep([kond, gen], []) --> [im], ['Falle'].

subj([temp], []) --> [als].
subj([kond], []) --> [falls].
subj([temp], []) --> [während].
subj([kaus], []) --> [weil].
subj([temp], []) --> [wenn].
subj([final], []) --> [damit]. % um + vp.infzu
```

Nicht erfaßt werden Finalsätze mit *um-zu*, wie *um ein Buch zu lesen*.

Die obigen Regeln erlauben es, Adverbiale zu konstruieren. Um Adverbiale verwenden zu können, muß man in den Satzregeln sagen, an welchen Stellen welche Art von Adverbialen auftreten dürfen. Dazu muß man in den bisherigen Regeln für (einfache) Sätze an geeigneten Stellen eine (oder mehrere) *advp*-Konstituente(n) einfügen.

Hier sind Satzregeln für (unkoordinierte) Aussagesätze im Präsens und Präteritum, die ein oder zwei Adverbiale enthalten dürfen (allerdings nicht nebeneinander):

```
(saetze-mit-adverbialen.pl (nicht extrahieren))≡
% Regeln für einfache Sätze mit Adverbialen (Tempus = praes,praet)
```

% mit intransitivem Verb und Verbstellungen ve, vz, vl (erst/zweit/letzt)

```
s([def],[Temp,Mod,ve]) --> % def = Aussagesatz
  v([[nom]],[Pers,Num,Mod,Temp]),
  np([Pers,_Gen,Num],[nom]),
  ( [] ; advp([_Art],[ ]) ).
```

```
s([def],[Temp,Mod,vz]) --> % def = Aussagesatz
  np([Pers,_Gen,Num],[nom]),
  v([[nom]],[Pers,Num,Mod,Temp]),
  ( [] ; advp([_Art],[ ]) ). % opt.Adverb inkl 'nicht'
```

```
s([def],[Temp,Mod,vz]) --> % def = Aussagesatz
  advp([Art],[ ]), { Art \= neg }, % Adverb: 'nicht' ausnehmen!
  v([[nom]],[Pers,Num,Mod,Temp]), % Verb mit Objekt-NP
  np([Pers,_Gen,Num],[nom]),
  ( [] ; advp([Art2],[ ]), { Art \= Art2 } ).
```

```
s([def],[Temp,Mod,vl]) --> % def = Aussagesatz
  np([Pers,_Gen,Num],[nom]),
  ( [] ; advp([_Art],[ ]) ), % opt.Adverb inkl 'nicht'
  v([[nom]],[Pers,Num,Mod,Temp]).
```

% Satzregeln mit transitivem Verb, Adverbiale z.T. noch zu ergaenzen:

```
s([def],[Temp,Mod,ve]) --> % Konditionalsatz (Verb an erster Stelle)
  v([[nom,Kas]],[Pers,Num,Mod,Temp]),
  np([Pers,_Gen,Num],[nom]),
  ( [] ; advp([Art1],[ ]), { Art1 \= neg } ),
  np([_Pers2,_Gen2,_Num2],[Kas]),
  ( [] ; advp([Art2],[ ]), { Art2 \= Art1 } ).
```

```
s([def],[Temp,Mod,vz]) --> % def = Aussagesatz
  np([Pers,_Gen,Num],[nom]),
  v([[nom,Kas]],[Pers,Num,Mod,Temp]), % Verb mit Objekt-NP
  ( [] , {Art1=(-)} ; advp([Art1],[ ]), { Art1 \= neg } ),
  np([_Pers2,_Gen2,_Num2],[Kas]),
  ( [] ; advp([Art2],[ ]), { Art2 \= Art1 } ).
```

```
s([def],[Temp,Mod,vz]) --> % def = Aussagesatz
  advp([Art1],[ ]), % Advp im Vorfeld
  { Art1 \= neg },
  v([[nom,Kas]],[Pers,Num,Mod,Temp]), % Verb mit Objekt-NP
  np([Pers,_Gen,Num],[nom]),
  np([_Pers2,_Gen2,_Num2],[Kas]),
  ( [] ; advp([Art2],[ ]), { Art2 \= Art1 } ).
```

```

s([def],[Temp,Mod,v1]) --> % untergeordneter Aussagesatz
  np([Pers,_Gen,Num],[nom]), % ohne Subjunktion
  ( [] ; advp([Art1],[ ]) ),
  np([_Pers2,_Gen2,_Num2],[Kas]),
  ( [] ; advp([Art2],[ ]), { Art2 \= Art1 } ).
v([[nom,Kas]],[Pers,Num,Mod,Temp]).

% TODO: 2-stellige Verben mit Präpositionalobjekt:

s([def],[Temp,Mod,vz]) --> % def = Aussagesatz
  np([Pers,_Gen,Num],[nom]),
  v([[nom,Prep,Kas]],[Pers,Num,Mod,Temp]), % Verb mit Objekt-PP
  ( [] ; advp([Art1],[ ]) ),
  { praeposition(Prep,Kas) },
  [Prep],
  np([_Pers2,_Gen2,_Num2],[Kas]),
  ( [] ; advp([Art2],[ ]), { Art2 \= Art1 } ).

% Relativsaetze: man braucht s([rel(Gen,Num)],Form) !

% Form anpassen: in (np -> det,n,srel) hat srel die Form [Mod]
s([rel(Gen,Num)], [Mod]) -->
  s([rel(Gen,Num)], [Temp,Mod,v1]),
  { Temp=praes ; Temp=praet ; Temp=(-) }.

% TODO: Adverbiale einfügen:

s([rel(Gen,Num)], [Temp,Mod,v1]) --> % Relativsatz mit RelPron.nom
  pron([rel],[Pers,Gen,Num,nom]),
  { Pers = 3 },
  np([_Pers2,_Gen2,_Num2],[Kas]),
  { Temp = praes ; Temp = praet },
  v([[nom,Kas]],[Pers,Num,Mod,Temp]).

s([rel(Gen2,Num2)], [Temp,Mod,v1]) --> % Relativsatz mit RelPron =/= nom
  { Temp = praes ; Temp = praet },
  pron([rel],[3,Gen2,Num2,Kas]),
  np([Pers,_Gen,Num],[nom]),
  v([[nom,Kas]],[Pers,Num,Mod,Temp]).

```

Eine Einschränkung an die Reihenfolge verschiedener Adverbiale kann man durch Nebenbedingungen erzwingen, z.B. *manche Studenten lernen am Samstag in der Bibliothek*:

(Adverbialbedingung: Zeit vor Ort (ungetestet))≡

```

s([def],[Temp,Mod,vz]) --> % def = Aussagesatz
  np([Pers,_Gen,Num],[nom]),
  v([[nom]],[Pers,Num,Mod,Temp]),
  { Temp = praes ; Temp = praet },

```

([] ; advp([Art1], [])),
 ([] ; advp([Art2], [])), { adverbialordnung(Art1,Art2) }.

adverbialordnung(temp,lok).

5 Koordinationsregeln

Bei vielen Ausdrucksarten ist eine Koordination mit **und**, **oder**, **sowohl - als auch**, **weder - noch** und ähnlichen Koordinatoren möglich.¹⁰ Grob gesehen entsteht zwar ein Ausdruck der gleichen Kategorie wie die kombinierten Teilausdrücke, sodaß Grammatikregeln wie ($S \rightarrow S \text{ und } S$) oder ($NP \rightarrow NP \text{ und } NP$) naheliegen. Es gibt hier aber ein Problem, wenn man die Art- und Formmerkmale berücksichtigt: da sich die Merkmalwerte der Teilausdrücke in derselben Merkmalsdimension unterscheiden können, hat das Merkmal beim Gesamtausdruck keinen Wert, der sich direkt aus den Werten der Teilausdrücke ergäbe. Wir müssen also statt des Regelschemas

$$NP_{Pers,Gen,Num}^{Kas} \rightarrow NP_{Pers,Gen,Num}^{Kas} \text{ und } NP_{Pers,Gen,Num}^{Kas}$$

genauer sagen, wie sich die Merkmalwerte des Gesamtausdrucks aus denen der Konstituenten ergeben. Z.B. ist die Koordination zweier Nominalphrasen schon möglich, wenn sie den gleichen Kasus haben:

$$NP_{Pers,Gen,Num}^{Kas} \rightarrow NP_{Pers_1,Gen_1,Num_1}^{Kas} \text{ und } NP_{Pers_2,Gen_2,Num_2}^{Kas},$$

wobei aber die anderen Merkmale $Pers, Gen, Num$ durch eine Nebenbedingung wie

$$\{p(Per_1, \dots, Num_2, Pers, Gen, Num)\}$$

mit einer Bedingung p festgelegt werden müssen. Bei Konfliktfällen vergeben wir den Wert -, außer beim Numerus, wo (wegen der Kongruenz zwischen Subjekt und Verb im Numerus) der Wert pl nötig sein kann, z.B.

$$((\text{der Mond und die Sterne}) : NP_{3,-,pl}^{nom} \text{ leuchten} : V_{[nom]}^{3,pl,ind,praes}) : S.$$

Ein Wert - zeigt oft, daß entsprechende Ausdrücke eine eingeschränkte Distribution haben:

$$(\text{mich} : NP_{1,-,sg}^{akk} \text{ oder euch} : NP_{2,-,pl}^{akk}) : NP_{-,-,-}^{akk}$$

ist eine Nominalphrase, bei der Person, Genus und Numerus undefiniert sind, die aber dennoch als Akkusativ-Objekt problemlos verwendbar ist. Dagegen ist

$$(\text{ich} : NP_{1,-,sg}^{nom} \text{ oder ihr} : NP_{2,-,pl}^{nom}) : NP_{-,-,-}^{nom}$$

als Subjekt eines Satzes unbrauchbar, weil Nominalphrasen $NP_{-,Gen,-}^{nom}$ für das Verb keine Werte für Person und Numerus vorgeben, also unklar ist, welche Verbform man nehmen sollte.

Man muß sich also jeweils überlegen, in welchen Merkmalsdimensionen die zu koordinierenden Ausdrücke übereinstimmen müssen und in welchen nicht.

Dazu kommt das allgemeine Problem, daß die Koordinationsregeln mit *und* und *oder* linksrekursiv sind, wenn die Merkmale der linken Konstituente und die des Gesamtausdrucks gleich sein können. In diesem Fall terminiert ein Top-Down-Parser (bzw. die Beweissuche von Prolog) nicht. Um

¹⁰Es sind auch Verbindungen verschiedener Ausdrucksarten möglich, wie *Die Euler'sche Vermutung ist ((ein bekanntes Problem):NP und (sehr schwierig):AP):?*, was wir aber ignorieren.

das zu vermeiden, sollte man im linken Koordinationsglied nur einen *einfachen* (unkoordinierten) Ausdruck erlauben.

Aber auch unabhängig vom Problem der Linksrekursion ist es klug, nur einfache Ausdrücke zu koordinieren: andernfalls kann die Anzahl der Analysen exponentiell in der Anzahl der Wörter eines Ausdrucks werden (wie bei arithmetischen Ausdrücken mit $+$, \cdot und Zahlen ohne implizite Klammerregeln). Das unten diskutierte Artmerkmal *Figur* zur Unterscheidung von einfachen (*si* für simple) und koordinierten (*co* für koordinierte) Ausdrücke ist also auch sinnvoll, wenn man keinen Top-Down-Parser verwendet.

5.1 Koordination von Sätzen

Auf den Vorlesungsfolien ist folgende Koordinationsregel für Sätze (Aufg.5.1) angegeben:

(Koordination von Sätzen mit Verhinderung der Linksrekursion) \equiv
 $s([Def], [-, Mod, Vst]) \rightarrow$ % Linksrekursion durch Temp \neq Temp1 verhindern
 { Temp1 = praes ; Temp1 = praet },
 $s([Def], [Temp1, Mod, Vst]),$
 ([und] ; [oder]),
 { Temp2 = praes ; Temp2 = praet },
 $s([Def], [Temp2, Mod, Vst]).$

Hier wird verlangt, daß die Teilsätze im Modus und in der Verbstellung übereinstimmen. Der Gesamtsatz braucht diese Merkmale, z.B. weil bei indirekter Rede alle Teilsätze im Konjunktiv stehen müssen und weil man die gleiche Verbstellung bei der Koordination von Aussage-, Relativ- oder Ja/Nein-Fragesätzen braucht.

Im Tempus müssen die Teilsätze nicht übereinstimmen, z.B. *Sie hat das Buch gelesen oder sie wird es bis morgen lesen*, Wir vergeben immer den Pseudowert $-$, auch wenn die Teilsätze denselben Tempuswert haben. Um die Linksrekursion auszuschließen, wird zusätzlich verlangt, daß die Teilsätze einen echten Tempuswert haben, und daher nicht selbst wieder koordiniert sein können, d.h. wir erlauben nur die Koordinierung *einfacher* Sätze.

Allerdings wird die Analyse mit einer solchen Koordinationsregel langsam. Das liegt wohl daran, daß mit { Temp1 = praes ; Temp1 = praet } erzwungen wird, daß vom gesuchten Teilsatz der Anfang bis zum Verb *zweimal* analysiert wird, da Temp1 mit dem Verbtempus verglichen werden muß. Man kann aber die Nebenbedingung nicht hinter $s([Def], [Temp1, Mod, Vst])$ stellen, weil dadurch die Koordinationsregel linksrekursiv würde.

Es ist daher effizienter, eine neue Kategorie koordinierter Sätze einzuführen, etwa¹¹ $S_{Def}^{Mod, Vst}$, und dann Satzkoordinationen durch die (nicht linksrekursiven) Regeln

$$S_{Def}^{Mod, Vst} \rightarrow S_{Def}^{Temp, Mod, Vst}, S_{Def}^{Mod, Vst} \\ | S_{Def}^{Temp, Mod, Vst} (und | oder) S_{Def}^{Temp, Mod, Vst}$$

zu erlauben. Damit lassen sich Koordinationen von mehreren *einfachen* Sätzen erkennen, z.B. *der Hund schläft und die Katze jagt die Maus* und *sie spielen, er schläft und sie jagt die Maus*. Man muß dann aber in allen Grammatikregeln, wo bisher nur einfache (Subjekt- oder Objekt-) Sätze erlaubt sind, eine Alternative ($S_{Def}^{Temp, Mod, Vst} | S_{Def}^{Mod, Vst}$) vorsehen – was ebenfalls zur Doppelanalyse des ersten Satzes führt. Der erste Teilsatz wird aber nicht mehr mit beiden Tempuswerten zu

¹¹da die Teilsätze im Tempus nicht zu kongruieren brauchen, oder $S_{Def, coord}^{Temp, Mod, Vst}$, wenn das doch der Fall ist

analysieren versucht.

```

(Koordination von Sätzen mit eigener Kategorie)≡
s([Def],[Vst,Mod]) -->
  s([Def],[Temp1,Mod,Vst]),
  ( ([und] ; [oder] ),
    s([Def],[Temp2,Mod,Vst])).
; % [' ',''] beim Einlesen gelöscht
  s([Def],[Vst,Mod])
).

```

5.2 Koordination von Nominalphrasen

Besser als die beschränkte Lösung auf den Vorlesungsfolien ist es, mit dem Merkmal *Figur* = {*si,co*} zwischen einfachen und koordinierten Nominalphrasen zu unterscheiden und dann die Koordinierung einfacher Nominalphrasen (hier: $NP_{Pers,Gen,Num}^{Kas}$ statt $NP_{Pers,Gen,Num,si}^{Kas}$) ohne Linksrekursion so zu formulieren:

```

(Koordination von Nominalphrasen)≡
np([Per,-,Num,co],[Kas]) -->      % co = koordiniert
  np([Per1,-,Num1],[Kas]),
  ( [Conj],
    np([Per2,-,Num2],[Kas])
  ; % [' ',''], % beim Einlesen gelöscht
    np([Per2,-,Num2,co],[Kas])
  ),
  { agreeOrNone(Per1,Per2,Per) },
  { Conj = und, Num = pl
  ; Conj = oder, agreeOrNone(Num1,Num2,Num)
  }.

```

Damit in den Satzregeln überall einfache oder koordinierte Nominalphrasen verwendbar sind, wäre es am einfachsten, dort die Kategorie $NP_{Pers,Gen,Num}^{Kas}$ durch $NP_{Pers,Gen,Num,Fig}^{Kas}$ mit der Variablen *Fig* zu ersetzen und in der Koordinationsregel durch $NP_{Pers,Gen,Num,si}^{Kas}$. (Das ist bisher nicht gemacht, sodaß man zwar selbständige koordinierte Nominalphrasen analysieren kann, aber nicht als Konstituenten von Sätzen.)

5.3 Koordination von Adjektivphrasen

Die Adjektivkoordination kann man mit einer Kategorie $AP_{Fig}^{Komp,Dekl,Gen,Num,Kas}$ natürlich ebenfalls ohne Linksrekursion beschreiben. Allerdings wird durch die Regel $N' \rightarrow AP N'$ schon eine Reihung von attributiven Adjektivphrasen erlaubt (ohne Konjunktion zwischen den letzten beiden). Das kann man unterdrücken, indem man im eingebetteten N' durch den Wert (-) seiner Adjektivdeklination ein Adjektiv-Attribut ausschließt und beim Attribut einfache und koordinierte Adjektivphrasen zuläßt:

```

(Einfaches oder koordiniertes Adjektiv-Attribut)≡
n1([Gen],[Dekl,Num,Kas]) -->
  { Komp = pos }, % TODO: Komp=komp, sup
  ap([Fig],[Komp,Dekl,GenD,Num,Kas]), % alte Hunde, die ...
  { GenD=(-) ; GenD=Gen },

```

```
n1([Gen],[-,Num,Kas]). % keine AP-Modifikation!
```

Mit den Figur-Merkmalen kann man dann die Linksrekursion ausschließen und nur einfache Adjektivphrasen koordinieren:

```
<Koordination von Adjektivphrasen>≡
% Koordination von einfachen Adjektivphrasen
ap([co],[pos,Dekl,Gen,Num,Kas]) -->
  ap([si],[pos,Dekl,Gen,Num,Kas]),
  ( ([und] ; [oder] ; []), % [] statt [' ','']
    ap([si],[pos,Dekl,Gen,Num,Kas])
  ; % [' ',''] beim Einlesen gelöscht
    ap([co],[pos,Dekl,Gen,Num,Kas])
  ).
```

Die bisherigen Regeln für Adjektivphrasen sind zu Regeln für einfache Adjektivphrasen zu ändern, hier Einfachheit halber durch

```
<Koordination von Adjektivphrasen>+≡
ap([si],[pos,Dekl,Gen,Num,Kas]) -->
  ap([], [pos,Dekl,Gen,Num,Kas]). % unnötige ap([],_-)-Knoten!
```

5.4 Koordination von erweiterten Nomina

Wenn man Koordinationsregeln für erweiterte Nomina $N'_{Gen}{}^{Dekl,Num,Kas}$ einbauen will, sollte man sich wieder darauf beschränken, daß die Koordinationsglieder *einfache* erweiterte Nomina sind. Eine Möglichkeit dazu wäre, das Genusmerkmal auf (-) zu setzen und bei den Gliedern einen echten Genuswert $g \in \{mask, fem, neut\}$ zu verlangen. Die Adjektivdeklinationsklasse und der Kasus der Glieder müssen wohl kongruieren. Im Numerus können die Glieder verschieden sein: *Sie verließ Land und Leute*¹².

Mit einer Kombination von erweiterten Nomina ergeben sich aber Mehrdeutigkeiten wie in

(europäische (Männer:N' und Frauen:N'):N'):N'
((europäische Männer):N' und Frauen:N'):N'

zusätzlich zu

((europäische Männer):NP und Frauen:NP):NP

Man könnte die Koordination *nicht* adjektivisch erweiterter Nomen zu einem N' erlauben:

```
<Koordination nicht adjektivisch erweiterter Nomen>≡
% nicht adjektivisch erweiterte Nomen:
n1([Gen],[Adekl,Num,Kas]) -->
  n([], [Gen1],[Num1,Kas]),
  ( ([und] ; [oder] ),
    n1([Gen2],[Adekl,Num2,Kas])
  ).
```

¹²Vielleicht behandelt man das besser als Koordination von Nomina $N'_{Gen}{}^{Num,Kas}$; andererseits sollte das eine Wortart sein und keine Koordination erlauben.

```

; % [,'], % beim Einlesen gelöscht
    n1([Gen2],[Adekl,Num2,Kas])
),
{ agreeOrNone(Gen1,Gen2,Gen), agreeOrNone(Num1,Num2,Num) }.

```

Aber das führt immer noch zu einigen Mehrdeutigkeiten. Außerdem würde nicht erlaubt, daß das linke Koordinationsglied Objekte enthielte oder durch einen Relativsatz modifiziert wäre. (Das könnte man durch eine Alternative

```
{Adekl = st ; Adekl = sw ; Adekl = ge}, n1([Gen1],[-,Num1,Kas])
```

statt des einfachen Nomens erlauben, auf Kosten der Effizienz.)

Wir lassen daher keine N' -Koordination zu.

5.5 Zusammenfassung der Koordinationsregeln

Wir fassen die Koordinationsregeln und die Beschränkungen linksrekursiver Regeln zusammen:

<coordination2.pl>≡

```
% Koordination von Nominalphrasen und Sätzen    HL, 22.1.2017
```

```
agree(X,Y,Z) :- ( X = Y, Z = X ).
```

```
% agreeOrNone(+X,+Y,?Z)
```

```
agreeOrNone(X,Y,Z) :-
    ( agree(X,Y,Z) -> true ; (X \= Y, Z = (-)) ).
```

```
% harmlose (nicht linksrekursive) Koordinationen
```

```
np([Per,Gen,pl],[Kas]) -->
    [sowohl], np([Per1,Gen1,_],[Kas]),
    [als], [auch], np([Per2,Gen2,_],[Kas]),
    { agreeOrNone(Per1,Per2,Per),
      agreeOrNone(Gen1,Gen2,Gen) }.

```

```
np([Per,Gen,pl],[Kas]) --> % oder Num1=Num2=sg ?
```

```
[weder], np([Per1,Gen1,_Num1],[Kas]),
[noch], np([Per2,Gen2,_Num2],[Kas]),
{ agreeOrNone(Per1,Per2,Per),
  agreeOrNone(Gen1,Gen2,Gen) }.

```

<Koordination von Nominalphrasen>

<Koordination von Sätzen mit eigener Kategorie>

<Einfaches oder koordiniertes Adjektiv-Attribut>

<Koordination von Adjektivphrasen>

*/**

<Koordination nicht adjektivisch erweiterter Nomen>

**/*

<NPgen als Determinator>

Keine Koordinationsregel ist die Verwendung von Nominalphrasen im Genitiv als Determinator

in Nominalphrasen, was im allgemeinen ein linksrekursive Konstruktion ist, die wir (ineffektiv, durch Alternativen) einschränken:

```

⟨NPgen als Determinator⟩≡
% NP mit NP.gen als Det, Linksrekursion vermieden:
np([Pers,Gen,Num],[Kas]) -->
  { kasus(Kas), Kas\=gen },
  np([Pers,_Gen2,_Num2],[gen]), % des Autors grosse Vorliebe
  { Pers=3 },
  n1([Gen],[Dekl,Num,Kas]), % alter Hund
  { Dekl=(-) ; Dekl=st }. % ggf. kein Adj in N1 (st:15.12.16)

```

6 Automatische Erzeugung von Satzregeln für einfache Sätze

Unter einer „Metaregel“ hatten G.Gazdar u.a.[1] in *Generalized Phrase Structure Grammar* (1985, Kapitel 4) eine bestimmte Art verstanden, Grammatikregeln, die gewissen Mustern entsprechen, in andere Grammatikregeln umzuwandeln, z.B. eine Regel für einfache Sätze im Aktiv in eine Regel für Sätze im Passiv.

Wir können diese Idee benutzen, um Grammatikregeln für einfache deutsche Sätze im Aktiv aus möglichen Komplementrahmen von Vollverben zu *berechnen*, denn

1. für jede Verbstellung und jedes Satztempus sagen die Schemata zur Felderstruktur, aus welchen finiten und infiniten Formen von Voll-, Hilfs- und Modalverben der linke und rechte Prädikatsteil bestehen,
2. aus dem Komplementrahmen und dem Prädikatsteil V^{fn} lassen sich Listen von Kategorien mit passend belegten Merkmalen, insbesondere *Pers,Num* des Subjekts, erzeugen und (erweitert um Kategorien für Adverbiale) auf Vorfeld, Mittelfeld, Nachfeld verteilen,
3. und durch Einfügen der Prädikatsteile zwischen diese Belegung der Felder ergibt sich eine Folge *Konstituenten* von Konstituentenkategorien und aus ihr eine DCG-Regel

$$(S_{Art}^{Temp,Mod,Vst} \rightarrow \text{Konstituenten}),$$

wobei *Temp* und *Mod* an den Prädikatsteilen und $Art \in \{def, rel(Gen,Num), qu\}$ von der Verbstellung und den Definitheitsmerkmalen der Konstituenten abhängt.

Dasselbe kann man natürlich auch für Passivsätze machen, wozu man den für das Aktiv gedachten Komplementrahmen des Verbs geeignet ändert. Ein großer Vorteil gegenüber von Hand erstellten Regeln besteht darin, daß man viele Wort- bzw. Konstituentenstellungen bei der Erzeugung von *Konstituenten* berücksichtigen und im Erzeugungsprogramm explizit formulieren kann. Die Erzeugung der Satz schemata aus den Komplementrahmen der Verben ist quasi eine Implementierung des „X-bar-Schemas“:

$$\bar{X} \rightarrow \dots (\text{Komplemente}) \dots X \dots (\text{Komplemente}) \dots$$

Wir beschränken uns erstmal bei den Verbarten, indem wir keine Verben mit abtrennbarem Präfix berücksichtigen und in den Komplementrahmen nur die Komplementabkürzungen

nom,akk,refl(akk),daß,ob

erlauben. Die Verwendbarkeit der erzeugten Satzregeln hängt natürlich davon ab, welche Verbeinträge im Lexikon vorhanden sind, d.h. welche DCG-Regeln $v(\text{Art}, \text{Form}) \rightarrow [\text{Vollform}]$. man direkt angegeben hat!

Eine Metaregel wird in der Form $(\text{Cat} \text{ --> } \text{Konstituenten}) :- \text{Rumpf.}$ angegeben; wenn Prolog den Rumpf ausführt, werden variable Teile in Cat und die Variable(!) Konstituenten belegt, wodurch Kategorie und Konstituentenfolge einer DCG-Regel $\text{Cat} \rightarrow \text{Konstituenten}$ entstehen.

(Erzeugung von DCG-Regeln)≡

```
:- op(1100,xfx,'-->'). % neues Infix-Relationssymbol

ersetze_metaregeln :-
    tell('satzregeln.pl'),      % Datei 'satzregeln.pl' öffnen
    (A --> B),                  % Instanz von (A --> B) beweisen
    portray_clause((A --> B)), % und als DCG-Regel
    nl, fail.                   % in 'satzregeln.pl' schreiben
ersetze_metaregeln :-
    told.                       % Datei 'satzregeln.pl' schliessen.
```

Ist eine Datei mit Metaregeln geladen, so wird durch

```
?- ersetze_metaregeln.
```

die Datei `satzregeln.pl` mit den gewünschten DCG-Regeln erzeugt (und ggf. überschrieben).

Zur Erzeugung von Grammatikregeln für einige einfache Aussagesätze und Ja/Nein-Fragen dient diese, im Folgenden erläuterte Metaregel:

(Metaregel für einfache Ja/Nein-Fragen und Aussagen)≡

```
(s([Def],[Temp,Mod,Vst]) --> Konstituenten) :-
    ( Vst=ve, Def=qu
    ; Vst=vz, Def=def
    ; Vst=vl, Def=def
    ),
    rahmenauswahl(Rahmen), % Auswahl für Teilgrammatik!
    praedikatsteile(VFIN,VINF, % = Praedikatsteile
                    Rahmen,Pers,Num,Temp,Mod,aktiv),
    expandiereSubjCat([Pers,Num],Rahmen,SubjRahmen),
    anordnung(SubjRahmen,SubjRahmenPerm),
    nachfeld(SubjRahmenPerm,SubjRahmenPerm2,Nachfeld),
    verbstellung(SubjRahmenPerm2,Vst,RahmenPraed),
    append(RahmenPraed,Nachfeld,RahmenPraed2),
    expandiere([Pers,Num],RahmenPraed2,VFIN,VINF,Cats),
    listToTuple(Cats,Konstituenten).
```

Hier wird am Anfang mit `Def` und `Vst` die Art und z.T. Form der einfachen Sätze festgelegt, für die man Grammatikregeln erzeugen will: für Interrogativsätze mit Verberststellung (Ja/Nein-Fragen) und für definite Sätze (Aussagesätze) in Verberst- und Verbletztstellung.

Im Prädikat `rahmenauswahl/1` legt man eine Reihe von Komplementrahmen fest. Daraus wird mit `rahmenauswahl(Rahmen)` ein Komplementrahmen gewählt, also die Verbart, für die einfache Satzregeln erzeugt werden sollen. Im nächsten Schritt,

```
praedikatsteile(VFIN,VINF,Rahmen,Pers,Num,Temp,Mod,aktiv),
```

werden aus einer vorgegebenen Tabelle der finite und der infinite Prädikatsteil entnommen, also VFIN und VINF belegt, und das darin auftretende Vollverb auf die gewünschte Verbart eingeschränkt, indem man den vorher gewählten **Rahmen** einbaut; an den Prädikatsteilen werden die Finitheitsmerkmale **Pers**, **Num** und das Satztempus **Temp** und der Modus **Mod** abgelesen.¹³ Zum Beispiel wird durch das schematische Faktum –vgl. die Tabelle auf den Vorlesungsfolien S.214/216–

```
praedikatsteile(v([haben],[Pers,Num,Mod,praes]),
                [v([Rahmen],[part2])),
                Rahmen,Pers,Num,perf,Mod,aktiv].
```

von **praedikatsteile/8** gesagt, daß für Sätze im Perfekt (**Temp = perf**) der finite Prädikatsteil VFIN eine finite Form des Perfekthilfsverbs¹⁴ *haben* im Präsens, der infinite Prädikatsteil VINF aus dem Partizip-2 des Vollverbs mit dem gegebenem Komplementrahmen **Rahmen** besteht (in anderen Fällen aus mehreren Wörtern, deshalb als Liste).

Neben den Prädikatsteilen muß die gewünschte Satzregel noch die Konstituenten für Subjekt und Objekte vorsehen, die im Komplementrahmen durch Abkürzungen wie **nom**, **refl(akk)**, **daß** angedeutet sind. Diese Abkürzungen müssen zu Kategoriensausdrücken mit Merkmalen „expandiert“ und geeignet angeordnet werden, z.B. wird **nom** durch **expandiereSubjCat/3** zur Kategorie **np([Pers,Gen,Num],[nom])** expandiert, wobei die Finitheitsmerkmale **Pers** und **Num** passend zum finiten Prädikatsteil eingesetzt werden. Mit **anordnung/**, **nachfeld/3** und **verbstellung/3** wird eine Anordnung und Verteilung der Komplementabkürzungen auf Vorfeld, Mittelfeld, Nachfeld berechnet und Markierungen **vfin** und **vinf** für die Prädikatsteile eingefügt. So erhält man z.B. aus einem Komplementrahmen **[nom,akk,daß]** für die Verbzweitstellung u.a.

```
RahmenPraed2 = [np([Pers, _G, Num], [nom]), vfin, akk, kor(daß), vinf, daß]
```

wo das Subjekt im Vorfeld, das Akkusativobjekt und ein Korrelat des Objektsatzes im Mittelfeld, und der Objektsatz im Nachfeld vorgesehen sind. Hieraus erstellt **expandiere/5** mit Hilfe der Finitheitsmerkmale (für reflexive Objekte) und den Prädikatsteilen VFIN und VINF eine Liste **Cats** der Konstituentenkategorien, die von **listToTuple/2** in ein Tupel **Konstituenten**, d.h. in die rechte Seite der zu bildenden Grammatikregel, umgewandelt wird. Die genaue Definition der Hilfsprädikate findet man in **metaregeln.pl**.

Die Anzahl der erzeugten Satzregeln hängt hauptsächlich von der Anzahl der erlaubten Komplementrahmen, deren Längen, und den erlaubten Anordnungen ab, während die Anzahl der Satztempora und Verbstellungen nur einen fixen Faktor dazu beitragen. (In **metaregeln.pl** sind sechs 2-stellige, ein 1-stelliger und ein 3-stelliger Komplementrahmen mit nur einer Anordnungsvariante vorgesehen, was 1100 Satzregeln ergibt; mit den Regeln für untergeordnete Sätze mit einleitender Subjunktion werden es 1800 Regeln. In beiden Fällen sind die Satzregeln *ohne* Modalverben gemeint.)

Für die Erzeugung von Regeln für einfache Relativsätze braucht man eine eigene Metaregel, die sicherstellt, daß die Merkmale **Gen** und **Num** der Satzart **rel(Gen,Num)** vom Relativpronomen stammen und eine Verbletzstellung vorliegt. Dazu erzeugen wir die Satzregel für einen definiten Verbletztsatz, der mit einer Nominalphrase von passendem Genus und Numerus beginnt, und ersetzen dann die Nominalphrase durch ein Relativpronomen:

<Metaregel für einfache Relativsätze>≡

¹³Um Regeln für Passivsätze zu erzeugen, muß man den gewählten (Aktiv-) **Rahmen** ändern und andere Belegungen von VFIN und VINF verwenden.

¹⁴Um das Perfekthilfsverb vom jeweiligen Vollverb abhängig zu machen, sollte aus der Verbart auch das Perfekthilfsverb hervorgehen, was man durch **v([PerfAux,Rahmen],Form)** erreichen kann. Dann kann man die Klausel von **praedikatsteile/8** verallgemeinern, indem man **PerfAux** statt **haben** einsetzt.

```
(s([rel(Gen,Num)], [Temp,Mod,v1]) -+-> Konstituenten) :-
    (s([def], [Temp,Mod,v1]) -+-> (np([Pers,Gen,Num], [Kas]), Rest)),
    Konstituenten = (pron([rel], [Pers,Gen,Num,Kas]), Rest).
```

Auf diese Weise bekommt man allerdings keine Relativsätze, die mit einer relativierenden Präpositionalphrase oder relativierenden Possessivkonstruktion beginnt, z.B. *an dem* oder *an dessen Ergebnis*.

Wenn die erzeugten Satzregeln auch Adverbiale vorsehen sollen, muß man weitere Elemente wie `advp` (oder genauere wie `kaus`, `lok`) in die Abkürzungsliste einbauen, bei der Anordnung berücksichtigen, und zu passenden Kategorien expandieren. Dabei ist es nützlich, wenn man `advp` –außer im Vorfeld– zu einer optionalen Kategorie (`[] ; advp([Art], [])`) expandiert, sodaß die Anzahl der Regeln nicht verdoppelt wird. Siehe dazu `metaregelnAdv.pl`, aber hier werden schnell so viele Regeln (mit Korrelaten und ins Nachfeld verschobenen Adverbialsätzen) erzeugt, daß das Parsen von Beispielen zu langsam wird.

6.1 Erzeugung einer Datei von DCG-Regeln für einfache Sätze

Auf den Vorlesungsfolien (S.240) ist folgende Prolog-Datei angegeben, die die Erzeugung von Satzregeln und das Laden verschiedener Teilgrammatiken steuert.

Beim Laden von `erzeugte-grammatik.pl` wird die Datei `metaregeln.pl` mit den obigen Metaregeln geladen, mit der eine Datei `satzregeln.pl` von DCG-Regeln erzeugt wird; dann wird mit `DCGparse/parser.pl` der Übersetzer von DCG-Regeln in Parseprädikate (mit Ausgabe von Syntaxbäumen) geladen; anschließend werden die Grammatiken für Nominalphrasen, Adverbiale und einfache Sätze geladen:

```
<erzeugte-grammatik.pl>≡
% Autor: Hans Leiß, LMU, CIS, Syntax nat.Sprachen, WS 2016/17

% Aus Metaregeln DCG-Satzregeln erzeugen und in satzregeln.pl schreiben:
:- [metaregeln].
:- ersetze_metaregeln.

% DCG-Regeln in Prolog-Klauseln
:- ['DCGparser/parser.pl'].

% Grammatiken laden:

:- ['np.folien',
    adverbiale,
    satzregeln
    ].

% Lexika laden:

:- [lexikon_np,
    lexikon-beispielnominalphrasen,
    lexikon-beispieladverbiale,
    lexikon-beispielsaetze,
    lexikon-v
```



```

].

% Startsymbole s/2 und np/2 überschreiben (falls nötig):

startsymbol(advp([_Art], [])).
startsymbol(np([_Pers, _Gen, _Num], [Kas])).
startsymbol(s([_Art], [Temp, Mod, Vst])). % einfache Sätze

```

6.2 Erzeugung von Regeln für einfache Sätze mit Adverbialen

Eine Variante, durch die Adverbiale verschiedener Art und Anordnungsbeschränkungen berücksichtigt werden, ist in `metaregelnAdv.pl` und `erzeugte-grammatikAdv.pl` angelegt.

Hierbei darf man nicht zu viele mögliche Positionen von Adverbialen erlauben, damit der Parser nicht überfordert wird. (Ein Anfang dazu ist gemacht, aber nicht auf der Kursseite.)

7 Zusammengesetzte Grammatik

Die erzeugten Regeln passen nicht in allen Details zu den von Hand geschriebenen Regeln. Wenn man z.B. die Koordinationsregeln so macht, daß zwischen einfachen und koordinierten Nominalphrasen, Adjektivphrasen oder Sätzen durch ein Artmerkmal unterschieden wird (was sinnvoll ist), dann sollten in den Satzregeln sowohl einfache wie koordinierte Nominalphrasen erlaubt werden, ebenso sollten in (einfachen) Nominalphrasen einfache und koordinierte Relativsätze und Adjektivphrasen erlaubt werden. Dazu müssen die Kategoriensymbole so angepaßt werden, daß sie das neue Artmerkmal enthalten.

Das erfordert mehr Muße, als ich am Semesterende hatte, und würde vermutlich auch die Rechenzeit für die Analyse von Beispielsätzen stark erhöhen: bei jedem Satz und jeder Nominalphrase müßten *auch* die Koordinationsregeln versucht werden, auch wenn keine Koordination in der Eingabe vorkommt. Die verwendete Top-Down-Strategie von Prolog braucht dabei zu lange (nicht für die ersten Analysen, aber um festzustellen, daß es keine weiteren Analysen gibt), und man sollte einen Tabellenparser (wie den Earley-Parser) verwenden.

In der folgenden, einigermaßen abgerundeten Zusammensetzung der von Hand geschriebenen Regeln zu Nominalphrasen und Koordinationen und der automatisch erzeugten Regeln für einfache Sätze ist die Erweiterung der Kategorien um ein Artmerkmal $Figur = \{si, co\}$ zur Unterscheidung zwischen *simplen* und *koordinierten* Ausdrücken gemacht. Es werden also die Kategorien

$$NP_{Pers, Gen, Num, Figur}^{Kas}, \quad AP_{Fig}^{Komp, Pers, Gen, Num, Kas} \quad \text{und} \quad S_{Def, Figur}^{Temp, Mod, Vst}$$

verwendet und ein paar Änderungen an den Regeln und Metaregeln vorgenommen. Der genaue Text folgt, aber es wurde nicht alles gründlich getestet.¹⁵

7.1 Ladefile

Die Gesamtgrammatik besteht aus mehreren Dateien,

¹⁵Anscheinend funktionieren Präpositionalobjekte nicht richtig mit dem Prolog-Top-Down-Parser; mit einem Earley-Parser gab es keine Probleme.

1. einer Ladedatei, `syntax16-grammatik.pl`, die
2. der Datei `nominalphrasen.pl` mit den Grammatikregeln zu Nominalphrasen, Adjektivphrasen und Adverbialen,
3. der Datei `koordinationen.pl`, den Grammatikregeln zur Koordination von Nominalphrasen, Adjektivphrasen, und Sätzen, und
4. der Datei `saetze.pl`, den aus den Metaregeln in `saetze-meta.pl` erzeugten Grammatikregeln für Aussagesätze, Relativsätze und Fragesätze,
5. der Datei `syntax16.wrd`, die Lexikoneinträge der Form `word(Vollform,Kategorie)` enthält.

Man muß in Prolog nur `:- [syntax16-grammatik].` aufrufen und kann dann mit `?- parse.` oder `?- parsed.` eine Eingabe eintippen. Mit `?- parse(Beispieldatei).` (oder `parsed/1`) kann man eine Datei von Beispielen analysieren lassen. (Der Top-Down-Parser ist langsam, da er sehr viel Doppelarbeit macht; man muß sich in `saetze-meta.pl` auf wenige Komplementrahmen beschränken. Besser wäre, einen Earley-Parser zu verwenden.)

```

<syntax16-grammatik.pl>≡
% Autor: Hans Leiß, LMU, CIS, Syntax natürlicher Sprachen, WS 2016/17

% Aus Metaregeln DCG-Satzregeln erzeugen und in saetze.pl schreiben:
:- [saetze-meta].
:- ersetze_metaregeln.

% DCG-Regeln in Prolog-Klauseln übersetzen, Baumdarstellung
:- ['DCGparser/parser.pl'].

% Grammatiken laden:
:- [nominalphrasen, koordinationen, saetze].

% Startsymbole:
startsymbol(np([_Pers,_Gen,_Num,_Fig],[_Kas])). % einf+koord. NPs
startsymbol(s([_Art,_Fig],[_Temp,_Mod,_Vst])). % einf+koord. Sätze
startsymbol(advp([_Art],[_])). % einf.Adverbiale

% Lexikon laden: die Wörter der Beispiellexika, gesammelt
:- ['syntax16.wrd'].
word(W,Cat) :- syntax16:word(W,Cat).

pron(Art,Form) --> [W], { word(W,pron(Art,Form)) }.
pn(Art,Form) --> [W], { word(W,pn(Art,Form)) }.
cn(Art,Form) --> [W], { word(W,cn(Art,Form)) }.
mn(Art,Form) --> [W], { word(W,mn(Art,Form)) }.
na(Art,Form) --> [W], { word(W,na(Art,Form)) }.
nv(Art,Form) --> [W], { word(W,nv(Art,Form)) }.
n(Art,Form) --> [W], { word(W,n(Art,Form)) }.
art(Art,Form) --> [W], { word(W,art(Art,Form)) }.
qnt(Art,Form) --> [W], { word(W,qnt(Art,Form)) }.
poss(Art,Form) --> [W], { word(W,poss(Art,Form)) }.
a(Art,Form) --> [W], { word(W,a(Art,Form)) }.

```

```

v(Art,Form) --> [W], { word(W,v(Art,Form)) }.
adv(Art,Form) --> [W], { word(W,adv(Art,Form)) }.
prep(Art,Form) --> [W], { word(W,prep(Art,Form)) }.
subj(Art,Form) --> [W], { word(W,subj(Art,Form)) }.
kor(Art,Form) --> [W], { word(W,kor(Art,Form)) }.

```

Die lexikalischen Regeln ausgelagert wurden hier nur deshalb in eine Moduldatei `syntax16.wrd` ausgelagert, um die Grammatik auch mit einem Earley-Parser testen zu können, der die syntaktischen Regeln anders behandelt als das Lexikon.

7.2 Einfache Nominalphrasen, Adjektivphrasen, Adverbiale

Die Kategorie der Nominal- und Adjektivphrasen sowie der Sätze ist hier um das Artmerkmal *Figur* := {*si*, *co*} zur Unterscheidung zwischen *simplen* und *koordinierten* Ausdrücken.

Für die einfachen Nominalphrasen, einfachen Adjektivphrasen und (einfachen) Adverbiale werden die Kategorien

$$NP_{Pers,Gen,Num,Fig}^{Kas}, \quad AP_{Fig}^{Komp,ADekl,Pers,Gen,Num}, \quad AdvP_{Art}$$

benutzt. Die Regeln dazu sind in folgender Datei:

```

⟨nominalphrasen.pl⟩≡
% In mehreren Dateien (koordinationen.pl) definierte Prädikate,
% nach DCG-Übersetzung 5-stellig: cat(Art,Form,-Baum,+I,J):

:- multifile ap/5, np/5, s/5, n/5.

% Lexikalische Regeln: siehe syntax16-grammatik.pl, syntax16.wrd

np([Pers,Gen,Num,si],[Kas]) -->
  ( pron([], [Pers,Gen,Num,Kas])           % er, sie, es
  %   ; pn([Gen],[Kas]), { Pers=3,Num=sg } % Hans, Maria
  %   ; n1([Gen],[Dekl,Num,Kas]),         % alte Menschen
    { Pers=3,(Num=pl;Num=sg),             % ohne Adj nicht
      (Dekl=st ; (Dekl=(-), Kas\=gen)) } % im pl.gen
  %   ; det([Pers,Num,Dekl],[GenD,Kas]), % der alte Hund
    { GenD=(-) ; GenD=Gen },              % ein alter Hund
    n1([Gen],[Dekl1,Num,Kas]),            % ich alter Hund
    { Dekl1=(-) ; Dekl1=Dekl } % ggf. kein Adj in N1
  ).

% N mit Komplement und Attribut: n1([Gen],[Adekl,Num,Kas])

n1([Gen],[-,Num,Kas]) --> % Nomen+Komplement
%   { Dekl = (-) },        % keine Adjektive vorhanden
  n([Kompl,Gen],[Num,Kas]), % ggf. Kompl = []
  ( [] ; np([3,-,si],[gen]) ), % opt. gen.poss/subj
  ( [] % Komplement optional: Hoffnung (auf NP)
  %   ; { Kompl=[gen] }, [] % gen-Objekt statt gen.subj
  %   ; { Kompl=[Praep,KasP], % Angst vor den Hunden
    praeposition(Praep,KasP) },

```

```

    [Praep], np([_Pers,_Gen,_Num,_Fig],[KasP])
; { Kompl=[daß] },
  ( s([daß,-],[Temp,Mod,vl]) % These, daß sie rund ist|sei
  ; s([def,-],[Temp,konj,vz]) % These, sie *ist|sei rund
  )
; { Kompl=[ob] },      % Frage, (ob | wo) Maria lebt
  s([qu,-],[Temp,Mod,vl])
; { Kompl=[infzu] },  % Absicht, zu rauchen
  s([infzu,-],[Temp,-,-])
),
( []      % optionaler Relativsatz: Menschen,
; { GenR=Gen ; GenR=(-), Num=pl }, % an die wir denken
  s([rel(GenR,Num),-],[_,Mod,vl])
).

n1([Gen],[Dekl,Num,Kas]) -->
{ Komp = pos }, % TODO: Komp=komp, sup
ap([Fig],[Komp,Dekl,GenD,Num,Kas]), % alte Hunde, die ...
{ GenD=(-) ; GenD=Gen },
n1([Gen],[-,Num,Kas]).

det([Pers,Num,Dekl],[Gen,Kas]) -->
( ( art([Dekl,Num],[Gen,Kas])          % der, ein
  ; qnt([Dekl,Num],[Gen,Kas])          % kein, jeder
  ; poss([_,-,],[Gen,Num,Kas]),{Dekl=ge} % mein,unser
  ), {Pers=3}
; pron([], [Pers,Gen,Num,Kas]),        % in manchen
  { (Pers=1 ; Pers=2)                  % Formen:
  , (Kas=nom ; Kas=dat ; Kas=akk)
  , Dekl=st }                          % ich armer Student
).

% Zusammenfassung von Nomenarten: n([Kompl,Gen],[Num,Kas])

n([], [Gen],[Num,Kas]) -->
( pn([Gen],[Kas]), {Num=sg} % Peter, Maria
; cn([Gen],[Num,Kas])      % Hund, Menschen
; mn([Gen],[Num,Kas])      % Wein(e), Wasser, Ruhe
).

n([Kompl,Gen],[Num,Kas]) -->
( na([Kompl,Gen],[Num,Kas]) % Gute, Neid auf
; nv([Kompl,Gen],[Num,Kas]) % Hoffnung auf
).

% einfache Adjektivphrasen: ap([si],[Komp,Dekl,Gen,Num,Kas])

ap([si],[pos,Dekl,Gen,Num,Kas]) -->
( [] % Komplemente stets optional
; { Kompl=[Pr,KasP], praeposition(Pr,KasP) },

```

```

    [Pr], np([_,_,_,si],[KasP])
; { Kompl=[Pr,KasP,KasA0], praeposition(Pr,KasP) },
    [Pr], np([_,_,_,si],[KasP]),
    (np([_,_,_,si],[KasA0]) ; [])
    % Linksrekursion durch Stellung vermieden!
),
( [] ; [sehr] ),
a([Kompl],[Dekl,Gen,Num,Kas]). % NP-Regeln der folgenden Folien

% Adverbialphrasen: advp([Art],[ ]) (mit einf+koord.Sätzen)

advp([Art],[ ]) -->
( ([ ] ; ([sehr], { Art=dur })),
  adv([Art],[ ])
; prep([Art,Kas],[ ]),
  np([_,_,_,_],[Kas])
; subj([Art],[ ]), % in metaregelnAdv.pl erzeugt
  s([def,si],[_,ind,vl]) % ohne s.def.vl-Kategorie
).
advp([kond],[ ]) -->
s([def,si],[praes,ind,ve]). % , ( [ ] ; [dann] ).

```

7.3 Koordination von Nominal- und Adjektivphrasen und Sätzen

Die Koordinationsregeln sind so geschrieben, daß jeweils die linke Konstituenten eine *einfache* und der gesamte Ausdruck ein *koordinierter* Ausdruck ist. Dadurch wird insbesondere die Nichtterminierung des Top-Down-Parsers von Prolog verhindert. Allerdings ist diese Einschränkung der Koordination auch allgemein nützlich und sinnvoll, damit die Anzahl der Analysen eines Ausdrucks klein bleibt.

<koordinationen.pl>≡

```
% Koordination von Nominalphrasen und Sätzen HL, 22.1./25.2.2017
```

```
% agreeOrNone(+X,+Y,?Z)
agreeOrNone(X,Y,Z) :-
( agree(X,Y,Z) -> true ; (X \= Y, Z = (-)) ).
agree(X,Y,Z) :- ( X = Y, Z = X ).
```

```
% harmlose (nicht linksrekursive) Koordinationen
np([Per,Gen,pl,co],[Kas]) -->
[sowohl], np([Per1,Gen1,_,si],[Kas]),
[als], [auch], np([Per2,Gen2,_,si],[Kas]),
{ agreeOrNone(Per1,Per2,Per),
  agreeOrNone(Gen1,Gen2,Gen) }.

```

```
np([Per,Gen,pl,co],[Kas]) -->
[weder], np([Per1,Gen1,Num1,si],[Kas]),
```

```

[noch], np([Per2,Gen2,Num2,si],[Kas]),
{ agreeOrNone(Per1,Per2,Per),
  agreeOrNone(Gen1,Gen2,Gen) }.

% Linksrekursion bei Koordinationen (co) ausschließen:

np([Per,-,Num,co],[Kas]) -->      % co = koordiniert
  np([Per1,-,Num1,si],[Kas]),      % linke np: simpel!
  ( [und], np([Per2,-,Num2,si],[Kas]),
    { Num = pl }
  ; [oder], np([Per2,-,Num2,si],[Kas]),
    { agreeOrNone(Num1,Num2,Num) }
  ; % [' ',''] % beim Einlesen gelöscht
    np([Per2,-,Num2,co],[Kas]),
    { Num2 = pl -> Num = pl ; agreeOrNone(Num1,Num2,Num) }
  ),
  { agreeOrNone(Per1,Per2,Per) }.

% Koordination von Sätzen mit Verhinderung der Linksrekursion
s([Def,co],[Temp,Mod,Vst]) -->
  s([Def,si],[Temp1,Mod,Vst]),
  ( ( [und] ; [oder] ),
    s([Def,si],[Temp2,Mod,Vst])
  ; % [' ',''] beim Einlesen gelöscht
    s([Def,co],[Temp2,Mod,Vst])
  ),
  { agreeOrNone(Temp1,Temp2,Temp) }.

% Koordination von Adjektivphrasen mit einfacher AP links
ap([co],[pos,Dekl,Gen,Num,Kas]) -->
  ap([si],[pos,Dekl,Gen,Num,Kas]),
  ( ([und] ; [oder] ; []), % [] statt [' ','']
    ap([si],[pos,Dekl,Gen,Num,Kas])
  ; % [' ',''] beim Einlesen gelöscht
    ap([co],[pos,Dekl,Gen,Num,Kas])
  ).

% NP mit NP.gen als Det, Linksrekursion vermieden:
np([Pers,Gen,Num,si],[Kas]) -->
  { Kas = nom ; Kas = dat ; Kas = akk }, % nicht: gen
  np([Pers,_Gen2,_Num2,si],[gen]), % des Autors grosse Vorliebe
  { Pers=3 },
  n1([Gen],[Dekl,Num,Kas]), % alter Hund
  { Dekl=(-) ; Dekl=st }. % ggf. kein Adj in N1 (st:15.12.16)

```

7.4 Metaregeln für Aussagesätze, Relativsätze und Fragen

Die Metaregeln sind ähnlich der oben beschriebenen Metaregel für einfache Aussagen. Allerdings werden hier Regeln für *einfache* Sätze und Fragen formuliert; eine Regel für einfache Relativsätze und eine für Verbalphrasen (hier als „infinite Sätze“ $S_{infzu, Fig}^{Praes, -, -}$ kategorisiert) sind Modifikationen davon.

Es wird nur *ein* Adverbial außer dem Negationsadverb erlaubt, damit die Anzahl der erzeugten Satzregeln und damit die Parsezeit erträglich bleibt (etwas über 1000 Satzregeln).

```
<saetze-meta.pl>≡
% Metaregeln zur Erzeugung von Satzregeln für einfache Sätze
% aus Komplementrahmen von Verben.                CIS, H.Leiß, 1.3.2017

/* Wenn man nur die Anordnung der Konstituenten sehen will,
   kann man die Expansion der Kategorienabkuerzungen unterdrücken,
   in dem man

       expandiereSubjCat([Pers,Num],Rahmen,SubjRahmen)
       expandiere([Pers,Num],VFIN,VINF,RahmenPraed2,Cats)

so ändert (vgl. jeweils die erste Klausel unten), daß sich

       SubjRahmen = Rahmen
       Cats       = RahmenPraed2

ergibt (was natürlich nicht zu Parsen erlaubt).
*/

:- op(1100,xfx,'-->'). % neues Infix-Relationssymbol

ersetze_metaregeln :- tell('saetze.pl'),
    (A --> B),      % Instanz von (A --> B) beweisen
    portray_clause((A --> B)), % und als DCG-Regel
    nl, fail.      % in 'saetze.pl' schreiben
ersetze_metaregeln :- told. % Datei schliessen.

% Metaregel für einfache Ja/Nein-Fragen und vz/vl-Sätze:

(s([Def,si],[Temp,Mod,Vst]) --> Konstituenten) :-
    ( Vst=ve, Def=qu % Def=def: Konditionalsaeetze, weglassen
    ; Vst=vz, Def=def
    ; Vst=vl, Def=def % Relativsaetze s.u.
    ),
    tempusauswahl(Temp),
    rahmenauswahl(Rahmen), % Auswahl für Teilgrammatik!
    adverbialauswahl(Advs), % Auswahl für Teilgrammatik!
    praedikatsteile(VFIN,VINF, % = Praedikatsteile
                    Rahmen,Pers,Num,Temp,Mod,aktiv),
    expandiereSubjCat([Pers,Num],Rahmen,SubjRahmen),
    felderverteilung(SubjRahmen,Advs,Vst,Kurzkonstituenten),
```

```

    expandiere([Pers,Num],VFIN,VINF,Kurzkonstituenten,Cats),
    listToTuple(Cats,Konstituenten).

% Metaregel für einfache Relativsätze:

% Ersetze in s.def.vl die np am Anfang durch pron.rel mit Gen,Num:
(s([rel(Gen,Num),si],[Temp,Mod,vl]) --> Konstituenten) :-
    (s([def,si],[Temp,Mod,vl]) --> (np([Pers,Gen,Num,_Fig],[Kas]),Rest)),
    Konstituenten = (pron([rel],[Pers,Gen,Num,Kas]),Rest).

% Metaregel für einfache daß-Sätze und untergeordnete Fragesätze:

(s([Def,si],[Temp,Mod,vl]) --> Conj, s([def,si],[Temp,Mod,vl])) :-
    (Def = daß, Conj = ([dass];[daß])
    ; Def = qu, Conj = [ob]
    ).

% Metaregel für einfache zu-Infinitive: (Hack!)

(s([infzu,si],[praes,-,-]) --> Konstituenten) :-
    rahmenauswahl([Subj|Objs]),% Auswahl für Teilgrammatik!
    adverbialauswahl(Adv), % Auswahl für Teilgrammatik!
    praedikatsteile(VFIN,VINF, % = Praedikatsteile
        [Subj|Objs],Pers,Num,fut1,ind,aktiv),
    felderverteilung(Objs,Adv,ve,[vfin|Konst]),
    reverse(Konst,[Inf|KonstRev]),
    reverse(KonstZu,[Inf,[zu]|KonstRev]),
    expandiere([Pers,Num],VFIN,VINF,KonstZu,Cats),
    listToTuple(Cats,Konstituenten).

% Metaregeln, die durch Ersetzen von --> zu DCG-Regeln werden:

% Korrelat für Subjekt- und Objektsätze:
(kor(daß) --> [es]).

% Pronominaladverbiale als Korrelate: (sollten dann aber keine adv([Art],[ ]) sein)
(kor(temp) --> [dann]).
(kor(kaus) --> [deshalb]).
(kor(lok) --> [dort]).
(kor(mod) --> [so]).
(kor(inst) --> [damit]).

% Auswahl von Verbarten, für die Satzregeln erzeugt werden sollen:

rahmenauswahl(Rahmen) :-
    member(Rahmen, % Beispielverben: siehe syntax16.wrd
        [[nom], % intransitive
        [nom,refl(akk)], % reflexive
        [nom,akk], % transitve
    ).

```



```

%      [nom,dat],      % andere zweistellige
%      [nom,auf,dat],  % zweist. mit Präpositionalobjekt
      [nom,auf,akk],    % (Aufg. 10.1 f)
      [nom,daß],      % mit Objektsatz
%      [nom,ob],      % mit interrog.Objektsatz
%      [nom,refl(akk),ob], % reflexiv mit interrog.Objektsatz
      [daß,akk]       % zweistellig mit Subjektsatz
%      [nom,dat,akk]   % dreistellig (Aufgabe 10.1 f)
    ]).

tempusauswahl(Temp) :-
    member(Temp, [praes,perf]). % ohne fut1,praet,plqm,fut2

adverbialauswahl([]).          % keine Adverbiale
adverbialauswahl([advp]).      % ein Adverbial+Korrelat
% adverbialauswahl([advp,advp]). % zwei Adverbiale

% Prädikatsteile und daran abgelesene Merkmale (i.a. von Vst abhängig):
% praedikatsteile/8: belegt VFIN und Liste VINF

praedikatsteile(v([Rahmen], [Pers,Num,Mod,praes]), [[]],
    Rahmen,Pers,Num,praes,Mod,aktiv).
praedikatsteile(v([Rahmen], [Pers,Num,Mod,praet]), [[]],
    Rahmen,Pers,Num,praet,Mod,aktiv).
praedikatsteile(v([haben], [3,Num,Mod,praes]),
    [v([Rahmen], [part2])],
    Rahmen,3,Num,perf,Mod,aktiv).
praedikatsteile(v([haben], [3,Num,Mod,praet]),
    [v([Rahmen], [part2])],
    Rahmen,3,Num,plqm,Mod,aktiv).
praedikatsteile(v([werden], [3,Num,Mod,praes]),
    [v([Rahmen], [inf])],
    Rahmen,3,Num,fut1,Mod,aktiv).
praedikatsteile(v([werden], [3,Num,Mod,praes]),
    [v([Rahmen], [part2]), v([haben], [inf])],
    Rahmen,3,Num,fut2,Mod,aktiv).

% mit Modalverben: (falsch bei vl)

praedikatsteile(v([modal], [Pers,Num,Mod,praes]),
    [v([Rahmen], [inf])],
    Rahmen,Pers,Num,praes,Mod,aktiv).
praedikatsteile(v([modal], [Pers,Num,Mod,praet]),
    [v([Rahmen], [inf])],
    Rahmen,Pers,Num,praet,Mod,aktiv).
praedikatsteile(v([haben], [3,Num,Mod,praes]),
    [v([Rahmen], [inf]), v([modal], [inf])],
    Rahmen,3,Num,perf,Mod,aktiv).

```

```

praedikatsteile(v([haben],[3,Num,Mod,praet]),
    [v([Rahmen],[inf]),v([modal],[inf])],
    Rahmen,3,Num,plqm,Mod,aktiv).
praedikatsteile(v([werden],[3,Num,Mod,praes]),
    [v([Rahmen],[inf]),v([modal],[inf])],
    Rahmen,3,Num,fut1,Mod,aktiv).
praedikatsteile(v([werden],[3,Num,Mod,praes]),
    [v([haben],[inf]),v([Rahmen],[inf]),
    v([modal],[inf])],
    Rahmen,3,Num,fut2,Mod,aktiv).

% Permutationen von Argumenten und Adverbialen:

felderverteilung(Rahmen,Advs,Vst,Kurzkonstituenten) :-
    felder(Rahmen,Advs,Vst,Vor,Mitte,Nach),
    append(Vor,Mitte,Satzanfang),
    verbstellung(Satzanfang,Vst,RahmenPraed),
    append(RahmenPraed,Nach,Kurzkonstituenten).

% felder(+Rahmen,+Vst,-Vorfeld,-Mittelfeld,-Nachfeld)

felder(Rahmen,Advs,Vst,Vor,Mitte,Nach) :-
    praepObjekte(Rahmen,RahmenPraep), % an,akk -> an(akk)
    append(RahmenPraep,Advs,RahmenAdvs),
    permutation(RahmenAdvs,Perm), % i.a. zu viel, teuer!
    nachfeld(Perm,VorMitte,Nach),
    ( Vst=vz -> length(Vor,1) ; length(Vor,0) ),
    append(Vor,Mitte,VorMitte).

praepObjekte([X,Y|Zs],[Pr|PZs]) :-
    ( praeposition(X,Y),
      Pr =.. [X,Y], praepObjekte(Zs,PZs)
    ; praepObjekte([Y|Zs],PZs), Pr=X
    ), !.
praepObjekte([X],[X]).
praepObjekte([],[]).

% daß-Sätze oder Adverbiale evtl.ins Nachfeld, Korrelat im Mittelfeld:
nachfeld([daß|ArgsPraed],[([ ] ;kor(daß))|ArgsPraed],[daß]).
% nachfeld([daß|ArgsPraed],[kor(daß)|ArgsPraed],[ ]).
nachfeld([daß|ArgsPraed],[kor(daß)|Mittelf],Nachf) :-
    !, nachfeld(ArgsPraed,Mittelf,Nachf).
nachfeld([advp|ArgsPraed],[([ ] ;kor(Dim))|ArgsPraed],
    [(advp([Dim],[ ]), {Dim\neg})]).
nachfeld([advp|ArgsPraed],[(advp([Dim],[ ]),{Dim\neg})|Mittelf],Nachf) :-
    !, nachfeld(ArgsPraed,Mittelf,Nachf).

nachfeld([A|ArgsPraed],[A|ArgsPraed2],Nachfeld) :-
    nachfeld(ArgsPraed,ArgsPraed2,Nachfeld).

```

```

nachfeld([], [], []).

% Besetzung der Satzklammern durch Verbteile (mit optionaler Negation):

verbstellung(Args,ve,ArgsPraed) :-
    append([vfin|Args],[neg,vinf],ArgsPraed).
verbstellung([Arg|Args],vz,ArgsPraed) :-
    append([Arg,vfin|Args],[neg,vinf],ArgsPraed).
verbstellung(Args,vl,ArgsPraed) :-
    append(Args,[neg,vinf,vfin],ArgsPraed).

% Expansion von Kategorienabkürzungen:

expandiereSubjCat([_Pers,_Num],Rahmen, Rahmen) :-
    fail, !. % fail := expandieren, true := nicht expandieren

expandiereSubjCat([Pers,Num],[Subj|Objs],[SubjE|Objs]) :-
    expandiereCat(Subj,SubjE),
    kongruenz_sfin(SubjE,[Pers,Num,nom]).

expandiereCat(nom, np([_,_,_,_],[nom])) :- !.
expandiereCat(akk, np([_,_,_,_],[akk])) :- !.
expandiereCat(dat, np([_,_,_,_],[dat])) :- !. % dat-NPs: nur Pron
expandiereCat(daß, s([daß,_,_Form])) :- !.
expandiereCat(ob, s([qu,_,_Form])) :- !.
expandiereCat(neg, ([ ; adv([neg],[ ]])) :- !.
expandiereCat(advp, (advp([Art],[ ]), {Art\=neg})) :- !.
expandiereCat((X;Y),(XE;YE)) :-
    !, expandiereCat(X,XE), expandiereCat(Y,YE).
expandiereCat(PrKas, ([Pr], Cat)) :-
    PrKas =.. [Pr,Kas],
    praeposition(Pr,Kas),
    !, expandiereCat(Kas, Cat).
expandiereCat(X,X) :- !. % (Vereinfachung)

% Präpositionen ohne adverbiale Bedeutung (für Präpositionalobjekte):
praeposition(in,akk). praeposition(an,akk).
praeposition(auf,akk). praeposition(um,akk).
praeposition(für,akk). praeposition(an, dat).
praeposition(mit,dat). praeposition(vor,dat).
praeposition(von,dat). praeposition(auf,dat).

kongruenz_sfin(np([Pers,_Gen,Num,_Fig],[Kas]),[Pers,Num,Kas]).
kongruenz_sfin(s([daß,_Fig],[_,_,v1]),[3,sg,nom]).

% Expansion der Komplementabkürzungen relativ zu den
% Finitheismerkmalen Fin und den Verbteilen Vfin,Vinf:

expandiere(Fin,Vfin,Vinf,[vfin|Args],[Vfin|Kats]) :-

```

```

    !, expandiere(Fin,Vfin,Vinf,Args,Kats).
expandiere(Fin,Vfin,Vinf,[vinf|Args],VinfKats) :-
    !, expandiere(Fin,Vfin,Vinf,Args,Kats),
    append(Vinf,Kats,VinfKats).

expandiere(Fin,Vfin,Vinf,[Arg|Args],[Arg|Kats]) :-
    fail, !, % fail=expandieren, true=nicht expandieren
    expandiere(Fin,Vfin,Vinf,Args,Kats).

expandiere(Fin,Vfin,Vinf,[Arg|Args],[Kat|Kats]) :-
    !, expandiereCat(Fin,Arg,Kat),
    expandiere(Fin,Vfin,Vinf,Args,Kats).
expandiere(_Fin,_Vfin,_Vinf,[],[]).

% expandiereCat/3: Finitheitsmerkmale ins Reflexivpronomen
expandiereCat([Pers,Num], refl(akk),
    pron([refl],[Pers,Num,akk])) :- !.
expandiereCat([_Pers,_Num], X,C) :- expandiereCat(X,C).

% Umwandlung einer Liste in ein Tupel:
listToTuple(X,_) :- var(X), !, fail.
listToTuple([Cat1,Cat2|Cats],[Cat1,Konstituenten]) :-
    listToTuple([Cat2|Cats],Konstituenten).
listToTuple([Cat],Cat).
listToTuple([],[]).

```

7.4.1 Anmerkungen zur Erzeugung von Satzregeln

1. Wenn man bei daß- und untergeordneten Fragesätzen keine s.def.vl-Konstituente haben möchte, kann man folgende Variante der Metaregel für daß-Sätze verwenden:

```

<daß- und Adverbialsätze ohne s.def-Konstituente (unbenutzt)>≡
Alternativ ohne s.def.vl-Konstituente:
(s([Def,si],[Temp,Mod,vl]) --> Conj, Konstituenten) :-
    (s([def,si],[Temp,Mod,vl]) --> Konstituenten)
    ( Def = daß, Conj = ([dass];[daß])
    ; Def = qu, Conj = [ob] ).

```

2. Wenn man nur Adverbiale bestimmter Dimensionen zulassen will, legt man diese fest, z.B.

```

<Adverbiale bestimmter Arten>≡
adverbialdimension(Dim) :-
    member(Dim,[temp,kaus,lok,mod,inst]). % dur

```

und verwendet in den Satzregeln nur gewisse Kombinationen davon. Dazu muß man bei der Auswahl von vorgesehenen Adverbialen, bei der Expansion der Kategorienabkürzungen, und bei der Verschiebung ins Nachfeld ein paar Änderungen vornehmen:

```

<Adverbiale bestimmter Art>≡
adverbialauswahl([Dim]) :-
    !, Dim=mod. % nur modales Adverbial zulassen

```

```

adverbialauswahl([Dim]) :-          % ein Adverbial erlaubter Art
    adverbialdimension(Dim).
adverbialauswahl([lok,temp]).       % zwei bestimmte Adverbiale

<Expansion bestimmter Adverbialarten>≡
    expandiereCat(Dim, advp([Dim], [])) :-
        adverbialdimension(Dim), !.

<Verschiebung bestimmter Adverbiale ins Nachfeld>≡
    nachfeld([Dim|ArgsPraed], [([];kor(Dim))|ArgsPraed], [Dim]) :-
        adverbialdimension(Dim).
    nachfeld([Dim|ArgsPraed], [(kor(Dim);Dim)|Mittelf], Nachf) :-
        adverbialdimension(Dim),
        !, nachfeld(ArgsPraed,Mittelf,Nachf).

```

Bei der Verteilung der Konstituentenabkürzungen auf die Satzfelder könnte man dafür sorgen, daß die Anordnung der Adverbiale im Mittelfeld auf temp < kaus < lok < mod beschränkt wird, indem man das folgende *adverbialordnung/1* auf das Mittelfeld anwendet:

```

<Beschränkung der Anordnung bestimmter Adverbiale>≡
    % Falls man Adverbiale nicht-variabler Dimension einbaut,
    % kann man adverbialordnung(Mitte) auf das Mittelfeld
    % anwenden, damit temp < kaus < lok < mod gilt:

adverbialordnung([temp|Xs]) :-
    !, adverbialordnung(Xs).
adverbialordnung([kaus|Xs]) :-
    !, not(memberchk(temp,Xs)),
    adverbialordnung(Xs).
adverbialordnung([lok|Xs]) :-
    !, not(memberchk(temp,Xs)),
    adverbialordnung(Xs).
adverbialordnung([mod|Xs]) :-
    !, not(memberchk(lok,Xs)),
    not(memberchk(kaus,Xs)),
    not(memberchk(temp,Xs)),
    adverbialordnung(Xs).
adverbialordnung([_X|Xs]) :- % keine Einschränkung an X
    !, adverbialordnung(Xs).
adverbialordnung([]).

```

Die Verwendung konkreter Adverbialarten führt aber zu einer Vervielfältigung der erzeugten Satzregeln, die man sich mit dem Top-Down-Parser von Prolog kaum leisten kann.

3. Wenn man erreichen will, daß Personalpronomen (im Vergleich anderen Nominalphrasen) im Mittelfeld besonderen Anordnungsbedingungen genügen, muß man die Kategorie *NP* aus einer Kategorie *Pron* und einer Kategorie *NP-pro* zusammensetzen und vor der Expansion der Abkürzungen im Mittelfeld eine Aufspaltung von *nom* in die Alternativen *pronom* und *nom-pro* usw. einbauen. Danach testet man, ob die gewählte Anordnung erlaubt ist, grob:

```

<Anordnung von Pronomen im Mittelfeld>≡
    % erlaubte Anordnungen:
    pronordnung([X|Xs]) :-

```

```

( (X = dat-pro ; X = prodat)
-> not(memberchk(proakk,Xs)), % proakk < X.dat
    not(memberchk(kor(daß),Xs)) % es < X.dat (aber: ihm < das möglich)
; (X = proakk ; X = kor(daß))
-> not(memberchk(pronom,Xs)) % pronom < proakk, pronom < kor(daß)
; true ),
pronordnung(Xs).
pronordnung([]).

```

4. Die Bestimmung der Prädikatsteile ist nicht ganz richtig. Erstens erfaßt sie keine Verben mit betontem Präfix, das bei der Verberst- und Verbzweitstellung zum infiniten Prädikatsteil gehört: *er fängt an, fängt er an?, daß er anfängt*, und zweitens ist die Besetzung der rechten Satzklammer bei modalisierten Sätzen in der Verbletzstellung nicht so, wie mit `vinf,vfin` in `verbstellung/3` gesagt wird: statt *daß sie das Buch lesen wollen hatte* heißt es *daß sie das Buch hatte lesen wollen*, im Unterschied zu unmodalisierten Sätzen, wo es *daß sie das Buch gelesen hatte* heißt.

⟨Variante bei den Prädikatsteilen für Sätze mit Modalverben⟩≡

```

praedikatsteile([],
    [v([haben],[3,Num,Mod,praet]),
     v([Rahmen],[inf]),v([modal],[inf])],
    Rahmen,3,Num,plqm,Mod,aktiv).
praedikatsteile([],
    [v([haben],[3,Num,Mod,praes]),
     v([Rahmen],[inf]),v([modal],[inf])],
    Rahmen,3,Num,perf,Mod,aktiv).

```

Möglicherweise gibt es noch weitere Fehler in der Besetzung der Satzklammern.

5. Die Metaregeln mit den oben angegebenen Auswahlen, also 2 Satztempora, mit modalisierten Sätzen, einem 1-stelligen und 5 2-stelligen Verbarten erzeugen 1.210 DCG-Regeln in `saetze.pl`.

Wenn man mehr Tempora oder Komplementrahmen erlaubt, wird das Parsen langsamer.

7.5 Das Lexikon

In `syntax16.wrd` werden etwa 1.500 Vollformen vordefiniert; einige sind auf den Musterlösungen der Hausaufgaben [2] zu finden. Natürlich kann man dieses Lexikon um weitere Einträge anreichern. Bei den Verben gibt es nur wenige Formen im Konjunktiv, damit man bei der Eingabe von Beispielsätzen normalerweise nur die Lesart im Indikativ bekommt.

Literatur

- [1] Gerald Gazdar, Ewan Klein, Geoffrey Pullum and Ivan Sag. Generalized Phrase Structure Grammar. Basil Blackwell, Oxford 1985
- [2] Hans Leiß. Vorlesungsfolien, Aufgabenblätter, Musterlösungen und Programmdateien zum Modul „Syntax natürlicher Sprachen“, WS 2016/17, Ludwig-Maximilians-Universität München, CIS. Siehe <http://www.cis.uni-muenchen.de/~leiss/syntax-16-17/>