Formale Grundlagen der Head Phrase Structure Grammar

Sommersemester 2007
CIS, Universität München

Hans Leiß
leiss@cis.uni-muenchen.de

1 Einleitung: HPSG-Grammatiken

Ein wesentlicher Grundgedanke von *HPSG*-Grammatiken besteht darin, aus den einfachsten Grundeinheiten der Sprache nicht nur syntaktische Baumstrukturen, sondern gleichzeitig semantische, phonologische, diskursrelevante und pragmatische Information von Ausdrücken aufzubauen (Statt von Ausdrücken spricht man in diesem Sinne von sprachlichen Zeichen und sogenannten 'sign-based grammars').

Der Vorteil ist, daß z.B. Syntax, Semantik und Pragmatik von Ausdrücken *simultan* definiert werden kann; da z.B. die Syntax keine Priorität hat, wird die Semantik von Ausdrücken nicht nachträglich rekursiv über ihre syntaktsche Struktur definiert, sondern semantische Eigenschaften der Teilausdrücke können zum Aufbau der syntaktischen Struktur verwendet werden. (Wo wird diese Möglichkeit wirklich genutzt?)

Der Begriff des *sprachlichen Zeichens* wird in HPSG durch getypte Merkmalstrukturen erfaßt. Die Darstellung dieser Strukturen und des Typsystems ist in der Literatur nicht besonders klar. Wir beschreiben zuerst, was Merkmalstrukturen sind und in wie sie in einem Typsystem mit Feldtypen und Subtypbeziehungen getypt werden. Die genaue Wahl des HPSG-Typsystems und der Subtypbeziehungen folgt später; im Grunde erfordert jede Grammatik ein spezifisches Typsystem.

Ein weiterer Grundgedanke der HPSG besteht darin, grammatische Relationen nicht als Äquivalenzrelationen auf Baumstrukturen darzustellen, sondern als gemeinsame Unterstrukturen von Merkmalstrukturen ('structure sharing').

2 Merkmalstrukturen und getypte Merkmalstrukturbeschreibungen

Die Grundbegriffe der Theorie werden von Pollard/Sag 1994 an Beispielen erläutert, aber nicht genau definiert. (Ganz unklar bleibt etwa, ob die Subtypbeziehung strukturell über den Aufbau der Typen definiert ist, oder nur eine Teilrelation davon.) Wir versuchen, passende Definitionen nachzuliefern.

Definition 2.1 Seien $\mathcal{F}$ und $\mathcal{A}$ endliche Mengen. Eine *Merkmalstruktur*

$$\mathcal{M} = (M, \langle F^M \rangle_{F \in \mathcal{F}}, 0, v)$$

über $\mathcal{A}$ und $\mathcal{F}$ besteht aus einer endlichen Trägermenge M, einem ausgezeichneten *Ursprungselement* $0 \in M$, partiellen Funktionen $F^M \subseteq M \times M$ zur Interpretation der *Merkmale* $F \in \mathcal{F}$ und einer partiellen *Bewertung* $v : M \to \mathcal{A}$ so daß für die reflexive transitive Hülle H^* von $H := \bigcup \{ F^M \mid F \in \mathcal{F} \}$ gilt:

(i) $(0, m) \in H^*$ für jedes $m \in M$, und

(ii) $dom(v) = M \setminus dom(H)$.

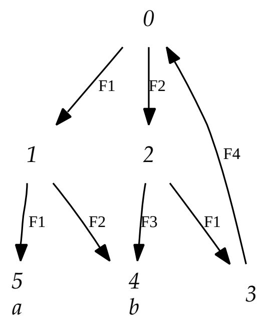
Eine Merkmalstruktur $\mathcal{M}$ ist also ein endlicher Multigraph mit funktionalen Kantenrelationen F^M und einer Markierung seiner *Endpunkte* $m \notin dom(H)$ durch Elemente von $\mathcal{A}$.

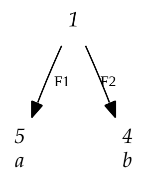
Zu jedem Punkt $m' \in M$ hat $\mathcal{M}$ einen Teilgraphen

$$\mathcal{M}(m') := (M', \langle F^{M'} \rangle_{F \in \mathcal{F}}, m', v'),$$

dessen Trägermenge $M' := \{\, m \in M \mid (m', m) \in H^* \,\}$ aus allen in $\mathcal{M}$ von m' aus erreichbaren Punkten besteht, dessen Ursprungselement m' ist, und dessen Kantenrelationen und Bewertung die Einschränkungen der entsprechenden Operationen von $\mathcal{M}$ sind, d.h. $F^{M'} := F^M \cap M' \times M'$ und $v' := v \cap M' \times \mathcal{A}$.

Beispiel 2.1 Das Bild links zeigt eine Merkmalstruktur M_1 über $\mathcal{F} = \{F1, F2, F3, F4, \ldots\}$ und $\mathcal{A} = \{a, b, c, \ldots\}$: Die Nummern sind die Punkte $m \in M$, die Kanten mit gleicher Beschriftung F die Relationen F^M, die Beschriftungen an den Blättern m die Werte $v(m)$, und die Wurzel ist 0.





Das Bild rechts zeigt die Teilstruktur zum Punkt 1.

Definition 2.2 Ein *Homomorphismus* zwischen Merkmalstrukturen $\mathcal{M}$ und $\mathcal{N}$ über $\mathcal{F}$ und $\mathcal{A}$ ist eine Abbildung $h : M \to N$, sodaß für alle $m, k \in M$, $F \in \mathcal{F}$, $a \in \mathcal{A}$,

(i) $h(0^M) = 0^N$,

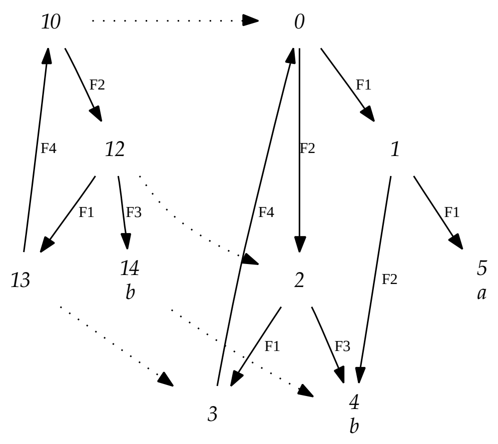
(ii) ist $F^M(m, k)$ so ist auch $F^N(h(m), h(k))$, und

(iii) ist $m \in dom(v^M)$, so ist $h(m) \in dom(v^N)$ und $v^N(h(m)) = v^M(m)$.

Zwei Merkmalstrukturen $\mathcal{M}$ und $\mathcal{N}$ über $\mathcal{F}$ und $\mathcal{A}$ sind *isomorph*, wenn es eine Bijektion $h : M \to N$ gibt, sodaß $h : M \to N$ und $h^{-1} : N \to M$ Homomorphismen von $\mathcal{M}$ nach $\mathcal{N}$ bzw. von $\mathcal{N}$ nach $\mathcal{M}$ sind; dafür schreiben wir $h : \mathcal{M} \xrightarrow{\sim} \mathcal{N}$. Wir betrachten isomorphe Merkmalstrukturen als gleich, d.h. ignorieren die Wahl der Punktmenge M.

Betrachtet man die Werte $a \in \mathcal{A}$ als Prädikate $a^M := \{\, m \in M \mid v^M(m) = a \,\}$ auf $\mathcal{M}$, so werden durch einen Homomorphismus $h : \mathcal{M} \to \mathcal{N}$ die in $\mathcal{M}$ wahren atomaren Aussagen $m = 0$, $F(m, k)$ und $a(m)$ über m und k zu in $\mathcal{N}$ wahren Aussagen über die Bildpunkte $h(m)$ und $h(k)$.

Beispiel 2.2 Eine zweite Merkmalstruktur M_2 (mit Wurzel 10) und ein Homomorphismus $h : M_2 \to M_1$:

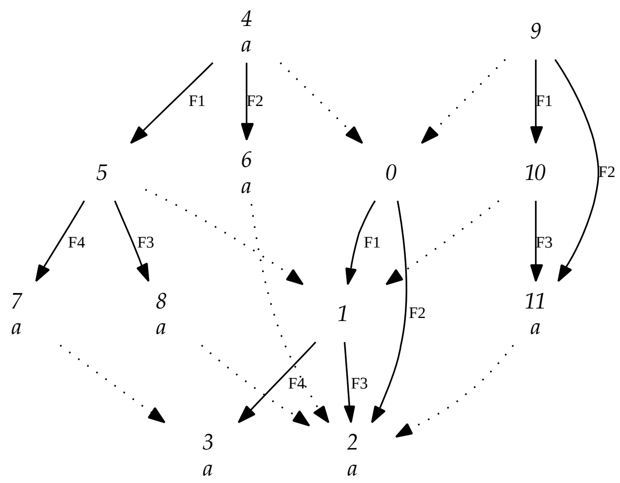


Definition 2.3 Eine Merkmalstruktur $\mathcal{N}$ ist *mindestens so fein wie* $\mathcal{M}$, in Zeichen: $\mathcal{N} \sqsubseteq \mathcal{M}$, wenn es einen Homomorphismus $h : \mathcal{M} \to \mathcal{N}$ gibt, in Zeichen: $h : \mathcal{M} \overset{\sqsupseteq}{\to} \mathcal{N}$. Die Merkmalstrukturen $\mathcal{M}$ und $\mathcal{N}$ sind *unifizierbar*, wenn es eine *allgemeinste gemeinsame Verfeinerung* oder *Unifikation* von $\mathcal{M}$ und $\mathcal{N}$ gibt, d.h. eine Merkmalstruktur $\mathcal{M} \sqcap \mathcal{N}$ mit

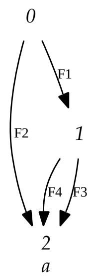
(i) Verfeinerungen $h : \mathcal{M} \overset{\sqsupseteq}{\to} \mathcal{M} \sqcap \mathcal{N}$ und $j : \mathcal{N} \overset{\sqsupseteq}{\to} \mathcal{M} \sqcap \mathcal{N}$, und

(ii) für jede gemeinsame Verfeinerung $h' : \mathcal{M} \overset{\sqsupseteq}{\to} \mathcal{K}$ und $j' : \mathcal{N} \overset{\sqsupseteq}{\to} \mathcal{K}$ von $\mathcal{M}$ und $\mathcal{N}$ gibt es eine Verfeinerung $k : \mathcal{M} \sqcap \mathcal{N} \overset{\sqsupseteq}{\to} \mathcal{K}$, so daß $h' = k \circ h$ und $j' = k \circ j$.

Beispiel 2.3 Merkmalstrukturen $\mathcal{M}$ mit den Punkten 4,5,6,7,8, $\mathcal{N}$ mit den Punkten 9,10,11, und ihre allgemeinste gemeinsame Verfeinerung $\mathcal{M} \sqcap \mathcal{N}$ mit den Punkten 0,1,2 und den entsprechenden Homomorphismen $h : \mathcal{M} \overset{\sqsupseteq}{\to} \mathcal{M} \sqcap \mathcal{N}$ und $j : \mathcal{N} \overset{\sqsupseteq}{\to} \mathcal{M} \sqcap \mathcal{N}$:



Eine weitere Verfeinerung $k : \mathcal{M} \sqcap \mathcal{N} \overset{\sqsupseteq}{\to} \mathcal{K}$ erhält man mit $k(0) = 0, k(1) = 1$, $k(2) = k(3) = 2$:



Textdarstellung einer Merkmalstruktur: Im Text wird eine Merkmalstruktur so dargestellt, daß man für die Endpunkte nur ihre Werte und für die Verzweigungspunkte eine Klammerstruktur schreibt, wobei die einzelnen Zweige als Gleichungen $F = M'$ mit der Textdarstellung M' der Teilstruktur dargestellt werden, auf die die F-Kante zeigt. Falls zwei Kanten auf denselben Punkt m zeigen, wird dies mit eine Markierung $\boxed{\text{m}}$ vor dem Wert bei m oder der Textdarstellung der bei m beginnenden Teilstruktur angedeutet.

Beispiel 2.4 Die Merkmalstruktur M_1 von Beispiel 2.1 wird im Text also wie folgt dargestellt:

$$\boxed{0}\begin{bmatrix} \text{F1} = \begin{bmatrix} \text{F1} = a \\ \text{F2} = \boxed{4} b \end{bmatrix} \\ \text{F2} = \begin{bmatrix} \text{F3} = \boxed{4} b \\ \text{F1} = \begin{bmatrix} \text{F4} = \boxed{0} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Man kann durch Merkmalstrukturen viele andere Objekte definieren, z.B. geordnete Paare und Listen:

Ein *geordnetes Paar* $\langle M_1, M_2 \rangle$ von Merkmalstrukturen M_i kann man als eine Merkmalstruktur

$$\begin{bmatrix} 1 = M_1 \\ 2 = M_2 \end{bmatrix}$$

verstehen, wobei $1, 2$ besondere Merkmale aus $\mathcal{F}$ sind. (Bei $\langle M, M \rangle$ ist gemeint, daß in der zweiten Komponente eine isomorphe Kopie der Merkmalstruktur M der ersten Komponente verwendet wird!)

Eine *Liste* $\langle M_1, \ldots, M_n \rangle$ von n Merkmalstrukturen M_i ist entweder (bei $n = 0$) die leere Liste $\langle\rangle$, die man als Merkmalstruktur $(M, \langle F^M \rangle_{F \in \mathcal{F}}, 0, v)$ mit $M = \{0\}$ $F^M = \emptyset$ und $v(0) = nil \in \mathcal{A}$ verstehen kann, oder (bei $n > 0$) das geordnete Paar $\langle M_1, \langle M_2 \ldots, M_n \rangle \rangle$, dessen erste Komponente die Merkmalstruktur M_1 und dessen zweite Komponente die (kürzere, also schon definierte) Liste $\langle M_2, \ldots, M_n \rangle$ von Merkmalstrukturen ist. Die Komponenten bezeichnet man oft mit Kopf und Rumpf:

$$\langle M_1, M_2, \ldots, M_n \rangle \simeq \begin{bmatrix} \text{HEAD} = M_1 \\ \text{TAIL} = \langle M_2, \ldots, M_n \rangle \end{bmatrix}$$

2.1 Typen

Zwar sollen die Merkmalstrukturen die Objekte sein, über die man in einer HPSG-Grammatik spricht, (nämlich strukturierte sprachliche Zeichen), aber in der Grammatik treten nur *Beschreibungen von* Merkmalstrukturen auf.

Eine Merkmalstrukturbeschreibung drückt nur Teilaspekte der beschriebenen Struktur aus; durch das Ergänzen weiterer Aspekte soll jede Beschreibung „vervollständigt“ werden können. Durch eine *vollständige Merkmalstrukturbeschreibung* soll (bis auf Isomorphie) nur *eine* Merkmalstruktur beschrieben werden.

Um diese Vorstellung zu präzisieren, führt man *Typen* (von Merkmalbeschreibungen) und eine partielle Ordnung $\sqsubseteq$ zwischen ihnen, eine *Typhierarchie* ein, die die Verfeinerung der Beschreibungen definiert. Die Beschreibungen von feinstem Typ sind die vollständigen Beschreibungen.

Jede HPSG-Grammatik definiert eine Reihe von Typen und eine Typhierarchie, bevor sie Aussagen mit Hilfe getypter Merkmalbeschreibungen macht; die Idee ist, daß Aussagen mit groberen Beschreibungen umfassender sind als die mit feineren Beschreibungen. Z.B. will man allgemeine Aussagen über alle Verben machen können und spezifischere Aussagen über Verben, die bestimmte Argumente fordern; für diese sollen aber die allgemeinen Aussagen auch gelten.

2.1.1 Typdefinitionen

Definition 2.4 Die *Typen (über* Σ *und* $\mathcal{F}$*)* seien induktiv definiert durch

(i) jede Typkonstante $\sigma \in \Sigma$ ist ein (Basis-)Typ,

(ii) jede Typvariable α einer unendlichen Liste *TyVar* ist ein (Basis-)Typ,

(iii) sind $\sigma_1, \dots, \sigma_n$ Typen und $F_1, \dots, F_n \in \mathcal{F}$ verschieden, so ist $\{F_1 : \sigma_1, \dots, F_n : \sigma_n\}$ ein Typ,

(iv) sind $\sigma_1, \dots, \sigma_n$ Typen, so ist $\bigcup\{\sigma_1, \dots, \sigma_n\}$ (auch $\sigma_1 + \ldots + \sigma_n$ geschrieben) ein Typ,

(v) sind $\alpha_1, \dots, \alpha_n$ verschiedene Typvariable und $\sigma_1, \dots, \sigma_n$ Typen (in denen $\alpha_1, \dots, \alpha_n$ vorkommen dürfen), so ist auch $\mu_i(\alpha_1, \dots, \alpha_n).(\sigma_1, \dots, \sigma_n)$ ein Typ, für jedes $i = 1, \dots, n$.

Bei den Summen $\bigcup\{\sigma_1, \dots, \sigma_n\}$ sei stets $n \geq 1$, und $\bigcup\{\sigma_1\}$ sei dasselbe wie σ_1.

Die letzte Konstruktionsmöglichkeit dient dazu, simultan rekursive Typen einzuführen. Dabei stellt man ein Gleichungssystem

$$\begin{array}{ccc} \alpha_1 & = & \sigma_1(\alpha_1, \dots, \alpha_n), \\ \vdots & \vdots & \vdots \\ \alpha_n & = & \sigma_n(\alpha_1, \dots, \alpha_n), \end{array}$$

mit variablen Typen α_i auf, dessen ausgezeichnete Lösung[1] mit $\mu(\alpha_1, \dots, \alpha_n).(\sigma_1, \dots, \sigma_n)$ bezeichnet wird; $\mu_i(\alpha_1, \dots, \alpha_n).(\sigma_1, \dots, \sigma_n)$ ist die i-te Komponente hiervon.

Bemerkung 2.5 Simultan rekursiv sind in HPSG (nach Pollard/Sag) beispielsweise die Typen *synsem*,*category* und *local* definiert, die man mit geeigneten Typen $\sigma_i(\alpha_1, \alpha_2, \alpha_3)$ durch

$$\begin{array}{rcl} \mathit{synsym} & := & \mu_1(\alpha_1, \alpha_2, \alpha_3).\ (\sigma_1, \sigma_2, \sigma_3) \\ \mathit{category} & := & \mu_2(\alpha_1, \alpha_2, \alpha_3).\ (\sigma_1, \sigma_2, \sigma_3) \\ \mathit{local} & := & \mu_3(\alpha_1, \alpha_2, \alpha_3).\ (\sigma_1, \sigma_2, \sigma_3) \end{array}$$

definiert. Dafür schreibe ich kürzer

$$(\mathit{synsym}, \mathit{category}, \mathit{local}) := \mu(\alpha_1, \alpha_2, \alpha_3).(\sigma_1, \sigma_2, \sigma_3).$$

Ein einfacher rekursiver Typ ist der Typ σ *list* der Listen von Objekten des Typs σ, den man sich rekursiv durch die Gleichung

$$\sigma\ \mathit{list} = \mathit{nil} + \{\mathit{first} : \sigma,\ \mathit{rest} : \sigma\ \mathit{list}\}$$

[1] wie man die Existenz der Lösung sichert und die Lösung konstruiert, lassen wir hier offen

definiert vorstellt; Formal ist er eine Abkürzung

$$\sigma\ \mathit{list} := \mu\alpha\ (\mathit{nil} + \{\mathit{first} : \sigma,\ \mathit{rest} : \alpha\}).$$

Die Semantik von Typen ist so zu definieren, daß $\sigma\ \mathit{list}$ den kleinsten Bereich α von Objekten (d.h. Merkmalstrukturen) bezeichnet, für den

$$\alpha = \mathit{nil} \cup \{\mathit{first} : \sigma,\ \mathit{rest} : \alpha\ \mathit{list}\}$$

gilt, wenn σ den Bereich der Objekte vom Typ σ und nil den Bereich der Merkmalstrukturen mit $v(0) = \mathit{nil} \in \mathcal{A}$ (die alle die leere Liste $\langle\rangle$ darstellen, s.o.) ist.

2.1.2 Getypte Merkmalstrukturbeschreibungen

Definition 2.5 Eine *getypte Merkmalstrukturbeschreibung* $(\mathcal{M}, \mathit{type})$ über $\mathcal{F}$ und einer Menge T von Typen ist eine Merkmalstruktur

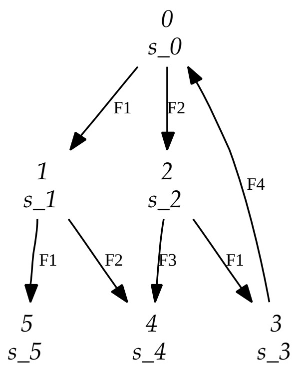
$$\mathcal{M} = (M, \langle F^{\mathcal{M}} \rangle_{F \in \mathcal{F}}, 0)$$

über $\mathcal{F}$ (ohne Bewertung der Endpunkte) mit einer verträglichen Annotierung $\mathit{type} : M \to T$ der Punkte durch Typen. Die Annotierung ist *verträglich*, wenn gilt: ist $m \in M$ ein Verzweigungspunkt mit n ausgehenden Kanten $m \xrightarrow{F_i} m_i$, $1 \leq i \leq n$, so ist der Typ

$$\mathit{type}(m) = \{F_1 : \mathit{type}(m_1), \ldots, F_n : \mathit{type}(m_n), \ldots\}.$$

Wir fordern nicht, daß die Endknoten mit atomaren Typen annotiert werden oder daß in $\mathit{type}(m)$ nur die im Graphen bei m gezeigten Kanten ausgehen. Die Typen geben an, Kanten mit welchen Merkmalen noch ergänzt werden müssen, damit eine Merkmalstruktur vom entsprechenden Typ entsteht.

Beispiel 2.6 Sind $\sigma_0, \ldots, \sigma_5$ Typen aus T, so ist (mit *s_i* statt σ_i) der folgende annotierte Graph



eine getypte Merkmalbeschreibung, falls gilt:

$$\begin{aligned}
\sigma_0 &= \{F_1 : \sigma_1,\ F_2 : \sigma_2,\ \ldots\},\\
\sigma_1 &= \{F_1 : \sigma_5,\ F_2 : \sigma_4,\ \ldots\},\\
\sigma_2 &= \{F_3 : \sigma_4,\ F_1 : \sigma_3,\ \ldots\},\\
\sigma_3 &= \{F_4 : \sigma_0,\ \ldots\}
\end{aligned}$$

Da es für σ_4 und σ_5 keine Bedingungen gibt, kann man diese frei wählen, aber die übrigen σ_i hängen von den gewählten σ_4, σ_5 ab. Beachte, daß in σ_0 als Typ des Felds eines Felds eines Felds wieder σ_0 auftritt; daher braucht man die Möglichkeit, rekursive Typen zu definieren.

Bem. Man muß noch genauer sagen, wie die Annotierung mit einem rekursiven Typ gemeint ist. Ein mit σ *list* markierer Knoten soll wegen $\sigma\ list = nil + \{first : \sigma, rest : \sigma\ list\}$ ein Endpunkt mit der Markierung *nil* oder ein Verzweigungspunkt sein können, dessen zwei Nachfolger die Markierungen σ bzw. σ *list* haben. (Und wie ist die Annotierung mit Summentypen gemeint?)

Textdarstellung einer getypten Merkmalbeschreibung Im Text stellen wir eine getypte Merkmalstrukturbeschreibung wie eine Merkmalstruktur dar, außer daß Typen als unterer Index der schließenden Feldklammern angegeben werden, und bei den Endpunkten Typen an die Stelle der Werte geschrieben werden.

Die Textdarstellung der Merkmalstrukturbeschreibung des Beispiels lautet also:

$$\boxed{0}\begin{bmatrix} \mathrm{F1} = \begin{bmatrix} \mathrm{F1} = \sigma_5 \\ \mathrm{F2} = \boxed{4}\sigma_4 \end{bmatrix}_{\sigma_1} \\ \mathrm{F2} = \begin{bmatrix} \mathrm{F3} = \boxed{4} \\ \mathrm{F1} = \begin{bmatrix} \mathrm{F4} = \boxed{0} \end{bmatrix}_{\sigma_3} \end{bmatrix}_{\sigma_2} \end{bmatrix}_{\sigma_0}$$

Da sich bei einer verträglichen Typannotierung die Typen an den inneren Punkten aus dem Typ an der Wurzel ergeben, wird oft nur der Typ an der Wurzel gezeigt.

2.1.3 Subtypen und Typhierarchie

Durch die Definition einer Menge T von Typen ergibt sich eine Verfeinerungsrelation $\sigma \sqsubseteq \tau$ zwischen Typen und daraus später eine Verfeinerung zwischen getypten Merkmalstrukturbeschreibungen.

Im allgemeinen hängt $\sigma \sqsubseteq \tau$ von einer Menge Γ von Annahmen $\alpha \sqsubseteq \rho$ oder $\rho \sqsubseteq \alpha$ für freie Typvariable α von $\sigma + \tau$ ab. Wir definieren durch Induktion über den Aufbau der Typen σ und τ die Bedeutung der Aussage „unter den Annahmen Γ gilt $\sigma \sqsubseteq \tau$“, kurz: $\Gamma \triangleright \sigma \sqsubseteq \tau$:

$$\frac{}{\Gamma \triangleright \sigma \sqsubseteq \sigma} \quad (\mathit{refl})$$

$$\frac{\Gamma \triangleright \sigma \sqsubseteq \rho \qquad \Gamma \triangleright \rho \sqsubseteq \tau}{\Gamma \triangleright \sigma \sqsubseteq \tau} \quad (\mathit{trans})$$

$$\frac{\text{Für jedes } \sigma \in S \text{ gibt es } \tau \in T \text{ mit } \Gamma \triangleright \sigma \sqsubseteq \tau}{\Gamma \triangleright \bigcup S \sqsubseteq \bigcup T} \quad (\mathit{sum})$$

$$\frac{\Gamma \triangleright \sigma_i \sqsubseteq \tau_i \text{ für alle } i \in I \subseteq J}{\Gamma \triangleright \{\, F_i : \sigma_i \mid i \in J \,\} \sqsubseteq \{\, F_i : \tau_i \mid i \in I \,\}} \quad (\mathit{record})$$

$$\frac{\Gamma \triangleright \sigma \sqsubseteq \tau_i[\alpha_1/\mu_1\vec{\alpha}.\vec{\tau}, \ldots, \alpha_n/\mu_n\vec{\alpha}.\vec{\tau}]}{\Gamma \triangleright \sigma \sqsubseteq \mu_i\vec{\alpha}.\vec{\tau}} \quad (\sqsubseteq \mu_i)$$

$$\frac{\Gamma \cup \{\alpha_1 \sqsubseteq \sigma_1, \ldots, \alpha_n \sqsubseteq \sigma_n\} \triangleright \tau_j(\alpha_1, \ldots, \alpha_n) \sqsubseteq \sigma_j, \quad j = 1, \ldots, n}{\Gamma \triangleright \mu_i(\alpha_1, \ldots, \alpha_n).(\tau_1, \ldots, \tau_n) \sqsubseteq \sigma_i} \quad (\mu_i \sqsubseteq)$$

Das besagt im wesentlichen,

(i) daß $\sqsubseteq$ reflexiv und transitiv ist,

(ii) daß ein Summentyp feiner als ein zweiter ist, wenn er nur feinere Summanden als der zweite hat,

(iii) daß Feldtypen mit mehr oder feiner getypten Feldern feiner sind als solche mit weniger oder grober getypten Feldern,

(iv) daß ein Typ feiner als die i-te Komponente eines rekursiven Typs ist, wenn er feiner als der definierende Typ der i-ten Komponente ist, in dem die Rekurionsvariablen durch die jeweiligen rekursiven Typen ersetzt sind, und

(v) daß ein rekursiv definierter Typ feiner ist als ein vorgegebener Typ, wenn *unter der Zusatzannahme* die Rekursionsvariablen seien feiner als vorgegebene Typen, die definierenden Typen der rekursiven Definition feiner sind als die vorgegebenen.

Wie gesagt, kann man Paartypen

$$(\sigma \times \tau) := \left\{ \begin{array}{l} 1 : \sigma \\ 2 : \tau \end{array} \right\}$$

und damit einen Typ der Listen von Objekten vom Typ σ definieren,

$$\sigma\ \mathit{list} := \mu\alpha.(\mathit{nil} + (\sigma \times \alpha))$$

Mit den obigen Subtypisierungsregeln kann man für diese folgende Subtypisierungsregeln herleiten:

$$\frac{\Gamma \triangleright \sigma_1 \sqsubseteq \tau_1 \qquad \Gamma \triangleright \sigma_2 \sqsubseteq \tau_2}{\Gamma \triangleright \sigma_1 \times \sigma_2 \sqsubseteq \tau_1 \times \tau_2} \quad (\mathit{pair})$$

$$\frac{\Gamma \triangleright \sigma \sqsubseteq \tau}{\Gamma \triangleright \sigma\ \mathit{list} \sqsubseteq \tau\ \mathit{list}} \quad (\mathit{list})$$

Die Regel (*pair*) ist ein Spezialfall von (*record*), während man für (*list*) die Regeln für μ anwenden muß.

Beispiel 2.7 Seien *mask* und *fem* zwei Grundtypen, und damit $\mathit{Genus} := \mathit{fem} + \mathit{mask}$ definiert. Nach (*sum*) ist dann $\triangleright \mathit{fem} \sqsubseteq \mathit{Genus}$ und $\triangleright \mathit{mask} \sqsubseteq \mathit{Genus}$. Daher wird mit (*record*)

$$\triangleright \left\{ \begin{array}{ll} \textsc{Name} & : \mathit{String} \\ \textsc{Geburt} & : \mathit{Datum} \\ \textsc{Geschlecht} & : \mathit{mask} \end{array} \right\} \sqsubseteq \left\{ \begin{array}{ll} \textsc{Name} & : \mathit{String} \\ \textsc{Geschlecht} & : \mathit{Genus} \end{array} \right\}$$

Nun sei (einfach) rekursiv der Typ *person* durch

$$\mathit{person} := \mu\gamma.\pi := \mu\gamma. \left\{ \begin{array}{ll} \textsc{Name} & : \mathit{String} \\ \textsc{Geburt} & : \mathit{Datum} \\ \textsc{Eltern} & : \gamma \times \gamma \end{array} \right\}$$

definiert, weshalb wir die Gleichung

$$\mathit{person} = \left\{ \begin{array}{ll} \textsc{Name} & : \mathit{String} \\ \textsc{Geburt} & : \mathit{Datum} \\ \textsc{Eltern} & : \mathit{person} \times \mathit{person} \end{array} \right\} = \pi[\gamma/\mathit{person}]$$

benutzen können.

Weiter seien simultan rekursiv die beiden folgenden Typen definiert:

$$(\mathit{frau}, \mathit{mann}) := \mu(\alpha, \beta).(\tau_1, \tau_2)$$

mit

$$\tau_1(\alpha, \beta) := \left\{ \begin{array}{ll} \textsc{Name} & : \mathit{String} \\ \textsc{Geburt} & : \mathit{Datum} \\ \textsc{Eltern} & : \alpha \times \beta \\ \textsc{Geschlecht} & : \mathit{fem} \end{array} \right\} \qquad \tau_2(\alpha, \beta) := \left\{ \begin{array}{ll} \textsc{Name} & : \mathit{String} \\ \textsc{Geburt} & : \mathit{Datum} \\ \textsc{Eltern} & : \alpha \times \beta \\ \textsc{Geschlecht} & : \mathit{mask} \end{array} \right\}$$

Wir wollen zeigen, daß $\mathit{frau} \sqsubseteq \mathit{person}$ und $\mathit{mann} \sqsubseteq \mathit{person}$ auf Grund der Definitionen.

Um $\mathit{frau} \sqsubseteq \mathit{person}$ zu zeigen, genügt es nach ($\mu_i \sqsubseteq$) zu zeigen:

Aus den Zusatzannahmen $\alpha \sqsubseteq \mathit{person}$ und $\beta \sqsubseteq \mathit{person}$ kann man zeigen, daß

$$\tau_1(\alpha,\beta) \sqsubseteq \mathit{person} \quad \text{und} \quad \tau_2(\alpha,\beta) \sqsubseteq \mathit{person}.$$

Das besagt gerade der Spezialfall der Regel (mit $\sigma_1 = \sigma_2 = \mathit{person}$)

$$\frac{\{\alpha \sqsubseteq \sigma_1, \beta \sqsubseteq \sigma_2\} \rhd \tau_j(\alpha,\beta) \sqsubseteq \sigma_j, \quad j = 1,2}{\Gamma \rhd \mu_i(\alpha,\beta)(\tau_1,\tau_2) \sqsubseteq \sigma_i} \quad (\mu_i \sqsubseteq)$$

wegen $\mathit{frau} := \mu_1(\alpha,\beta).(\tau_1,\tau_2)$. Um nun die Prämissen hiervon zu zeigen, also

$$\{\alpha \sqsubseteq \mathit{person}, \beta \sqsubseteq \mathit{person}\} \rhd \tau_j(\alpha,\beta) \sqsubseteq \mathit{person},$$

genügt es nach $(\sqsubseteq \mu_i)$

$$\{\alpha \sqsubseteq \mathit{person}, \beta \sqsubseteq \mathit{person}\} \rhd \tau_j(\alpha,\beta) \sqsubseteq \pi[\gamma/\mathit{person}],$$

zu zeigen, und das heißt für $j = 1$

$$\{\alpha \sqsubseteq \mathit{person}, \beta \sqsubseteq \mathit{person}\} \rhd \left\{\begin{array}{ll} \textsc{Name} & : \mathit{String} \\ \textsc{Geburt} & : \mathit{Datum} \\ \textsc{Eltern} & : \alpha \times \beta \\ \textsc{Geschlecht} & : \mathit{fem} \end{array}\right\} \sqsubseteq \left\{\begin{array}{ll} \textsc{Name} & : \mathit{String} \\ \textsc{Geburt} & : \mathit{Datum} \\ \textsc{Eltern} & : \mathit{person} \times \mathit{person} \end{array}\right\}. \tag{1}$$

Nun können wir mit den Annahmen gerade die Komponenten der Paartypen $\alpha \times \beta$ und $\mathit{person} \times \mathit{person}$ vergleichen, weshalb nach der Regel (*pair*) folgt:

$$\{\alpha \sqsubseteq \mathit{person}, \beta \sqsubseteq \mathit{person}\} \rhd \alpha \times \beta \sqsubseteq \mathit{person} \times \mathit{person}.$$

Da die Typen in den anderen gemeinsamen Feldern von τ_1 und $\pi[\gamma/\mathit{person}]$ gleich sind und der zweite Feldtyp weniger Felder als τ_1 hat, folgt (1) mit der Regel (*record*). Analog kann man aus den Zusatzannahmen auch $\tau_2(\alpha,\beta) \sqsubseteq \mathit{person}$ zeigen, und damit ist insgesamt die Behauptung $\mathit{frau} \sqsubseteq \mathit{person}$ gezeigt.

Beachte, daß der Subtyp $\mathit{frau} \sqsubseteq \mathit{person}$ nicht nur die Zusatzinformation über das Geschlecht enthält, sondern auch einen genaueren Typ der Eltern, nämlich $\mathit{frau} \times \mathit{mann} \sqsubseteq \mathit{person} \times \mathit{person}$.

Man verallgemeinert die durch eine Menge T definierter Typen gegebene Subtypbeziehung $\sqsubseteq$ auf Merkmalstrukturbeschreibungen:

Definition 2.6 Sind $\mathcal{M} = (M, F^M, 0^M, \mathit{type}^M)$ und $\mathcal{N} = (N, F^N, 0^N, \mathit{type}^N)$ getypte Merkmalbeschreibungen über T, so ist $\mathcal{M}$ *mindestens so fein wie* $\mathcal{N}$, kurz $\mathcal{M} \sqsubseteq \mathcal{N}$, wenn $\rhd \mathit{type}^M(0^M) \sqsubseteq \mathit{type}^N(0^N)$.

Beispiel 2.8 Für Typen $\tau_0 = \left\{\begin{array}{l} \mathrm{F1} : \tau_1 \\ \mathrm{F2} : \tau_2 \end{array}\right\}$, $\tau_1 = \{\mathrm{F2} : \tau_4\}$ und $\tau_2 = \left\{\begin{array}{l} \mathrm{F1} : \tau_3 \\ \mathrm{F3} : \tau_4 \end{array}\right\}$ ist $\sigma_i \sqsubseteq \tau_i$ und damit

$$\boxed{0}\left[\begin{array}{l} \mathrm{F1} = \left[\begin{array}{l} \mathrm{F1} = \sigma_5 \\ \mathrm{F2} = \boxed{4}\sigma_4 \end{array}\right]_{\sigma_1} \\ \mathrm{F2} = \left[\begin{array}{l} \mathrm{F3} = \boxed{4} \\ \mathrm{F1} = \left[\mathrm{F4} = \boxed{0}\right]_{\sigma_3} \end{array}\right]_{\sigma_2} \end{array}\right]_{\sigma_0} \quad \sqsubseteq \quad \left[\begin{array}{l} \mathrm{F1} = \left[\mathrm{F2} = \boxed{4}\tau_4\right]_{\tau_1} \\ \mathrm{F2} = \left[\begin{array}{l} \mathrm{F3} = \boxed{4} \\ \mathrm{F1} = [\,]_{\tau_3} \end{array}\right]_{\tau_2} \end{array}\right]_{\tau_0}$$

eine feinere und eine grobere Merkmalstrukturbeschreibung der Merkmalstruktur aus Beispiel 2.1.

Die Subtypisierung wird in der Literatur zu HPSG etwas anders verstanden als in der Informatik: man möchte eine getypte Merkmalstrukturbeschreibung immer weiter verfeinern können, bis man eine Merkmalstruktur bekommt. Daher werden die Werte $a \in \mathcal{A}$ auch als atomare Typen verstanden, und größere Wertmengen durch Summen $a_1 + \ldots + a_n$ aufgebaut.

Definition 2.7 Die Menge $\mathcal{A}$ der atomaren Werte sei entsprechend den atomaren Typen τ in disjunkte Teilmengen $\mathcal{A}_\tau \subseteq \mathcal{A}$ zerlegt. Eine Merkmalstrukturbeschreibung $\mathcal{N} = (N, F^N, 0^N, type^N)$ *beschreibt* die Merkmalstruktur $\mathcal{M} = (M, F^M, 0^M, v^M)$, falls es eine Typannotation $type^M : M \to T$ gibt, so daß $\mathcal{M}' \sqsubseteq \mathcal{N}$ für $\mathcal{M}' = (M, F^M, 0^M, type^M)$ und für jeden Endpunkt $m \in M$ der Typ $type(m)$ atomar und $v(m) \in \mathcal{A}_{type(m)}$ ist.

Beispiel 2.9 Aus den Grundtypen *nom*, *gen*, *dat*, *akk* und *fem*, *mask* seien die Typen

$$Genus := fem + mask, \qquad Kasus := nom + gen + dat + akk$$

definiert (mit jeweils nur einem Objekt, also $\mathcal{A}_{nom} = \{nom\}, \ldots, \mathcal{A}_{akk} = \{akk\}$), und damit

$$Nomen := \left\{ \begin{array}{ll} \textsc{Genus} : & Genus \\ \textsc{Kasus} : & Kasus \end{array} \right\}.$$

Bezüglich der dadurch gegebenen Typhierarchie beschreibt die getypte Strukturbeschreibung

$$[\,\textsc{Genus} = fem]_{Nomen}$$

die vier Merkmalstrukturen

$$\begin{bmatrix} \textsc{Genus} = fem \\ \textsc{Kasus} = nom \end{bmatrix}, \quad \begin{bmatrix} \textsc{Genus} = fem \\ \textsc{Kasus} = gen \end{bmatrix}, \quad \begin{bmatrix} \textsc{Genus} = fem \\ \textsc{Kasus} = dat \end{bmatrix}, \quad \begin{bmatrix} \textsc{Genus} = fem \\ \textsc{Kasus} = akk \end{bmatrix}.$$

Definition 2.8 Sei T eine Menge definierter Typen über $\mathcal{A}$ und $\mathcal{F}$ und $\sqsubseteq$ die dadurch gegebene Verfeinerungsbeziehung zwischen Typen. Ein bezüglich $\sqsubseteq$ *feinster* Typ ist ein Typ $\sigma \in T$, zu dem es keine Verfeinerung $\rho \sqsubseteq \sigma$ mit $\rho \neq \sigma$ gibt. Eine Merkmalstruktur $\mathcal{M} = (M, \langle F^M \rangle_{F \in \mathcal{F}}, 0^M, v^M)$ *ist vom Typ* τ, kurz $\mathcal{M} : \tau$, wenn es einen feinsten Typ $\sigma \sqsubseteq \tau$ und eine verträgliche Typannotation $type : M \to T$ gibt, so daß $type(0) = \sigma$ und $v(m) \in \mathcal{A}_{type(m)}$ für jeden Endpunkt $m \in M$ ist.

Beispiel 2.10 Die vier Merkmalstrukturen aus Beispiel 2.9 sind vom Typ *Nomen*. Das gilt auch dann, wenn man *Genus* und *Kasus* als atomare Typen einführt und $\mathcal{A}_{Genus} \supseteq \{fem, mask\}$ und $\mathcal{A}_{Kasus} \supseteq \{nom, gen, akk, dat\}$ ist.

Jedem Typ $\tau \in T$ entspricht die Menge $D_\tau := \{\, \mathcal{M} \mid \mathcal{M} : \tau \,\}$ aller Merkmalstrukturen vom Typ τ, die mit Hilfe der Verfeinerung $\sqsubseteq$ definiert ist. Aber nicht alle Strukturen gelten als erlaubte Objekte; eine Grammatik sondert daraus mit gewissen Konstruktionsschemata und Auswahlprinzipien noch die (als sprachliche Zeichen) erlaubten Strukturen aus.

3 „Schemata“ und „Prinzipien“ von HPSG

Eine *Head-Phrase-Structure*-Grammatik besteht nicht nur aus einer partiell geordneten Menge $(T, \sqsubseteq)$ definierter Typen. Hinzu kommen „Prinzipien“ der Form $\tau \to \varphi$ aus einem Typ $\tau \in T$ und einer Merkmalstrukturberschreibung φ, durch die der Bereich der Objekte vom Typ τ von

$$D_\tau = \{\, \mathcal{M} \mid \mathcal{M} : \tau \,\} \quad \text{auf} \quad E_\tau = \{\, \mathcal{M} \in D_\tau \mid \mathcal{M} \models \varphi \,\}$$

eingeschränkt wird, wobei $\mathcal{M} \models \varphi$ für „$\mathcal{M}$ wird von φ beschrieben“ (s.o.) oder „$\mathcal{M}$ erfüllt φ“ steht. Durch die Strukturbeschreibungen kann man (über Strukturteilung, aber nicht über Variablen, die nach einer Belegung ja weg wären) erzwingen, daß auch nach Verfeinerungen die Beschreibungen von gewissen Teilstrukturen gleich sein müssen ; in den Typen werden keine Gleichheiten erzwungen).

Außerdem kommen noch „Schemata“ zur Konstruktion größerer Ausdrücke (sprachlicher Zeichen) hinzu; sie entsprechen den Grammatikregeln.

Um die Anzahl der Konstruktionsschemata klein zu halten, wird Information über den Aufbau von Ausdrücken, die bei kontextfreien Grammatiken in die Grammatikregeln eingebaut ist, in die Konstituenten und somit letztlich in die Lexikoneinträge verlagert.

In der folgenden (etwas groben) Formulierung der Prinzipien verwende ich statt „sprachliches Zeichen“ weiterhin „Ausdruck“ oder „Struktur“, meine damit aber nicht nur den syntaktischen Anteil (im Merkmal Cat), sondern auch den phonetischen und semantischen Anteil (in Phon und Cont).

Konstruktionsschemata:

1 *Kopf-Argument-Schema*: Ausdrücke vom Typ *head-argument-structure* haben eine Hauptkonstituente, den Kopf, der die Anzahl und Strukturbeschreibungen der Nebenkonstituenten enthält.

Wenn man in *einem* Konstruktionsschritt alle Ergänzungen mit dem Kopf zu einem gesättigten Ausdruck kombinieren will, lautet das Schema so:

$$\textit{head-argument} \to \begin{bmatrix} \textsc{Head-Dtr} & = \big[\textsc{Cat} = \big[\textsc{Subcat} = \langle \boxed{1}, \ldots, \boxed{n} \rangle\big]\big] \\ \textsc{Non-Head-Dtrs} & = \langle \boxed{1}, \ldots, \boxed{n} \rangle \\ \textsc{Cat} & = \big[\textsc{Subcat} = \langle\rangle\big] \end{bmatrix}$$

Wenn man in einem Konstruktionsschritt nur *eine*, die letzte, Ergänzung mit dem Kopf kombiniert, erhält man i.a. einen ungesättigten Ausdruck (Notation: *Liste* = *Anfang* $\oplus$ $\langle$*letztes Element*$\rangle$):

$$\textit{head-argument} \to \begin{bmatrix} \textsc{Head-Dtr} & = \big[\textsc{Cat} = \big[\textsc{Subcat} = \boxed{1} \oplus \langle \boxed{2} \rangle\big]\big] \\ \textsc{Non-Head-Dtrs} & = \langle \boxed{2} \rangle \\ \textsc{Cat} & = \big[\textsc{Subcat} = \boxed{1}\big] \end{bmatrix}$$

(Das hat bei Implementierungen den Vorteil, daß man die Länge n der Listen nicht festlegen muß.)

2 *Kopf-Adjunkt-Schema*: In einer Kopf-Adjunkt-Verbindung ist die Adjunktionskonstituente gesättigt (erwartet keine Komplemente) und enthält eine Strukturbeschreibung der Kopfkonstituente.

$$\textit{head-adjunct} \to \begin{bmatrix} \textsc{Head-Dtr} & = \boxed{1} \\ \textsc{Non-Head-Dtrs} & = \left\langle \left[\textsc{Cat} = \begin{bmatrix} \textsc{Head} & = \big[\textsc{Mod} = \boxed{1}\big] \\ \textsc{Subcat} & = \langle\rangle \end{bmatrix} \right] \right\rangle \end{bmatrix}$$

3 *Kopf-Spezifikator-Schema*: Kopf-Spezifikator-Struktur enthält die Beschreibungen von Spezifikatoren, die die Kopfkonstituente enthält, außer einer, die durch die Nebenkonstituente (den Spezifikator) gesättigt wird:

$$\textit{head-specifier} \rightarrow \left[\begin{array}{lcl} \textsc{Head-Dtr} & = & \left[\textsc{Cat} = \left[\begin{array}{lcl} \textsc{Spr} & = & \boxed{1} \oplus \langle \boxed{2} \rangle \\ \textsc{Subcat} & = & \langle\rangle \end{array}\right]\right] \\ \textsc{Non-Head-Dtrs} & = & \langle \boxed{2} \rangle \\ \textsc{Cat} & = & \left[\textsc{Spr} = \boxed{1}\right] \end{array}\right]$$

In den anderen Strukturen mit Hauptkonstituente wird kein Spezifikator der Hauptkonstituenten gesättigt:

$$\textit{head-non-specifier} \rightarrow \left[\begin{array}{lcl} \textsc{Head-Dtr} & = & \left[\textsc{Cat} = \left[\textsc{Spr} = \boxed{1}\rangle\right]\right] \\ \textsc{Cat} & = & \left[\textsc{Spr} = \boxed{1}\right] \end{array}\right]$$

4 *Kopf-Füller-Schema*:

Prinzipien

1 *Kopfmerkmalprinzip*: In Strukturen mit einer Hauptkonstituenten stimmen die (syntaktischen) Hauptmerkmale mit den Hauptmerkmalen der Hauptkonstituenten überein.

$$\textit{headed} \rightarrow \left[\begin{array}{lcl} \textsc{Cat} & = & \left[\textsc{Head} = \boxed{1}\right] \\ \textsc{Head-Dtr} & = & \left[\textsc{Cat} = \left[\textsc{Head} = \boxed{1}\right]\right] \end{array}\right]$$

Zu den Hauptmerkmalen zählen u.a. die Wortart, gewisse Formmerkmale, die Beschreibung der modifizierten Konstituente bei Adjunkten.

2 *Semantikprinzip*:

(a) Strukturen mit einer Haupt-, aber ohne eine Adjunktkonstituente, haben dieselbe Semantik wie ihre Hauptkonstituente.

$$\textit{head-non-adjunct} \rightarrow \left[\begin{array}{lcl} \textsc{Cont} & = & \boxed{1} \\ \textsc{Head-Dtr} & = & \left[\textsc{Cont} = \boxed{1}\right] \end{array}\right]$$

(b) Strukturen mit einer Haupt- und einer Adjunktkonstituenten haben dieselbe Semantik wie die Adjunktkonstituente.

$$\textit{head-adjunct} \rightarrow \left[\begin{array}{lcl} \textsc{Cont} & = & \boxed{1} \\ \textsc{Non-Head-Dtrs} & = & \langle \left[\textsc{Cont} = \boxed{1}\right] \rangle \end{array}\right]$$

3 *Valenzprinzip*: Strukturen mit Hauptkonstituente enthalten noch genau die Beschreibungen (fehlender Ergänzungen), die ihre Hauptkonstituente enthält und die nicht durch deren Nebenkonstituenten gesättigt sind.

Wir erlauben einfachheitshalber nur *eine* Nebenkonstituente (binäre Verzweigung in Teilausdrücke).

(a) Einerseits haben wir dann die Fälle, wo die Nebenkonstituente eine Ergänzung ist (die ein Argument sättigt):

$$\textit{head-argument} \rightarrow \left[\begin{array}{lcl} \textsc{Cat} & = & \left[\textsc{subcat} = \boxed{1}\right] \\ \textsc{Head-Dtr} & = & \left[\textsc{Cat} = \left[\textsc{Subcat} = \boxed{1} \oplus \langle \boxed{2} \rangle\right]\right] \\ \textsc{Non-Head-Dtrs} & = & \langle \boxed{2} \rangle \end{array}\right]$$

(b) Andererseits haben wir die Fälle, wo die Nebenkonstituente eine Adjunktion oder ein verschobener Ausdruck (Lückenfüller) ist; dann wird kein Argument gesättigt:

$$\textit{head-non-argument} \rightarrow \begin{bmatrix} \textsc{Cat} & = \begin{bmatrix} \textsc{Subcat} = \boxed{1} \end{bmatrix} \\ \textsc{Head-Dtr} & = \begin{bmatrix} \textsc{Cat} = \begin{bmatrix} \textsc{Subcat} = \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

4 *Spezifikatorprinzip*: Wenn in einer Struktur mit Hauptkonstituente eine Nebenkonstituente die Beschreibung einer Mitkonstituenten enthält (im Spec-Merkmal), so ist dies eine Beschreibung der Hauptkonstituenten.

$$\textit{head-specifier} \rightarrow \begin{bmatrix} \textsc{Head-Dtr} & = \boxed{1} \\ \textsc{Non-Head-Dtrs} & = \langle \begin{bmatrix} \textsc{Cat} = \begin{bmatrix} \textsc{Head} = \begin{bmatrix} \textsc{Spec} = \boxed{1} \end{bmatrix} \end{bmatrix} \end{bmatrix} \rangle \end{bmatrix}$$

Man könnte das Spezifikatorprinzip in die Formulierung des Kopf-Spezifikator-Schemas einbauen.

5 *Prinzip nichtlokaler Merkmale*:

Lexikoneinträge

1 Bei *finiten Verbformen* gibt die Strukturbeschreibung im Lexikoneintrag eine Beziehung (Linking) zwischen den Elementen des Komplementrahmens und den Rollen des durch das Verb gegebenen Prädikats P an.

$$\begin{bmatrix} \textsc{Cat} & = \begin{bmatrix} \textsc{Head} & = [\,\textsc{vform} = \textit{fin}\,]_{verb} \\ \textsc{Subcat} & = \langle [\,]_{\boxed{1}}, \ldots \rangle \end{bmatrix} \\ \textsc{Cont} & = \begin{bmatrix} \textsc{Rolle}_1 & = \boxed{1} \\ \vdots & = \vdots \end{bmatrix}_P \end{bmatrix}$$

Die Komplemente sollen gesättigte Ausdrücke sein, beschrieben durch

$$[\,]_{\boxed{i}} := \begin{bmatrix} \textsc{Cat} & = [\,\textsc{Subcat} = \langle\rangle\,] \\ \textsc{Cont} & = [\,\textsc{Ind} = \boxed{i}\,] \end{bmatrix},$$

und einen „referentiellen Index“ $\boxed{i}$ einführen, der als Argument im logischen Prädikat der Verbbedeutung auftritt.

Z.B. sollen nominale Komplemente eine Individuenvariable mit gewissen morphologischen Eigenschaften einführen, die für die Anaphernauflösung benutzt werden:

$$NP^{nom}_{\boxed{i}} := \begin{bmatrix} \textsc{Cat} & = \begin{bmatrix} \textsc{Head} & = [\,\textsc{Case} = \textit{nom}\,]_{noun} \\ \textsc{subcat} & = \langle\rangle \end{bmatrix} \\ \textsc{Cont} & = [\,\textsc{Ind} = \boxed{i}\,] \end{bmatrix}$$

Bei *Nomenformen* wird ein Determinator als Komplement verlangt, der aber nicht im Komplementrahmen Subcat, sondern in einem eingenen „Spezifikatorrahmen“ Spr beschrieben wird; im Inhalt wird ein referentieller Index (eine Individuenvariable mit morphologischen Eigenschaften) eingeführt und ein (durch das Nomen bezeichnetes) Prädikat, in dem der Index als Argument auftritt:

$$\begin{bmatrix} \textsc{Cat} & = \begin{bmatrix} \textsc{Head} & = [\,]_{noun} \\ \textsc{Spr} & = \langle \textit{Det} \rangle \end{bmatrix} \\ \textsc{Cont} & = \begin{bmatrix} \textsc{Ind} & = \boxed{1} \\ \textsc{Restr} & = \langle [\,\textsc{Inst} = \boxed{1}\,]_P \rangle \end{bmatrix} \end{bmatrix}$$

2 Modifikatoren, z.B. *Adjektive*, enthalten eine Strukturbeschreibung des modifizierten Ausdrucks:

$$\begin{bmatrix} \textsc{Cat} = \begin{bmatrix} \textsc{Head} & = \begin{bmatrix} \textsc{Mod} = [\,]_{sign} \end{bmatrix} \\ \textsc{Subcat} & = \langle \ldots \rangle \end{bmatrix} \end{bmatrix}.$$

Diese Beschreibung ist unter den Kopfmerkmalen, sodaß sie bei der Anbindung von Argumenten weitervererbt wird. Adjektive (und die Ergänzung um ihre Argumente) modifizieren (ggf. schon modifizierte) Nomen, d.h. haben $[\textsc{Mod} = \overline{N}]$ mit

$$\overline{N} = \begin{bmatrix} \textsc{Cat} = \begin{bmatrix} \textsc{Head} & = [\,]_{noun} \\ \textsc{Subcat} & = \langle\rangle \\ \textsc{Spr} & = \langle Det \rangle \end{bmatrix} \end{bmatrix}$$

3 Spezifikatoren, z.B. Quantoren und Possessivpronomen, enthalten eine Strukturbeschreibung des spezifizierten Ausdrucks:

$$\begin{bmatrix} \textsc{Cat} & = \begin{bmatrix} \textsc{Head} & = \begin{bmatrix} \textsc{Spec} = \begin{bmatrix} \textsc{Cont} = \boxed{1} \end{bmatrix}_{sign} \end{bmatrix} \\ \textsc{Subcat} & = \langle\rangle \end{bmatrix} \\ \textsc{Cont} & = \begin{bmatrix} \textsc{Restind} = \boxed{1} \end{bmatrix}_P \end{bmatrix}.$$

Ihre Bedeutung ist ein Prädikat über die Bedeutung des spezifizierten Ausdrucks.

4 Verschobene Konstituenten werden im Lexikon als *Spuren* beschrieben, d.h. als „leere Wörter“ ohne Phonologie und Konstituenten, aber mit einer syntaktischen Beschreibung:

$$\begin{bmatrix} \textsc{Phon} & = \langle\rangle \\ \textsc{Loc} & = \boxed{1} \begin{bmatrix} \textsc{Cat} = \begin{bmatrix} \textsc{Head} & = [\,] \\ \textsc{Subcat} & = \langle\rangle \end{bmatrix} \end{bmatrix} \\ \textsc{Nonloc} & = \begin{bmatrix} \textsc{Slash} = \langle \boxed{1} \rangle \end{bmatrix} \end{bmatrix}_{word}$$

(Das entspricht der kontextfreien Löschregel $Cat/Cat \to \epsilon$.)

Beispiel 3.1 1 Kopf-Argument: Verb+Objekt; Verb+Subjekt; Präposition+Nomen(gesättigt)

2 Kopf-Spezifikator: Nomen (gesättigt) + Determinator

3 Kopf-Adjunktion: Nomen + Adjektiv; Nomen + Relativsatz; Nomen + Präpositionalphrase; Verb+Adverbial

Typhierarchie

Die Konstruktionen sollen die erlaubten Ausdrücke erschöpfend beschreiben. Daher sind die Typen so zu diefinieren, daß sie in folgendem Sinn komplementär zu einander sind:

$$\begin{aligned} \textit{phrase} &= \textit{headed} + \textit{non-headed} \\ \textit{headed} &= \textit{head-argument} + \textit{head-non-argument} \\ \textit{headed} &= \textit{head-adjunct} + \textit{head-non-adjunct} \\ \textit{headed} &= \textit{head-specifier} + \textit{head-non-specifier} \\ \textit{headed} &= \textit{head-filler} + \textit{head-non-filler} \end{aligned}$$

Außerdem sollen gelten:

$$\begin{aligned} \textit{headed} &= \textit{head-non-adjunct} + \textit{head-non-argument} + \textit{head-non-specifier} + \textit{head-non-filler} \\ \textit{head-non-adjunct} &= \textit{head-argument} + \textit{head-specifier} + \textit{head-filler} \\ \textit{head-non-argument} &= \textit{head-adjunct} + \textit{head-specifier} + \textit{head-filler} \\ \textit{head-non-specifier} &= \textit{head-argument} + \textit{head-adjunct} + \textit{head-filler} \\ \textit{head-non-filler} &= \textit{head-argument} + \textit{head-adjunct} + \textit{head-specifier} \end{aligned}$$

Die vier Typen *head-argument*, *head-adjunct*, *head-specifier* und *head-filler* sollen die verschiedenen, unterhalb *head* feinsten Typen der Hierarchie sein.

Konstituentenanordnung

Die Anordnung der Wörter eines Ausdrucks erfolgt im Merkmal PHON: sie ist (normalerweise) eine Verkettung der Wortanordnungen seiner Konstituenten:

$$\begin{bmatrix} \text{PHON} & = f(\boxed{1}, \boxed{2}) \\ \text{HEAD-DTR} & = \begin{bmatrix} \text{PHON} = \boxed{1} \end{bmatrix} \\ \text{NON-HEAD-DTRS} & = \langle \begin{bmatrix} \text{PHON} = \boxed{2} \end{bmatrix} \rangle \end{bmatrix}$$

Hierbei ist $f(\boxed{1}, \boxed{2}) = \boxed{1} \oplus \boxed{2}$ oder $\boxed{2} \oplus \boxed{1}$, je nachdem, ob der Kopf *vor* oder *hinter* seinen Argumenten und Adjunkten steht; das wird im Kopf durch ein Merkmal INITIAL mit den Werten $+$ oder $-$ gesteuert.

Die gewünschte Wahl von f drückt man meist durch sogenannte *Anordnungsregeln* (*linear precedence rules*, *LP*-Regeln) wie

$$\begin{array}{rcl} \text{Head}[\,\text{INITIAL} = +] & < & \text{Argument}, \\ \text{Argument} & < & \text{Head}[\,\text{INITIAL} = -], \\ \text{Adjunct}[\,\text{PRE-MOD} = +] & < & \text{Head}, \\ \text{Head} & < & \text{Adjunct}[\,\text{PRE-MOD} = -], \\ \text{Specifier} & < & \text{Head} \end{array}$$

aus, für die entsprechende Kopfmerkmale eingeführt werden; dadurch wird die Anordnung von den lexikalischen Elementen bestimmt. Allerdings muß man Abweichungen von der so festgelegten Anordnung z.B. durch Verschiebungen zulassen (vgl. Head-Filler-Konstruktionen).

Je nachdem, wie kompliziert die Anordnungsmöglichkeiten f sind, wird bei der Analyse von Eingabesätzen der Aufwand, die relevanten Teile in der Anordnung der Eingabe zu finden, aufwendig. Die Beschreibung in den Strukturen sagt ja nicht, wie die jeweiligen f zu invertieren sind.

Bem. Eine feinere Formulierung der Prinzipien muß Einschränkungen bei den Komplementrahmen machen: die Beschreibung der erforderlichen oder möglichen Ergänzungen (in SUBCAT, SPEC, MOD) darf nur auf die syntaktische Kategorie und den Inhalt (CAT und CONT) Bezug nehmen, z.B. nicht auf den genauen Wortlaut in PHON. Daher gruppiert man die syntaktische und semantische Information zu

$$\begin{bmatrix} \text{LOC} = \begin{bmatrix} \text{CAT} & = [\,]_{cat} \\ \text{CONT} & = [\,]_{cont} \end{bmatrix} \end{bmatrix}$$

und erlaubt nur Komplementrahmen wie in $[\,\text{SUBCAT} = [\,]_{loc\ list}]$; genauer gesagt, werden auch noch Beschreibungen der verschobenen Konstituenten erlaubt.