Frank Richter:
Syntax I

Zusammenfassung: Ein Formalismus für HPSG

Im folgenden wird kurz informell zusammengefaßt, was wir über den der HPSG zugrundeliegenden Formalismus gesagt haben.

1 Signatur

Eine Signatur besteht aus einer Menge von Sortennamen, der Sortenhierarchie, einer Menge von Attributsnamen, den Feature Declarations und einer Menge von Relationsnamen. Zusammen mit dem Relationsnamen wird die Stelligkeit jeder Relation angegeben.

Statt von Sorten spricht man manchmal von Typen, für Attribute sagen wir auch Feature (Merkmale), und die Feature Declarations werden manchmal Appropriatenessbedingungen genannt. Sorten, die keine Untersorten haben, nennt man auch *maximal spezifische* Sorten. Entsprechend spricht man davon, daß eine Sorte spezifischer ist als eine andere, wenn sie in der Sortenhierarchie unterhalb der anderen steht.

Die Appropriatenessbedingungen unterliegen gewissen Anforderungen, die mit dem Stichwort *Merkmalsvererbung* verbunden sind. Wenn ein Attribut für eine Sorte deklariert ist, dann muß sie auch für alle Untersorten dieser Sorte deklariert sein, und der Wert des Attributes ist an den Untersorten mindestens so spezifisch wie an ihren Obersorten. Da es sich hierbei um eine definitorisch festgelegte Eigenschaft von Signaturen handelt, verläßt man sich auf die Gültigkeit dieser Anforderung, wenn man eine Sortenhierarchie und Feature Declarations angibt, und bedient sich einer abkürzenden Schreibweise: Die Feature Declaration eines Attributes an einer Sorte und ihr Sortenwert wird einfach für die höchste Sorte angegeben, für die sie zutrifft, und man nimmt dann einfach implizit an, daß sie gemäß der angegebenen Regeln an die Untersorten weitervererbt wird. So wird zum Beispiel im Buch auf Seite 398 das Attribut SYNSEM für die Sorte *sign* mit dem Wert *synsem* deklariert. Es ist dann klar, daß diese Deklaration sich auf die beiden Untersorten von *sign*, *word* und *phrase*, entsprechend vererbt.

Mit der Signatur strukturiert man die Domäne, über die man sprechen möchte (zum Beispiel Fahrzeuge oder sprachliche Zeichen) und legt fest, mit welchen Sortennamen, Attributsnamen und Relationen man in Beschreibungen darüber sprechen kann.

2 Beschreibungen

Als Beschreibungen verwenden wir sogenannte AVMs oder Attribut-Wert Matrizen. Das sind im wesentlichen geklammerte Ausdrücke der weiter unten charakterisierten Gestalt, in denen die in der Signatur angegebenen Attributsnamen und Sortennamen auftreten dürfen. Wir lassen daneben das Negationssymbol $\neg$ vor AVMs zu, und verknüpfen AVMs durch die logischen Junktoren $\wedge$, $\vee$, $\rightarrow$ und $\leftrightarrow$ (Konjunktion, Disjunktion, Implikation und Äquivalenz). Relationsnamen, gefolgt von runden Klammern, zwischen denen die Argumente der Relation beschrieben werden, können ebenfalls durch die Junktoren untereinander und mit AVMs verknüpft werden.

Darüber hinaus haben wir eine Menge von Variablen zur Verfügung (zum Beispiel verwenden wir dafür die Symbole x, y, z); in AVMs und als Argumente von Relationen schreibt man in der HPSG die Variablen fast immer als natürliche Zahlen in quadratischen Boxen, die man *Tags* nennt. Dies ist eine reine Konvention, und es besteht kein Unterschied zwischen diesen Symbolen als Variablen und der Verwendung von lateinischen Kleinbuchstaben. Sie sind beliebig austauschbar. Neben den Variablen haben wir außerdem die beiden Quantoren $\exists$ und $\forall$ zur Verfügung (Existenz- und Allquantor).

Damit haben wir folgende Möglichkeiten, Beschreibungen zu bilden:

- Wenn *sorte* eine Sorte ist, dann ist [*sorte*] ein AVM. In diesem einfachen Fall lassen wir die eckigen Klammern manchmal weg und schreiben einfach *sorte*.

- Wenn *sorte* und $sorte_1$ bis $sorte_n$ Sorten sind, und ATTRIBUT_1 bis ATTRIBUT_n sind Attribute, dann ist folgendes ein AVM:

$$\begin{bmatrix} \textit{sorte} & \\ \text{ATTRIBUT}_1 & \textit{sorte}_1 \\ \vdots & \vdots \\ \text{ATTRIBUT}_n & \textit{sorte}_n \end{bmatrix}$$

  Wir erlauben es uns manchmal, *sorte* wegzulassen. Im HPSG-Buch wird im Unterschied zu hier *sorte* außen am unteren linken Eck der AVM-Box notiert. An die Stelle von $sorte_1$ bis $sorte_n$ können tiefer eingeschachtelte AVMs oder Tags treten:

$$\begin{bmatrix} \textit{sorte} & \\ \text{ATTRIBUT}_1 & \begin{bmatrix} \textit{sorte}_1 & \\ \text{ATTRIBUT}_{127} & \text{ATTRIBUT}_{13}\ \boxed{1} \\ \text{ATTRIBUT}_{23} & \boxed{1} \end{bmatrix} \\ \text{ATTRIBUT}_2 & \textit{sorte}_2 \end{bmatrix}$$

- Sei `relation` ein in der Signatur angegebenes Relationssymbol einer zweistelligen Relation, und x und y seien Variablen. Dann ist folgendes eine Beschreibung:

  `relation`(x, y)

  An die Stelle von x und y können auch AVMs (und selbstverständlich auch Tags) treten.

- Relationale Beschreibungen der gerade diskutierten Art und AVMs können untereinander und miteinander durch die logischen Junktoren $\vee, \wedge, \rightarrow, \leftrightarrow$ verknüpft werden. Desgleichen können wir das Negationssymbol $\neg$ vor AVMs und relationale Beschreibungen setzen. Zwei Beispiele:

$$\begin{bmatrix} \textit{sorte} \\ \text{ATTRIBUT}_1\ \boxed{1} \\ \text{ATTRIBUT}_3\ \boxed{2} \end{bmatrix} \quad \wedge \quad \texttt{relation}(\boxed{1}, \boxed{2}),$$

$$\begin{bmatrix} \textit{sorte} \\ \text{ATTRIBUT}_1\ \boxed{2} \\ \text{ATTRIBUT}_3\ \boxed{2} \end{bmatrix} \quad \rightarrow \quad \neg \begin{bmatrix} \text{ATTRIBUT}_5 \begin{bmatrix} \textit{sorte}_3 \\ \text{ATTRIBUT}_7\ \boxed{1} \\ \text{ATTRIBUT}_{15}\ \boxed{1} \end{bmatrix} \end{bmatrix}$$

(wobei die genannten Attribute, Sorten und Relationen natürlich aus der Signatur stammen).

- Zu einer Variable können wir weitere Angaben machen, indem wir sie vor einen AVM schreiben. Seien x eine Variable, *sorte* eine Sorte und ATTRIBUT ein Attribut, dann schreiben wir hierzu

  ${}^{x}[\text{ATTRIBUT} \quad \textit{sorte}]$.

- Vor Beschreibungen der diskutierten Art können wir die Quantoren $\exists$ und $\forall$, jeweils gefolgt von einer Variable schreiben. Wenn also zum Beispiel α ein AVM oder eine relationale Beschreibung ist, dann ist

  $\exists x \exists y\ \alpha$ eine Beschreibung.

3 Die Bedeutung von Beschreibungen

Bisher haben wir nur darüber gesprochen, wie wir Sorten, Attribute, Relationssymbole, Variablen und eine Reihe logischer Symbole dazu verwenden können, um komplexe Beschreibungen aufzuschreiben. Wir haben noch nichts über die Bedeutung dieser Symbole und damit über die Bedeutung der Beschreibungen gesagt. Wir wissen also noch nicht, was diese Beschreibungen eigentlich beschreiben.

Als Domäne wählen wir uns eine beliebige Menge von Objekten. Die Grundidee ist dann die, daß die maximal spezifische Sorten diese Menge von Objekten vollständig partitionieren, das heißt, jedes Objekt der Menge hat genau eine maximal spezifische Sorte. Attribute sind Namen für Funktionen, das heißt, ihre Bedeutung ist eine Funktion von der Menge unserer Objekte in die Menge unserer Objekte. Und die Bedeutung von n-stelligen Relationssymbolen ist eine Menge von n-Tupeln von Objekten, nämlich die Menge der n-Tupel von Objekten, die in der entsprechenden n-stelligen Relation sind.

Die Funktionen, welche die Attributssymbole interpretieren, werden zusätzlichen Bedingungen unterworfen. Die Idee ist dabei, daß die Domäne, über die wir sprechen, sich so verhalten soll, wie es die Appropriatenessbedingungen der Signatur vorgeben. Das ist dann der Fall, wenn erstens gilt, daß wenn die Funktion, die durch ein Attribut bezeichnet wird, auf einem Objekt definiert ist, das Attribut für die Sorte, von der dieses Objekt ist, deklariert sein muß, und der Wert der Funktion auf diesem Objekt muß wiederum ein Objekt sein, dessen Sorte dem entspricht, was die Feature Declarations fordern. Umgekehrt muß zweitens gelten, daß wenn die Feature Declarations ein Attribut für eine Sorte deklarieren, jedes Objekt in der Denotation dieser Sorte in der Definitionsmenge der Funktion, die von diesem Attribut benannt wird, sein muß. Letztere Bedingung ist in der Literatur unter dem Namen *total well-typedness* bekannt. Man kann sie auch so zusammenfassen: Entweder sind alle Objekte einer bestimmten maximal spezifischen Sorte in der Definitionsmenge der Funktion, oder keines ist es. Was der Fall ist, legen die Feature Declarations fest.

Nach diesen einleitenden Worten können wir uns jetzt der Bedeutung der oben eingeführten Beschreibungen im Detail widmen:

- *sorte* beschreibt die Menge der Objekte von Sorte *sorte*, wenn *sorte* laut der Sortenhierarchie eine maximal spezifische Sorte ist. Wenn *sorte* keine maximal spezifische Sorte ist, beschreibt es die Menge der Objekte aller maximal spezifischen Sorten, über denen *sorte* in der Sortenhierarchie steht.

- $$\begin{bmatrix} \mathit{sorte} \\ \textsc{attribut}_1\ \mathit{sorte}_1 \\ \textsc{attribut}_2\ \mathit{sorte}_2 \end{bmatrix}$$

  beschreibt die Menge der Objekte, die von *sorte* beschrieben werden und für die außerdem gilt, daß $\textsc{attribut}_1$ und $\textsc{attribut}_2$ auf ihnen definiert sind, sowie daß man von ihnen über das Attribut $\textsc{attribut}_1$ zu einem Objekt gelangt, das von sorte_1 beschrieben wird, und daß man über das Attribut $\textsc{attribut}_2$ zu einem Objekt kommt, das von sorte_2 beschrieben wird.

  $$\begin{bmatrix} \mathit{sorte} \\ \textsc{attribut}_1\ \boxed{1}\ \mathit{sorte}_2 \\ \textsc{attribut}_2\ \boxed{1} \end{bmatrix}$$

  ist die Menge der Objekte, die von *sorte* beschrieben werden, so daß $\textsc{attribut}_1$ und $\textsc{attribut}_2$ auf ihnen definiert ist und man über die beiden Attribute $\textsc{attribut}_1$ und $\textsc{attribut}_2$ zum selben Objekt gelangt, das außerdem durch sorte_2 beschrieben wird.

  Alle Objekte, die wir von einem gegebenen Startobjekt über eine Folge von Attributen auf die angedeutete Art und Weise erreichen können, nennen wir die *Komponenten* dieses Objekts. Wenn ein Objekt von der zuletzt angeführten Beschreibung beschrieben wird, dann ist also das Objekt, zu dem wir gleichermaßen über $\textsc{attribut}_1$ und $\textsc{attribut}_2$ gelangen, eine Komponente jenes Objektes.

- Wenn wir eine Beschreibung a haben, dann beschreibt $\neg a$ die Menge der Objekte, auf die a nicht zutrifft.

  Für zwei Beschreibungen a und b ist $a \vee b$ die Menge der Objekte, auf die a oder b zutrifft, $a \wedge b$ ist die Menge der Objekte, auf die sowohl a als auch b zutreffen, $a \rightarrow b$ ist die Menge der Objekte, so daß wenn a auf sie zutrifft auch b auf sie zutrifft, und $a \leftrightarrow b$ ist schließlich die Menge der Objekte, auf die a zutrifft genau dann wenn b auf sie zutrifft.

  Man bemerke, daß die Implikation $a \rightarrow b$ unter anderem alle Objekte beschreibt, auf die a nicht zutrifft.

- $\exists x\ {}^{x}[\textsc{attribut}\quad \mathit{sorte}]$ beschreibt alle Objekte, in denen es eine Komponente x gibt, auf der das Attribut $\textsc{attribut}$ definiert ist, so daß es einem zu einem Objekt führt, das von *sorte* beschrieben wird.

Entsprechend beschreibt $\forall x\ {}^{x}[\textsc{attribut}\quad \textit{sorte}]$ genau die Objekte, für die gilt, daß auf allen in ihnen enthaltenen Objekten das Attribut ATTRIBUT definiert ist, und dieses Attribut führt immer zu einem Objekt, das von *sorte* beschrieben wird.

- Zuletzt illustrieren wir die Bedeutung von Relationen.

  $$\left[\begin{array}{l} \textit{sorte} \\ \textsc{attribut}_1\ \boxed{1} \\ \textsc{attribut}_3\ \boxed{2} \end{array}\right] \quad \wedge \quad \texttt{relation}(\boxed{1}, \boxed{2})$$

  ist die Menge der Objekte, die vom AVM im ersten Konjunkt beschrieben werden und für die außerdem jeweils gilt, daß das Objekt, das man über das Attribut $\textsc{attribut}_1$ erreicht zu dem Objekt, das man über das Attribut $\textsc{attribut}_3$ erreicht, in der Relation `relation` steht.

4 Grammatik

Eine Grammatik ist eine Signatur zusammen mit einer Menge von Beschreibungen. Diese Menge von Beschreibungen nennen wir die Theorie der Grammatik. Wir können also sagen: Grammatik = $\langle$Signatur, Theorie$\rangle$.

In der HPSG werden die Beschreibungen in der Theorie einer Grammatik häufig *Prinzipien* genannt. Wir hatten uns überlegt, daß linguistische Prinzipien sinnvollerweise die Gestalt einer Implikation haben.

5 Exhaustive Modelle einer Grammatik

Der Begriff des Modells einer Grammatik ist deshalb wichtig, weil uns Modelle sagen, was eine Grammatik bedeutet. Die Objekte im Modell der Grammatik nennen Pollard und Sag gewöhnlich *Feature Structures* (Merkmalstrukturen). Wir werden üblicherweise einfach von Objekten sprechen, ohne uns darauf festzulegen, welcher Art diese Objekte sind.

Die Objekte im Modell einer Grammatik sind von der Art, daß jede Beschreibung in der Theorie der Grammatik jedes Objekt beschreibt, also auch alle Objekte, die Komponenten eines anderen Objektes sind. Das heißt, in Modellen der Grammatik sind nur solche Objekte zugelassen, deren sämtliche "Bestandteile" die Anforderungen der Theorie (und der Signatur) erfüllen.

Damit wir überprüfen können, ob unsere (linguistische) Grammatik keine nicht wohlgeformten, also ungrammatische sprachliche Zeichen zuläßt, fordern wir von den Modellen unserer Grammatik jedoch noch mehr. Nicht nur erfüllen alle darin enthaltenen Objekte mit allen ihren Komponenten die Beschreibungen der Theorie; die Modelle enthalten darüber hinaus sämtliche denkbaren Konfigurationen von Objekten, die von der Theorie zugelassen werden. Modelle mit dieser Eigenschaft nennen wir *exhaustive Modelle*. Für unser sprachwissenschaftliches Räsonieren heißt das folgendes: Wenn versehentlich vergessen wurde, bestimmte, nicht wohlgeformte Zeichen durch die Theorie (oder die Signatur) der Grammatik auszuschließen, dann finden sich Zeichen in dieser fehlerhaften Konfiguration unweigerlich in jedem exhaustiven Modell dieser Grammatik. In anderen Worten: Die Grammatik macht die falsche Vorhersage, daß solche Zeichen in der beschriebenen Sprache vorkommen. Der Fehler in der Grammatik kann also durch die Betrachtung ihrer exhaustiven Modelle gefunden werden.

Ein durch eine Grammatik korrekt spezifiziertes exhaustives Modell modelliert hingegen exakt die Sprache, die mit der Grammatik beschrieben werden soll, also zum Beispiel das Spanische, Englische oder Deutsche. Realistischerweise gibt man natürlich mit einer Grammatik nicht das exhaustive Modell einer gesamten natürlichen Sprache in allen ihren Aspekten an, sondern lediglich ein Fragment einer natürlichen Sprache. Die Autoren des HPSG-Buches sind bestrebt, ein schwerpunktmäßig syntaktisches Fragment des Englischen zu spezifizieren.