

GRUNDLAGEN FÜR DEN COMPUTEREINSATZ IN DER LINGUISTIK: ATTRIBUTE, WERTE, UNIFIKATION

ANDREAS WITT UND STEFAN MÜLLER

Sprache ist ein kompliziertes Gebilde, ein Verständnis ihrer Struktur ist außerordentlich schwer zu erreichen. Dies müssen nicht nur all diejenigen feststellen, die eine Fremdsprache erlernen, sondern auch diejenigen, die versuchen Computerprogramme zu schreiben, die Aspekte von Sprachen verarbeiten können. Solche Programme sollen die Menschen z.B. bei der Textproduktion unterstützen (Rechtschreib- bzw. Grammatikkorrekturprogramme), ihnen die Texterfassung abnehmen (Diktierprogramme) oder sogar selbstständig Übersetzungsaufgaben übernehmen. In all diesen Bereichen sind in den vergangenen Jahren große Fortschritte zu erkennen, dennoch sind die Ergebnisse noch weit von der perfekten Lösung der jeweiligen Aufgaben entfernt. Ein Grund hierfür liegt darin, dass Rechner nicht für die Verarbeitung von Sprache, sondern für mathematische Problemstellungen konzipiert wurden.

1. MODELLE UND MERKMALSSTRUKTUREN

Um Rechner für die Sprachverarbeitung nutzen zu können, müssen die Problemstellungen mathematisiert werden. Seit ca. 20 Jahren bedient sich die Linguistik hierfür eines Formalismus, der die sprachlichen Einheiten, z.B. die Wörter, mit Merkmalen – z.B. ›Kategorie‹ – versieht, die wiederum Werte – z.B. ›Verb‹ – erhalten. Bei Beschreibungen dieser Art wird von Merkmals-Wert-Spezifikationen oder auch Attribut-Wert-Spezifikationen gesprochen. Diese Spezifikationen werden formal in Attribut-Wert-Matrizen dargestellt. Attribut-Wert-Matrizen finden seit ihrer Entwicklung in mehr und mehr linguistischen Theorien Anwendung, insbesondere in den modernen Grammatiktheorien, z.B. in der *Generalized Phrase Structure Grammar* (GPSG, Gazdar et al. 1985), der *Head-Driven Phrase Structure Grammar* (HPSG, Pollard & Sag 1994) oder der *Lexical Functional Grammar* (LFG, Bresnan 2001). Die HPSG wurde auch im Verbmobil-Projekt (1993 bis 2000) zur maschinellen Übersetzung gesprochener Sprache verwendet. Mit ihr wurden syntaktische und semantische Strukturen für das Deutsche, das Englische und das Japanische modelliert.

Wegen dieser starken Verbreitung ist deshalb – trotz ihrer jungen Geschichte – die Verwendung von Attribut-Wert-Spezifikationen aus der modernen Linguistik nicht mehr wegzudenken. Dieses Kapitel führt in die formalen Grundlagen der

Nutzung dieser Formalismen ein, komplexe Grundlagen, die jedoch verstanden sein müssen, um die Theorien verstehen zu können.

1.1 Linguistische Modelle

Die Basis eines Sprachverarbeitungssystems ist die Modellierung linguistischer Phänomene. Die Anforderungen an ein linguistisches Modell können in Analogie zu physikalischen Modellen betrachtet werden. Linguistische Modelle sollen mathematisch formalisierte Konstrukte sein. Mathematische Strukturen modellieren somit die linguistische Realität.

Es gibt eine absolute Beziehung zwischen der Realität und dem Modell. Kiss beschreibt in einer HPSG-Einführung die von der HPSG beschriebenen Merkmalsstrukturen folglich als »[...] totale Objekte, d.h. sie enthalten alle Eigenschaften derjenigen linguistischen Entitäten, die sie modellieren.« (Kiss 1995:21). Diese Sichtweise auf die Modelle unterscheidet sich allerdings etwas von der von Carpenter (1992) gewählten. Carpenter, dessen Arbeit eine der formalen Grundlagen der HPSG bildet, sieht Modelle zwar auch als totale Objekte, allerdings nimmt er an, dass nicht alle Eigenschaften aus der Realität modelliert werden können. Von diesen philosophischen Fragestellungen abgesehen, die Konsequenz der Arbeit mit Modellen ist unstrittig: Der Linguistik stehen nur die modellierten Eigenschaften der Objekte zur Verfügung.

Die Modelle sind nicht Teil der Grammatik. Sie bilden den Beschreibungsgegenstand, die Objektsprache. Auf Instanzen der Objektsprache wird mit dem Terminus technicus ›Merkmalsstruktur‹ referiert. Die Metasprache ist die Beschreibung der Modelle.

Die Domäne des Sprachverarbeitungssystems ist das Modell. Das Ziel der Sprachverarbeitung ist eine (partielle) Beschreibung der Merkmalsstrukturen.

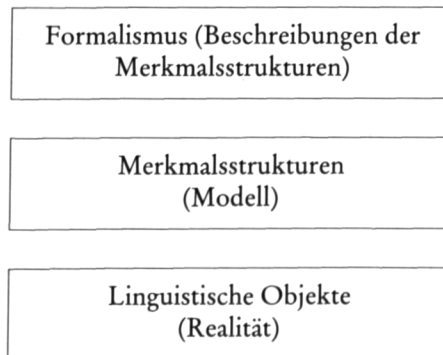


Abb. 1: Ein Ebenenmodell zur formalen Beschreibung sprachlicher Einheiten erlaubt der maschinellen Verarbeitung eine detaillierte Lokalisierung ihres Gegenstandes.

In Abbildung 1 ist die eingeführte Unterscheidung durch die Darstellung unterschiedlicher Ebenen visualisiert. Die Realität, die unterste Ebene bleibt, wird mittels einer abstrakteren Ebene modelliert, wobei diese Modellierung mit den unterschiedlichen Grammatikformalismen variiert. Die oberste Ebene wendet sich der Beschreibung der Modelle zu. Dort ist die maschinelle Verarbeitung situiert.

1.2 Merkmalsstrukturen und deren Beschreibungen

Merkmalsstrukturen sind mathematische Objekte – präziser ausgedrückt: gerichtete Graphen. In Abbildung 2 ist ein Beispiel einer Merkmalsstruktur zu sehen. Das Modell repräsentiert weibliche Eigennamen im Akkusativ (Im derzeit verwendeten Deutsch ist diese Form gleich der Nominativform des Wortes, zu Goethes Zeiten war dies jedoch noch nicht der Fall.).

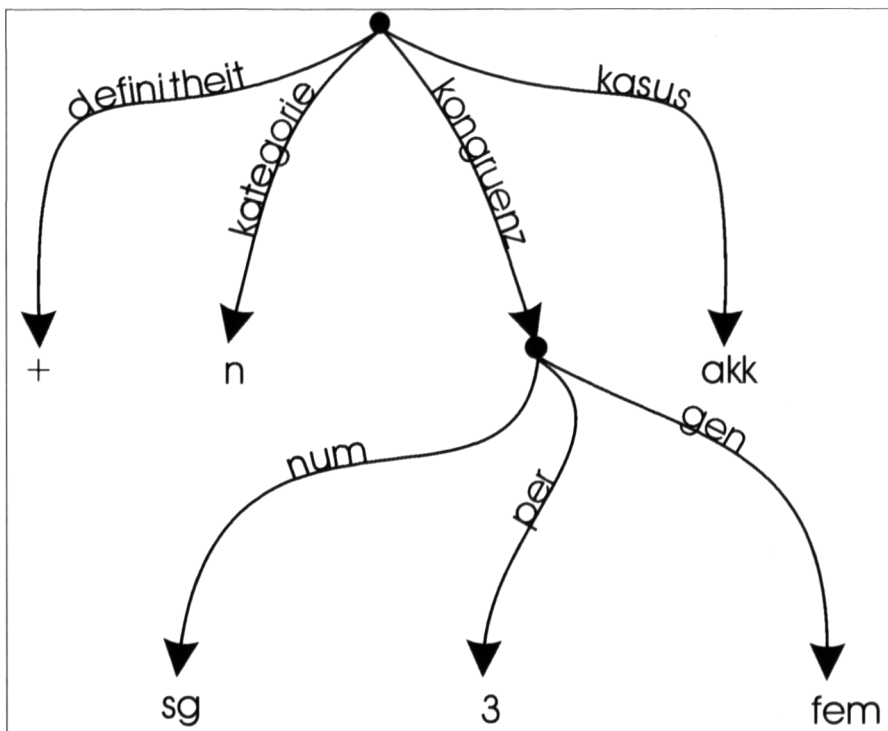


Abb. 2: Repräsentation einer Merkmalsstruktur als mathematische definierter Graph. Merkmalsstrukturen und Beschreibungen von Merkmalsstrukturen können mittels unterschiedlicher Notationen repräsentiert werden.

Die in Abbildung 2 wiedergegebene Repräsentation als Graph stellt nur eine von verschiedenen Möglichkeiten dar. Eine andere Möglichkeit der Darstellung einer Merkmalsstruktur wäre im Prinzip auch durch das Diagramm einer Attribut-Wert-Matrix (AWM; engl. *Attribute-Value-Matrix [AVM]*) möglich. Es ist jedoch auch möglich, dass Merkmalsstrukturen wiederum als Beschreibungsinstrumentarium für Merkmalsstrukturen benutzt werden. D.h., dass das Modell – die mittlere Ebene in Abbildung 1 – durch eine Merkmalsstruktur repräsentiert wird und ihre Beschreibung – die oberste Ebene in dieser Abbildung – ebenfalls durch eine Merkmalsstruktur erfolgt.

Dies kann natürlich zu Verwirrung führen, wie u.a. Pollard und Sag (1994) in ihrem einführenden Kapitel zu ihrer Grammatik HPSG betonen. Als Ausweg aus dieser Konfusion wird eine strikte Notationskonvention eingeführt. Sie verwenden Attribut-Wert-Matrizen als Notation für Beschreibungen von Merkmalsstrukturen, die ihrerseits als Graphen dargestellt werden. Dieser Vorgehensweise haben wir uns angeschlossen. Beispiele für Attribut-Wert-Matrizen finden sich nachfolgend (1a-d).

- (1) a. []
- b. [*kategorie* N]
- c.
$$\left[\begin{array}{l} \textit{kongruenz} \\ \textit{kategorie} \end{array} \begin{array}{l} \left[\begin{array}{ll} \textit{num} & \textit{sg} \\ \textit{per} & 3 \end{array} \right] \\ N \end{array} \right]$$
- d.
$$\left[\begin{array}{l} \textit{kongruenz} \\ \textit{kasus} \\ \textit{kategorie} \\ \textit{definitheit} \end{array} \begin{array}{l} \left[\begin{array}{ll} \textit{num} & \textit{sg} \\ \textit{per} & 3 \\ \textit{gen} & \textit{fem} \end{array} \right] \\ \textit{akk} \\ N \\ + \end{array} \right]$$

AWMs, die Merkmalsstrukturen beschreiben, beinhalten zutreffende – also nicht-falsche – Informationen der Merkmalsstrukturen. Die Beschreibung ist partiell. Eine AWM beschreibt somit alle Merkmalsstrukturen, die keine zur AWM widersprüchlichen Informationen enthalten. Eine Attribut-Wert-Matrix, die zu keiner Merkmalsstruktur im Widerspruch steht, ist die in (1a) zu findende. Es handelt sich hierbei um die leere AWM. Sie beschreibt alle Merkmalsstrukturen, also auch die in Abbildung 2 dargestellte. Genauere Beschreibungen der Merkmalsstruktur leisten die Attribut-Wert-Matrizen in (1b), in (1c) und – als exakte Beschreibung – in (1d).

2. BESCHREIBUNGEN VON MERKMALSSTRUKTUREN

2.1 Aufbau von Merkmalsstrukturbeschreibungen

Merkmalsstrukturen werden also mittels Attribut-Wert-Matrizen beschrieben. Diese sollen nun genauer betrachtet werden. Eine Attribut-Wert-Matrix besitzt endlich viele Merkmale. Jedes Merkmal besitzt einen Wert. Mit einer AWM können somit Merkmale beschrieben werden. Alle Merkmale besitzen Gültigkeit, d.h. es werden nur Merkmale beschrieben, die in der zu beschreibenden Merkmalsstruktur vorhanden sind. Die Merkmale sind somit implizit mit dem logischen UND verknüpft. Eine AWM die keine Merkmale spezifiziert, heißt leere AWM. Eine Darstellung der leeren Attribut-Wert-Matrix ist in (1a) zu finden. Alle nicht-leeren Attribut-Wert-Matrizen spezifizieren Merkmale. Die Werte der Merkmale können atomar oder komplex sein. Beispiele für atomare Werte sind +, *singular* und 3. Die AWMs (2a) und (2b) sind äquivalent. Die Reihenfolge, in der die Merkmale aufgeführt sind, ist irrelevant.

$$(2) \quad \begin{array}{l} \text{a.} \quad \left[\begin{array}{l} \text{attr 1} \quad a \\ \text{attr 2} \quad b \end{array} \right] \\ \text{b.} \quad \left[\begin{array}{l} \text{attr 2} \quad b \\ \text{attr 1} \quad a \end{array} \right] \end{array}$$

AWMs können Merkmale besitzen, deren Werte nicht atomar sind. Dies bedeutet, dass der Wert eines Merkmals eine AWM ist. Eine solche AWM ist in (3) dargestellt.

$$(3) \quad \left[\begin{array}{l} \text{attr 1} \quad \left[\begin{array}{l} \text{attr 11} \quad a \\ \text{attr 12} \quad b \end{array} \right] \\ \text{attr 2} \quad c \end{array} \right]$$

Diese AWM kann, wenn sie mit Leben gefüllt wird, z.B. aussagen, dass das Objekt eines Verbs einem bestimmten Kasus genügen muss. In dem Beispiel erfolgt eine (unvollständige) Aussage über die Wortform *isst*:

$$\text{Beispiel:} \quad \left[\begin{array}{l} \text{objekt} \quad \left[\begin{array}{l} \text{kasus} \quad \text{akk} \\ \text{belebt} \quad - \end{array} \right] \\ \text{kongruenz} \quad \left[\begin{array}{l} \text{person} \quad 3 \\ \text{numersus} \quad \text{sg} \end{array} \right] \end{array} \right]$$

Bei der Verbform *isst* handelt es sich um eine Flexionsform des Verbs *essen*. Sein Objekt muss auf syntaktischer Seite den Kasus Akkusativ besitzen. Auf der semantischen Seite wird ausgedrückt, dass das Patiens nicht belebt sein darf.

Eine fundamentale Eigenschaft von Attribut-Wert-Matrizen ist, dass verschiedene Attribute auf denselben Wert zugreifen können, d.h. ein Wert kann Spezifikation mehrerer Attribute sein. Eine AWM, deren beide Attribute denselben Wert besitzen, ist in (4) abgebildet.

$$(4) \left[\begin{array}{l} \text{attr 1} \\ \text{attr 2} \end{array} \left[\begin{array}{l} 1 \\ 1 \end{array} \right] \left[\begin{array}{l} \text{attr 11} \\ \text{attr 12} \end{array} \left[\begin{array}{l} a \\ b \end{array} \right] \right] \right]$$

Die Notation ist so zu interpretieren, dass die umrahmte Zahl ein Variable darstellt, deren Belegung meist nur einmal wiedergegeben wird. Jedes weitere Auftreten der Variable bedeutet demnach eine identische Belegung oder Bindung. Zur beispielhaften Verdeutlichung soll nochmals die Verbform *isst* dienen, diesmal werden jedoch Aussagen zum Subjekt vorgenommen.

Beispiel:

$$\left[\begin{array}{l} \text{subjekt} \\ \text{form} \end{array} \left[\begin{array}{l} \text{kasus} \\ \text{form} \\ \text{belebt} \end{array} \left[\begin{array}{l} \text{nom} \\ 1 \\ + \end{array} \right] \right] \right]$$

$$\left[\begin{array}{l} \text{form} \\ \text{person} \\ \text{numerus} \end{array} \left[\begin{array}{l} 1 \\ 3 \\ \text{sg} \end{array} \right] \right]$$

Diese AWM besagt, dass die durch die Verbform ausgedrückten Teilinformationen bezüglich der Person und des Numerus identisch sind mit den entsprechenden Teilinformationen des Subjekts. Bei anderen Verbformen, z.B. *isst*, *esse* usw., wird analog hierzu nur einmal spezifiziert, welche Form das Verb besitzt. Die Kongruenz des Verbs mit dem Subjekt wird durch eine Gleichsetzung der Werte der betreffenden Attribute erreicht. Solche Gleichsetzungen sind von einer großen Relevanz in allen unifikationsbasierten Grammatiken, da durch sie die in allen Sprachen vorkommenden Beziehungen zwischen den Worten eines Satzes, zwischen Lexikoneinträgen und Phrasen, zwischen den Wörtern und ihrer Bedeutung oder der Komposition der Satzbedeutung aus der Semantik der Satzteile modelliert werden können z.B. Shieber 1985).

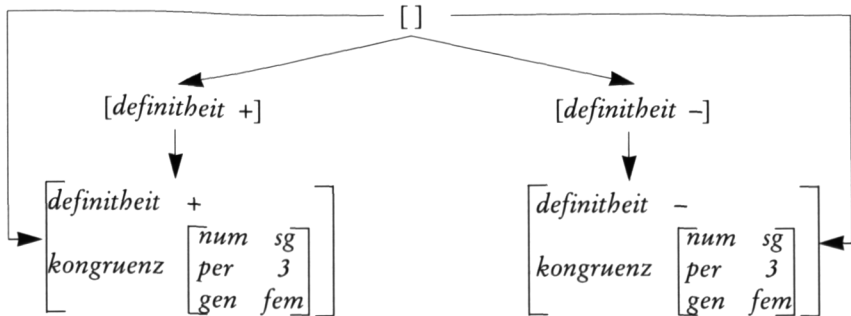


Abb. 3: Subsumtionsbeziehung zwischen Attribut-Wert-Matrizen. Die hier nicht in einer Subsumtionsbeziehung stehenden Einheiten der Sprachbeschreibung sind zueinander inkompatibel.

2.2 Eigenschaften und die Operation Unifikation

Nachdem wir Aufbau und die Relevanz der AWMs betrachtet haben, widmen wir uns in diesem Abschnitt dem zentralen Aspekt der unifikationsbasierten Grammatiken, der (maschinellen) Verarbeitung der AWMs. Ähnlich wie in der Mathematik, wo einerseits Zahlen (z. B. 1, π) und andererseits Operationen (z. B. +, \vee) Anwendung finden, werden die Attribut-Wert-Matrizen mittels Operationen miteinander kombiniert, um z.B. die Bedeutung oder die syntaktische Struktur eines Satzes zu »berechnen«. Die zentrale Operation bei allen unifikationsbasierten Grammatikformalismen heißt Unifikation. Wir werden uns zunächst die Eigenschaften der AWMs ansehen, die für diese Operation von Bedeutung sind.

Die Merkmalsstrukturbeschreibungen in (1) sind unterschiedlich spezifisch. Sie stehen zueinander in einer Subsumtionsbeziehung.

Subsumtion von Attribut-Wert-Matrizen:

Eine AWM D subsumiert eine AWM D' ($D \sqsubseteq D'$) genau dann, wenn alle Informationen, die in D vorhanden sind auch in D' enthalten sind. Dies bedeutet, dass alle von D' beschriebenen Attribut-Wert-Strukturen auch von D beschrieben werden.

Es ist leicht zu sehen, dass die leere AWM jede Merkmalsstruktur subsumiert. Die Attribut-Wert-Matrix (1b) wird folglich auch von (1a) subsumiert, subsumiert jedoch ihrerseits die AWMs (1c) und (1d). Aber auch alle anderen AWMs aus (1) stehen zueinander in der Beziehung der Subsumtion. Zwei Attribut-Wert-Matrizen müssen sich jedoch nicht in einer Subsumtionsbeziehung befinden. In den Abbildungen 3 und 4 sind Beschreibungen von Merkmalsstrukturen dargestellt.

Die Pfeile zeigen, welche AWMs andere Merkmalsbeschreibungen subsumieren. Zwei AWMs, die nicht durch Pfeile verbunden sind, stehen in keiner Subsumtionsbeziehung zueinander. Eine nicht existierende Subsumtionsbeziehung zweier Merkmalsstrukturbeschreibungen kann auf zwei verschiedene Ursachen zurückgeführt werden. Sie können erstens unterschiedliche oder zweitens inkompatible Informationen beinhalten. Die AWMs in Abbildung 3, die zueinander nicht in einer Subsumtionsbeziehung stehen, sind ein Beispiel für den letztgenannten Fall. Sie beinhalten miteinander unverträgliche Informationen. Zwei der Matrizen drücken aus, dass das beschriebene Objekt definit ist. Der Wert des Merkmals »definitheit« ist +. Zwei andere AWMs besitzen die Spezifizierung »definitheit« -. Die unterschiedlichen Werte des Definitheitsmerkmals sind widersprüchliche Informationen. Die AWMs können ausschließlich unterschiedliche Merkmalsstrukturen beschreiben. Deshalb sind sie inkompatibel.

Die nichtleeren Attribut-Wert-Matrizen aus Abbildung 4 hingegen können dieselbe Merkmalsstruktur beschreiben. In der Tat sind die beiden nichtleeren AWMs korrekte partielle Beschreibungen der in Abbildung 2 dargestellten Merkmalsstruktur. Die Beschreibungen beinhalten kompatible, nicht-widersprüchliche Informationen.

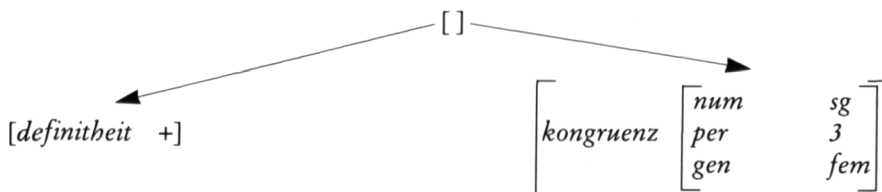


Abb. 4: (Fehlende) Subsumtionsbeziehung von zueinander kompatiblen Attribut-Wert-Matrizen.

Bei einer genaueren Betrachtung kompatibler und inkompatibler AWMs kann festgestellt werden, dass die in kompatiblen Merkmalsstrukturen enthaltenen unterschiedlichen Informationen durch eine AWM beschrieben werden können. Eine Attribut-Wert-Matrix, die all diese Informationen enthält, ist:

$$(5) \left[\begin{array}{l} \begin{array}{l} kategorie \\ definitheit \end{array} \\ kongruenz \end{array} \begin{array}{l} n \\ + \\ \begin{bmatrix} num & sg \\ per & 3 \\ gen & fem \end{bmatrix} \end{array} \right]$$

Diese Matrix beinhaltet neben der Information über die Definitheit und der der Kongruenzmerkmale auch die dazu kompatible, d.h. nicht widersprechende In-

formation für eine Beschreibung der Kategorie.

Eine Attribut-Wert-Matrix, die genau die Informationen zweier kompatibler Merkmalsstrukturen enthält, heißt allgemeinsten Unifikator oder, im Englischen, *most general unifier*.

Der allgemeinste Unifikator ist das Ergebnis der zentralen Operation der Unifikationsgrammatiken. Diese Operation heißt Unifikation oder, im Englischen, *unification*.

Die Operation Unifikation:

Das Ergebnis der Unifikation zweier Attribut-Wert-Matrizen D' und D'' ist die allgemeinste Attribut-Wert-Matrix D , die sowohl von D' als auch von D'' subsumiert wird. Wenn keine solche Struktur existiert, ist die Unifikation nicht definiert. Die Unifikation wird mittels des Symbols \sqcup dargestellt. Das Ergebnis von $D' \sqcup D''$ – beschreibt alle Merkmalsstrukturen, die sowohl von D' als auch von D'' beschrieben werden.

Mit Hilfe dieser Operation ist es somit möglich, die Kompatibilität verschiedener AWMs zueinander zu überprüfen und mehrere kompatible Beschreibungen linguistischer Objekte so in einer adäquaten Form miteinander zu kombinieren. Für beide Fälle sehen wir uns nun ein Beispiel an.

Beispiele für inkompatible AWMs

1. Wir haben bereit gesehen, dass es möglich ist, semantische Restriktionen in Attribut-Wert-Matrizen auszudrücken. Wenn wir versuchen, die bereits weiter oben abgebildete Subjektspezifizierung des Verbs *isst* mit einer Beschreibung eines unbelebten Subjekts, z.B. Wasser, zu unifizieren, ergibt sich:

$$\left[\begin{array}{cc} \textit{kasus} & \textit{nom} \\ \textit{form} & \left[\begin{array}{cc} \textit{person} & 3 \\ \textit{numerus} & \textit{sg} \end{array} \right] \\ \textit{belebt} & + \end{array} \right] \sqcup \left[\begin{array}{cc} \textit{kasus} & \textit{nom} \\ \textit{Art} & \textit{massennomen} \\ \textit{form} & \left[\begin{array}{cc} \textit{person} & 3 \\ \textit{numerus} & \textit{sg} \end{array} \right] \\ \textit{belebt} & - \end{array} \right] = \textit{fehlschlagende Unifikation}$$

Diese Gleichung modelliert die semantische Inadäquatheit des Satzes *Wasser isst*. Beim Versuch der Unifikation der beiden Matrizen führt der widersprüchliche Wert des Attributs ›belebt‹ zu dem Scheitern der Unifikation. Ein Computerprogramm würde einen solchen Satz als unkorrekt zurückweisen und z.B. die Autorin des Satzes auffordern, ihn zu korrigieren.

2. In Kapitel 6 haben wir anhand der Nominalphrase *die kleinen, grünen, runden Tomaten* die Modellierung der Produktivität der Sprache mittels Phrasenstrukturgrammatiken gesehen. Auf der Seite der Syntax lassen sich beliebig viele Adjektive vor das Substantiv stellen. Jedes dieser Adjektive trägt auch zur Bedeutung bei.

kongruenz	person	3		kongruenz	person	3	
kasus	numerus	plural	⊥	kasus	numerus	plural	⊥
semantik	objekt	Tomate		semantik	farbe	grün	
kongruenz	person	3		kongruenz	person	3	
kasus	numerus	plural	⊥	kasus	numerus	plural	
semantik	form	rund		semantik	größe	klein	
kongruenz	person	3		kongruenz	numerus	plural	
kasus	numerus	plural		kasus	nom	objekt	Tomate
semantik	objekt	Tomate		semantik	größe	klein	
	form	rund			farbe	grün	

2.3 Erweiterungen

Die oben beschriebenen Attribut-Wert-Matrizen sind in ihrer Ausdrucksmächtigkeit eingeschränkt. Karttunen (1984) deshalb schlägt Erweiterungen der Merkmalsstrukturbeschreibungen vor.

2.3.1 Disjunktion

Die Beschreibungen der Merkmalsstrukturen sind so aufgebaut, dass jedes Attribut-Wert-Paar, welches formuliert wurde, Gültigkeit besitzt. Eine AWM ist eine Konjunktion der Beschreibungen. Es ist nicht möglich auszudrücken, dass eines von mehreren zur Auswahl stehenden Attribut-Wert-Paaren gelten soll. Disjunktionen von Merkmalen bzw. Werten sind nicht möglich. Um eine Merkmalsstruktur des deutschen Artikels *die* zu beschreiben, können die folgenden vier AWMs formuliert werden:

(6) a.	infl	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">kasus</td> <td style="padding: 5px;">nom</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">kongruenz</td> <td style="padding: 5px;"> <table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">genus</td> <td style="padding: 5px;">fem</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">numerus</td> <td style="padding: 5px;">sg</td> </tr> </table> </td> </tr> </table>	kasus	nom	kongruenz	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">genus</td> <td style="padding: 5px;">fem</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">numerus</td> <td style="padding: 5px;">sg</td> </tr> </table>	genus	fem	numerus	sg
kasus	nom									
kongruenz	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">genus</td> <td style="padding: 5px;">fem</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">numerus</td> <td style="padding: 5px;">sg</td> </tr> </table>	genus	fem	numerus	sg					
genus	fem									
numerus	sg									
b.	infl	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">kasus</td> <td style="padding: 5px;">akk</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">kongruenz</td> <td style="padding: 5px;"> <table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">genus</td> <td style="padding: 5px;">fem</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">numerus</td> <td style="padding: 5px;">sg</td> </tr> </table> </td> </tr> </table>	kasus	akk	kongruenz	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">genus</td> <td style="padding: 5px;">fem</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">numerus</td> <td style="padding: 5px;">sg</td> </tr> </table>	genus	fem	numerus	sg
kasus	akk									
kongruenz	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">genus</td> <td style="padding: 5px;">fem</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">numerus</td> <td style="padding: 5px;">sg</td> </tr> </table>	genus	fem	numerus	sg					
genus	fem									
numerus	sg									
c.	infl	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">kasus</td> <td style="padding: 5px;">nom</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">kongruenz</td> <td style="padding: 5px;"> <table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">numerus</td> <td style="padding: 5px;">pl</td> </tr> </table> </td> </tr> </table>	kasus	nom	kongruenz	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">numerus</td> <td style="padding: 5px;">pl</td> </tr> </table>	numerus	pl		
kasus	nom									
kongruenz	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">numerus</td> <td style="padding: 5px;">pl</td> </tr> </table>	numerus	pl							
numerus	pl									
d.	infl	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">kasus</td> <td style="padding: 5px;">akk</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">kongruenz</td> <td style="padding: 5px;"> <table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">numerus</td> <td style="padding: 5px;">pl</td> </tr> </table> </td> </tr> </table>	kasus	akk	kongruenz	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">numerus</td> <td style="padding: 5px;">pl</td> </tr> </table>	numerus	pl		
kasus	akk									
kongruenz	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">numerus</td> <td style="padding: 5px;">pl</td> </tr> </table>	numerus	pl							
numerus	pl									

Durch Zulassung disjunkter Werte kann die in den vier Attribut-Wert-Matrizen enthaltene Information in einer Attribut-Wert-Matrix ausgedrückt werden:

(7)	infl	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">kasus</td> <td style="padding: 5px;">nom v akk</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">kongruenz</td> <td style="padding: 5px;"> <table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">genus</td> <td style="padding: 5px;">fem</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">numerus</td> <td style="padding: 5px;">sg</td> </tr> </table> </td> </tr> </table>	kasus	nom v akk	kongruenz	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">genus</td> <td style="padding: 5px;">fem</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">numerus</td> <td style="padding: 5px;">sg</td> </tr> </table>	genus	fem	numerus	sg	v [numerus pl]
kasus	nom v akk										
kongruenz	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;">genus</td> <td style="padding: 5px;">fem</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">numerus</td> <td style="padding: 5px;">sg</td> </tr> </table>	genus	fem	numerus	sg						
genus	fem										
numerus	sg										

Diese auf den ersten Blick sehr vernünftige Reduzierung redundanter Informationen hat allerdings einen Preis. Kasper und Rounds (1986) stellen heraus, dass der Berechnungsaufwand der Unifikation disjunkter AWMs eine exponentielle Komplexität besitzt, d.h. der Aufwand bei der Verarbeitung mit Rechnern außerordentlich hoch ist und dass die Disjunktion die Ursache hierfür bildet. Für eine schnellere Verarbeitung mit dem Rechner kann es daher durchaus sinnvoll sein, Merkmalsstrukturbeschreibungen mit disjunktiv spezifizierten Werten in disjunktionfreie AWMs umzuwandeln, bzw. die Erweiterung des Formalismus gar nicht erst zuzulassen.

2.3.2 Negation

Im Englischen müssen, wie im Deutschen auch, Subjekt und Verb in Numerus und Person übereinstimmen. Im Englischen werden die meisten Verben jedoch nur in der 3. Person Singular durch die Endung *-s* flektiert. Diese Verben verlangen dann ein Subjekt in der 3. Person Singular. Die unflektierte Form ist mit allen Subjekten verträglich, die nicht in der 3. Person Singular stehen. Dies kann mit-

tels eines negierten Wertes des Kongruenzmerkmals zum Ausdruck gebracht werden:

$$(8) \left[\begin{array}{l} \text{kongruenz} \end{array} \right] \neg \left[\begin{array}{ll} \text{person} & 3 \\ \text{numerus} & \text{sg} \end{array} \right]$$

Unter gewissen Voraussetzungen kann man eine Merkmalsstruktur mit Negation in eine ohne Negation umwandeln, die dafür Disjunktionen enthält.

2.3.3 Zyklizität

Karttunen (1984) stellt heraus, dass in vielen Implementierungen von Unifikationsgrammatiken zyklische Strukturen ausgesondert wurden. Zyklische Merkmalsstrukturen sind Merkmalsstrukturen, in denen Pfade von einem Merkmal auf dasselbe Merkmal zeigen. Jedoch wird von ihm angemerkt, dass zyklische Strukturen nicht notwendig sind, da es nicht gezeigt werden kann, dass sie notwendig sind.

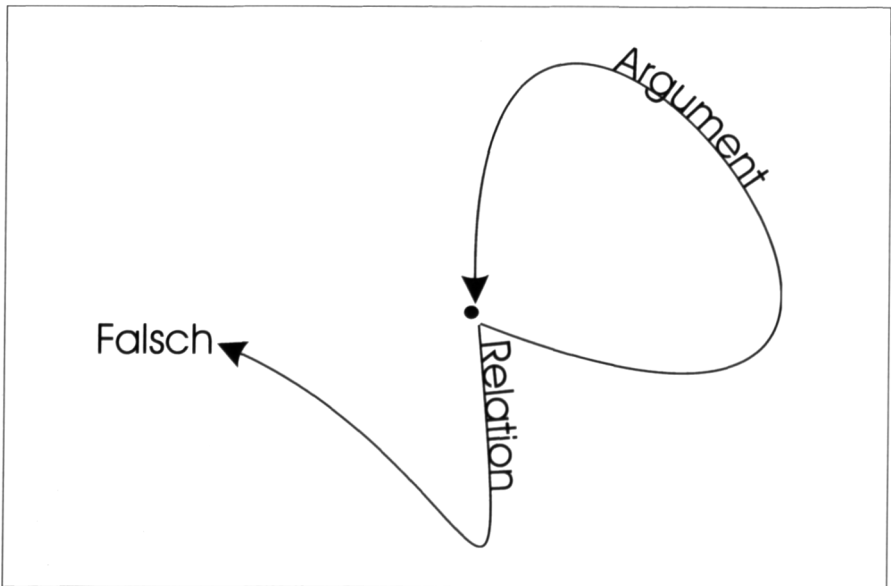


Abb. 5: Durch 9b beschriebene zyklische Attribut-Wert-Struktur. Derartige Strukturen sind sehr selten und werden deshalb von manchen Formalismen verboten.

Jedoch zeigt Carpenter (1992) an einem immer wieder herangezogenen Beispiel für widersprüchliche Informationen, dass es zyklische Strukturen in der natürlichen Sprache gibt. Eine Variation dieses Beispiels ist in der – in Abbildung 5 dargestellten – Modellierung der Semantik des Satzes (9a) zu finden. Eine Beschreibung dieser Merkmalsstruktur ist in (9b) angegeben.

- (9) a. Dieser Satz ist falsch.
 b.

1	Relation falsch
Argument	1

In dieser – in sich widersprüchlichen – Interpretation bezieht sich die Nominalphrase *dieser Satz* auf den propositionalen Gehalt desselben Satzes. Damit ist die ausgedrückte Proposition das Argument des Prädikates »Falsch«.

3. BESCHREIBUNGEN GETYPTER MERKMALSSTRUKTUREN

Getypte Attribut-Wert-Strukturen werden durch getypte Attribut-Wert-Matrizen beschrieben. Getypte AWMs unterscheiden sich von einfachen AWMs durch die Angabe eines Typs. Ein solcher Typ denotiert einen Typen in einem Typensystem.

Ein Typ einer Attribut-Wert-Struktur bestimmt, welche Merkmale diese Attribut-Wert-Struktur besitzt. Es wird davon gesprochen, dass Typen Angemessenheitsbedingungen aufstellen. Wenn eine Attribut-Wert-Struktur Merkmale besitzt, die von dem zugehörigen Typ nicht bestimmt sind, ist die Attribut-Wert-Struktur nicht korrekt getypt, oder – positiv formuliert – wenn eine Attribut-Wert-Struktur ausschließlich Merkmale besitzt, die durch ihren Typ legitimiert sind, ist diese Attribut-Wert-Struktur richtig getypt (engl. *well-typed*). Wenn alle durch ihren Typ erlaubten Merkmale einer Attribut-Wert-Struktur spezifiziert wurden, wird davon gesprochen, dass die Attribut-Wert-Struktur vollständig richtig getypt ist (engl. *totally well-typed*).

3.1 Das Typensystem

Das Typensystem besteht aus einer endlichen Menge von Typen. Diese Menge nennen wir *Type*. Die Typen besitzen einen unterschiedlichen Grad der Allgemeinheit. Dies bedeutet, dass der Informationsgehalt der Typen unterschiedlich ist. Manche Typen beschreiben Entitäten, die auch von anderen Typen beschrieben werden. Dies kann zwei Ursachen haben. Zum einen ist es möglich, dass zwei (oder mehr) Typen denselben Informationsgehalt besitzen. In diesem Fall sind die Typen nicht wirklich unterschiedlich, es handelt sich vielmehr um alpha-

betische Varianten desselben Typs. Wenn einer von zwei Typen jedoch einen höheren Informationsgehalt als der andere besitzt, wird davon gesprochen, dass der mit dem höheren Informationsgehalt spezifischer ist als der Typ mit dem geringeren Informationsgehalt. Das Typensystem besitzt eine Ordnung. Die Ordnung richtet sich nach der Spezifik der enthaltenen Typen.

Es ist hervorzuheben, dass die durch das Typensystem erreichte Ordnung bewirkt, dass allgemeinere Typen ihren Informationsgehalt an spezifischere Typen vererben. Es sei die Konvention eingeführt, dass Variablen, deren Werte Typen sind, mit kleinen griechischen Buchstaben ($\alpha, \beta, \gamma, \delta, \dots$) bezeichnet werden. Mengen von Typen werden mit griechischen Großbuchstaben ($\Gamma, \Delta, \Theta, \dots$) bezeichnet. Ein Typ τ ist spezifischer als ein Typ σ , wenn τ Informationen von σ erbt. In einem solchen Fall wird gesagt, dass $\sigma \tau$ subsumiert:

Subsumtion von Typen:

Der Typ σ subsumiert den Typ τ ($\sigma \sqsubseteq \tau$), wobei σ und $\tau \in Type$, genau dann, wenn jedes Objekt, welches vom Typ τ ist, auch vom Typ σ ist.

Wenn ein Typ σ einen Typ τ subsumiert, wird davon gesprochen, dass σ ein Supertyp des Typs τ und τ ein Subtyp von σ ist. Ein Subtyp des Typs σ ist spezifischer und ein Supertyp allgemeiner als σ .

Die Begriffe Subtyp und Supertyp sollen anhand des Beispiels einer einfachen Wissensbasis verdeutlicht werden.

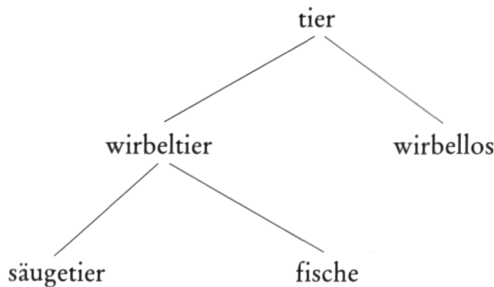


Abb. 6: Klassifikation von Tieren in einer Hierarchie.

Die in dieser Wissensbasis aufgeführte Typenmenge besteht aus:

$$T = \left\{ \begin{array}{l} \text{tier} \\ \text{wirbeltier} \\ \text{wirbellos} \\ \text{säugetier} \\ \text{fisch} \end{array} \right\}$$

Es gelten die folgenden Subsumtionsbeziehungen:

- $tier \sqsubseteq wirbeltier$,
- $tier \sqsubseteq wirbellos$,
- $tier \sqsubseteq säugetier$,
- $tier \sqsubseteq fisch$,
- $wirbeltier \sqsubseteq säugetier$ und
- $wirbeltier \sqsubseteq fisch$

Der Typ *tier* ist Supertyp seiner Subtypen *wirbeltier*, *wirbellos*, *säugetier* und *fisch*.

Die in den neueren Grammatiktheorien verwendeten Typensysteme müssen den folgenden Anforderungen genügen:

1. Das Typensystem besteht aus einer Menge von Typen.
2. Die Ordnung des Typensystems wird durch die Subsumtionsrelation ausgedrückt.
3. Zwei Typen des Typensystems, die gemeinsame Subtypen besitzen, besitzen genau einen gemeinsamen allgemeinsten Subtyp.

Das soll anhand des Beispiels einer einfachen Wissensbasis verdeutlicht werden.

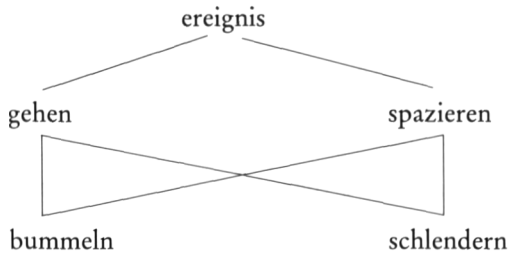


Abb. 7: Nicht akzeptierte Typenhierarchie.

Ein Typensystem, das unseren Anforderungen nicht genügt, ist in Abbildung 7 dargestellt. Die Typenmenge, die aus *ereignis*, *gehen* und *spazieren* besteht, ist konsistent. Die Typen *gehen* und *spazieren* haben jedoch keinen gemeinsamen allgemeinsten Subtyp, da sowohl *bummeln* als auch *schlendern* gleichberechtigte Subtypen sind. Damit verstößt diese Typenmenge gegen die dritte Anforderung. Es ist allerdings möglich, dass eine Typenhierarchie, die nur gegen die dritte Bedingung an ein Typensystem verstößt, immer durch Einfügung von zusätzlichen Typen in ein korrektes Typensystem überführt werden:

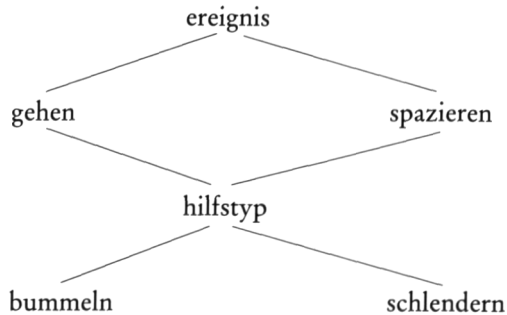


Abb. 8: Akzeptable Typenhierarchie.

Aus Abbildung 8 ist zu ersehen, dass es – im Gegensatz zu den bisherigen Beispielen – auch Hierarchien geben kann, bei denen ein Subtyp mehrere, verschiedene Supertypen besitzt. Es ist natürlich durchaus möglich, dass ein solches Typensystem nicht nur aufgrund eingeführter Hilfstypen zustande kommt, sondern eine sinnvolle Modellierung linguistischen oder nicht-linguistischen Wissens sein kann. Sinnvoll sind diese Hierarchien insbesondere deshalb, weil die Vererbung von Information von mehreren Supertypen ausgenutzt werden kann, wie das folgende Beispiel in Abbildung 9 zeigt.

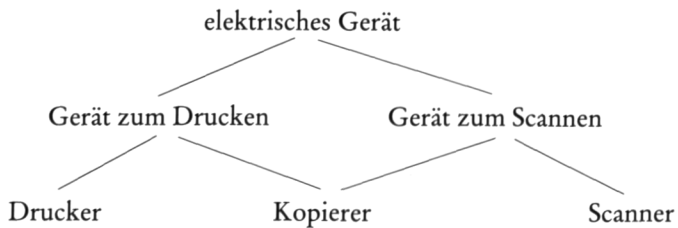


Abb. 9: Ein nichtlinguistisches Beispiel für eine Typenhierarchie.

Der Kopierer erbt vom ›Druckgerät‹ und vom ›Scanngerät‹ und diese vom elektrischen Gerät. Ein elektrisches Gerät besitzt z.B. die Eigenschaft, eine Stromversorgung zu haben, diese Eigenschaft wird an Drucker und an Scanner vererbt. Ein Drucker kann Informationen ausgeben, ein Scanner Informationen erfassen. Ein Kopierer kann beides. ›Kopierer‹ ist die Unifikation von Gerät zum Scannen und Gerät zum Drucken.

3.2 Getypte Merkmalsstrukturbeschreibungen

In diesem Kapitel werden die Attribut-Wert-Matrizen aus Abschnitt 2.1 in der Form erweitert, dass sie Merkmalsstrukturen beschreiben können, die mit Typen versehen sind. Die Typen sind im vorangegangenen Abschnitt eingeführt worden. Attribut-Wert-Matrizen, die getypte Attribut-Wert-Strukturen beschreiben können, heißen getypte Attribut-Wert-Matrizen.

Eine getypte Attribut-Wert-Matrix besitzt endlich viele Merkmale und einen Typ. Die Merkmale können atomar oder komplex sein. Eine Attribut-Wert-Matrix, die keine Merkmale beschreibt und den allgemeinsten Typ denotiert, ist das Analogon zur leeren Attribut-Wert-Matrix.

$$(10) \sigma []$$

Bei dieser Darstellung ist zu beachten, dass die Typenbezeichnungen σ relativ zu dem definierten Typensystem zu sehen ist. Das heißt, dass mit σ der Typ bezeichnet wird, der alle Typen des Typensystems subsumiert.

Subsumtion von getypten Attribut-Wert-Matrizen:

Eine AWM D_σ subsumiert eine AWM D'_τ ($D_\sigma \sqsubseteq D'_\tau$) genau dann, wenn alle Informationen, die in D vorhanden sind, auch in D' enthalten sind und ($\sigma \sqsubseteq \tau$) darüber hinaus gilt. Dies bedeutet, dass alle von D'_τ beschriebenen Attribut-Wert-Strukturen auch von D_σ beschrieben werden.

Unifikation von getypten Attribut-Wert-Matrizen:

Das Ergebnis der Unifikation zweier getypter Attribut-Wert-Matrizen D'_σ und D''_τ ist die allgemeinste getypte Attribut-Wert-Matrix D_γ , die von D'_σ und D''_τ subsumiert wird. Wenn keine solche Struktur existiert, ist die Unifikation nicht definiert.

4. AUSBLICK

Mit den hier vorgestellten AWMs werden alle Eigenschaften linguistischer Objekte gleichermaßen beschrieben. Teiltheorien für Phonologie, Morphologie, Syntax und Semantik benutzen dieselben Beschreibungsmittel. Linguistische Generalisierungen innerhalb der Teiltheorien und auch innerhalb der gesamten Theorie lassen sich über die Modellierung im Typsystem gut erfassen. Dinge, die für eine bestimmte Klasse von Objekten charakteristisch sind, werden in einer Typdefinition zusammengefasst.

Auch die Idee von der Universalgrammatik, einem Teil der Grammatik einer Sprache, der allen Sprachen gemein ist, lässt sich mit Typen adäquat umsetzen. Es gibt allgemeine Typen, die in den Typsystemen aller Sprachen benutzt werden, und es gibt von diesen Typen abgeleitete Subtypen, die nur für bestimmte Sprachfamilien oder Einzelsprachen relevant sind.

Aus der Sicht des Computerlinguisten hat das Typsystem aber auch noch andere Vorteile: Man kann das Wissen über die Grammatik kompakt und redundanzfrei repräsentieren. Ändert man in der Spezifikation eines Typs einen Wert, so wird diese Änderung durch die Vererbung automatisch auf alle Untertypen übertragen. Alle Objekte, die durch AWMs mit entsprechenden Typen beschrieben werden, haben automatisch die geänderte Spezifikation, ohne dass man alle Beschreibungen von Objekten einzeln ändern müsste.

In den letzten Jahrzehnten sind auf dem Gebiet der formalen Grammatiken große Fortschritte gemacht worden (Bresnan 2001; Pollard & Sag 1994; Müller 1999 und Bermann & Frank 1996), die sich auch in Implementierungen großer Grammatiken niedergeschlagen haben. Beispiele für Implementierungen sind die LFG-Grammatiken für das Englische, Französische und Deutsche und die am CSLI in Stanford und am DFKI in Saarbrücken entwickelten HPSG-Grammatiken für das Deutsche und das Englische, die im Verbmobil genutzt wurden. (Die CSLI-Grammatik und die von Müller (1999) beschriebene Grammatik kann im World Wide Web unter <http://lingo.stanford.edu/demo.html> bzw. <http://www.dfki.de/~stefan/Babel/> ausprobiert werden.)

Abschließend sei auf zwei ausführlichere, jedoch auch etwas formale deutschsprachige Einführungen, nämlich Haugeneder und Trost (1993) sowie Kolb (2001) verwiesen, die zur Vertiefung des Verständnisses der Thematik »Attribute, Werte, Unifikation« herangezogen werden können.

5. ARBEITSAUFGABEN ZU KAPITEL 21

- I. In Abschnitt 3.3.2 wurde gesagt, dass eine Merkmalsstruktur mit Negation in eine negationsfreie Merkmalsstruktur mit Disjunktionen umgewandelt werden. Zeigen Sie dies anhand der dort in Beispiel (8) aufgeführten Merkmalsstruktur für englische Verben.
- II. Erstellen Sie eine Merkmalsstruktur, die als Lexikoneintrag des Wortes »kenne« verwendet werden kann.
- III. Entwickeln Sie eine Typhierarchie. In ihr sollen (mindestens) die folgenden Typen enthalten sein: Gurke, Lebensmittel, Apfelsaft, Obst, Tomate, Apfel, Birne, Gemüse, künstlich, natürlich, Cola, Getränk.