# Natürlichsprachliche Systeme

Vorlesung WS 1995/6

Katharina Morik

Lehrstuhl VIII
Fachbereich Informatik
Universität Dortmund

17.Oktober 1995

**Natürlichsprachliche Systeme**

1. Worum geht es? ..... 3
1.1 Natürlichsprachliche Systeme ..... 3
1.2 Aspekte der Sprache ..... 4
1.3 Ebenen der Sprache ..... 5
1.4 Kommunikation ..... 6
1.5 Vorgehensweise beim "linguistic engineering" ..... 8
1.6 Architektur eines NLS ..... 9
1.7 Warum Prolog? ..... 9
2. Syntax ..... 11
2.1 Satzteile ..... 12
2.2 Parsing in Prolog ..... 14
2.2.1 Beispiel ..... 15
2.2.2 DCG ..... 15
2.2.3 Aufbau von Strukturbeschreibungen ..... 16
2.2.4 Syntax in DCG mit Merkmalen ..... 17
2.3 Unifikationsbasierte Grammatik ..... 18
2.3.1 Übersichtliche Repräsentation für Merkmalsstrukturen ..... 19
3. Lexikon ..... 20
3.1 Morphologische Analyse ..... 22
3.2 Lexikoneinträge ..... 24
3.2.1 Lexikonzugriff ..... 25
4. Semantik ..... 26
4.1 Prädikatenlogik als SRS ..... 28
4.1.1 Lambda-Abstraktion ..... 29
4.1.2 Aufbau der semantischen Beschreibung ..... 30
4.1.3 Begriffliches Wissen ..... 32
4.1.3 Behandlung von Quantoren ..... 34
4.1.4 Auswertung der semantischen Beschreibung ..... 36
5. Pragmatik ..... 38
5.1 Deixis ..... 39
5.2 Präsuppositionen ..... 39
5.3 Konversationsimplikatur ..... 41
5.4 Sprechhandlungen ..... 44
5.4.1 Sprechakttheorie ..... 45
5.4.2 Das KAMP System ..... 46
5.5 Dialogstruktur ..... 48
5.5.1 Sprecherwechsel ..... 48
5.5.2 Dialogsteuerung ..... 49
5.6 Anaphora ..... 50
6. Einige NLS ..... 51
6.1 PLIDIS ..... 52
6.1.1 Präsuppositionsbehandlung ..... 53
6.2 HAM-ANS ..... 54
6.2.1 Wissensquellen ..... 56
6.2.2 Nominalphrasenreferenz ..... 57
7. Und nun? ..... 58
8. Literatur ..... 59
Index ..... 60

## 1. Worum geht es?

In dieser Vorlesung wird in ein bedeutendes Teilgebiet der Künstlichen Intelligenz (KI) eingeführt - die natürlichsprachlichen Systeme (NLS). Die Sprachfähigkeit des Menschen ist eine Intelligenzleistung und ermöglicht intelligentes Verhalten.

> *"Sprache ist sowohl das Hauptmedium der gesellschaftlichen Konstruktion der Wirklichkeit, als auch das Hauptmedium der Vermittlung gesellschaftlich konstruierter Wirklichkeit. ... Die Sprache ist nicht nur ein quasi-ideales Bedeutungssystem, sondern auch der wichtigste Träger des gesellschaftlichen Wissensvorrats und ein Kommunikationssystem unter anderen. Sie ist sowohl Wissensform wie Handlungssystem."*
>
> (Thomas Luckmann "Aspekte einer Theorie der Sozialkommunikation" in: Althaus, Henne, Wiegand (eds): Lexikon der Germanistischen Linguistik, Tübingen, $1980^2$, S. 39)

Insofern ist die operationale Beschreibung natürlicher Sprache zentraler Gegenstand der KI. Das Forschungsgebiet befindet sich im Schnittbereich von Informatik und Linguistik. Es ist deshalb notwendig, auch Grundzüge der Linguistik zu vermitteln. Da viele natürlichsprachliche Systeme als Dialogschnittstellen zu Leistungssystemen (z.B. Datenbanken ) eingesetzt werden, wird auch der Bereich der Kommunikation angesprochen, mit dem sich Linguistik immer auch beschäftigt hat. Da die Beschreibungen der KI operational sind, brauchen wir ablauffähige Formalismen. In dieser Vorlesung wählen wir Prolog, insbesondere den DCG-Formalismus in Prolog zur operationalen Beschreibung.

Die Lernziele der Vorlesung sind:

- Überblick über natürlichsprachliche Systeme: was können sie, woraus bestehen sie, wo liegen die Probleme?
- Verwendung von DCGs: nicht nur bei natürlichsprachlichen Benutzerschnittstellen müssen Strukturen von Eingabeketten behandelt werden - die Technik ist vielfältig einsetzbar.
- Respekt vor natürlicher Intelligenz, insbesondere vor dem Reichtum an Phänomenen, der durch die natürliche Sprache und die menschliche Kommunikation gegeben ist.

Das Skript ersetzt nicht die Teilnahme an der Vorlesung. Es ersetzt auch nicht eigenständiges wissenschaftliches Arbeiten. Vielmehr soll es als Wegweiser dienen, der auf sorgfältig ausgewählte Inhalte und Schriften hinweist. Das Skript enthält die wichtigsten Stichpunkte zum Thema und verweist auf relevante Literatur. Die wichtigste Literatur ist als Anhang angegeben und kann als Einstieg in vertiefende Studien genutzt werden. Weitere Literaturhinweise sind im Text angebracht. Diese Hinweise sind als Quellenverweise zu betrachten.

### 1.1 Natürlichsprachliche Systeme

Natürliche Sprache kann als vorgegebene Zeichenkette von Programmen ausgegeben werden (canned text) oder mit statistischen Verfahren bearbeitet werden oder in Programmnamen oder Menüpunkten einer Benutzerschnittstelle vorkommen - all diese Verwendungen natürlicher Sprache fallen nicht unter natürlichsprachliche Systeme, weil sie kein Wissen über die Sprache zu ihrer Generierung oder Analyse verwenden.

> **Def.: Natürlichsprachliche Systeme** (NLS) sind Systeme, die natürliche Sprache analysieren und/oder generieren, wobei sie Wissen über Sprache verwenden.

Natürlichsprachliche Systeme können Hilfsmittel einer linguistischen Theoriebildung sein. Sie können auch Hilfsmittel der Psycholinguistik sein, die beschreiben will, wie tatsächlich im Menschen die Prozesse ablaufen, die natürlichsprachliche Äußerungen generieren oder verstehen.

Anwendungen von NLS sind:

- Dialogsysteme, die natürlichsprachlichen Zugang etwa zu Datenbanksystemen, Expertensystemen, Bildverarbeitungssystemen ermöglichen - das Spektrum reicht von Frage-Antwort-Systemen bis hin zu Beratungssystemen,
- Dialogkomponenten in multimodalen/multimedialen Systemen,

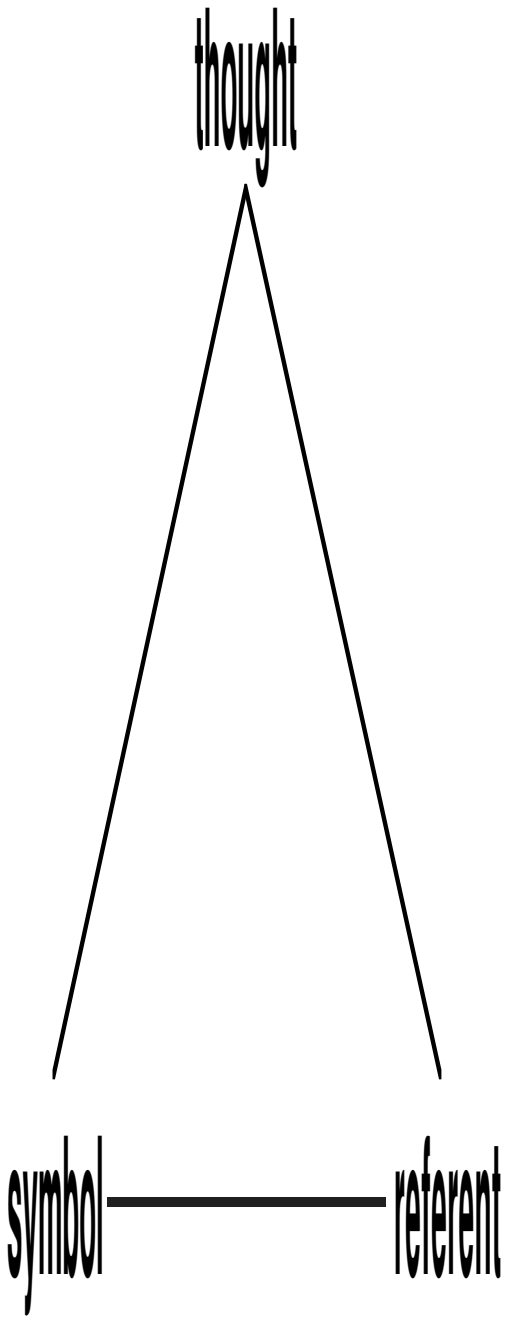
- textverstehende Systeme,
- automatische Übersetzungssysteme oder Übersetzungshilfen.

## 1.2 Aspekte der Sprache

Was ist Sprache? Diese Frage läßt sich nicht in Form einer Definition beantworten. Sprache ist unter anderem

- ein Zeichensystem,
- ein Kommunikationsmittel,
- ein Handlungssystem.

Durch die Subsummierung unter diese Oberbegriffe erbt die natürliche Sprache die kennzeichnenden Eigenschaften von Zeichensystemen, Kommunikationsmitteln und Handlungssystemen. Auf Kommunikation und (sprachliches ) Handeln werden wir noch eingehen. Mit Zeichensystemen beschäftigt sich die Semiotik. Zeichensysteme verbinden Zeichen mit Gedanken und den Sachverhalten oder Gegenständen, auf die sich die Gedanken beziehen, und dadurch indirekt Zeichen mit Sachverhalten oder Gegenständen. Dies wird meist in Form des semiotischen Dreiecks von Ogden und Richards dargestellt:




(C. G.Ogden, I. A. Richards "The Meaning of Meaning", London, 1923; deutsch: Frankfurt a.M. 1974)

Einige Eigenschaften natürlicher Sprachen, die sie nicht notwendigerweise mit anderen Kommunikationsmitteln oder Zeichensystemen teilt, sind:

- Sprache ist nicht eine Aufzählung von Äußerungen, sondern ein System, das unendlich viele Äußerungen generieren kann.
- Sprache ist arbiträr, d.h. es gibt keine natürliche oder notwendige Verbindung zwischen einem Wort oder Satz und dem, was damit ausgedrückt wird.
- Sprache wird aus Bestandteilen aufgebaut, die einzeln keine Bedeutung haben (ein Laut, ein Buchstabe bedeuten nichts).

- Alle natürlichen Sprachen sind gleich komplex - nur die Ursache der Komplexität kann verschieden sein.

- Jeder Mensch hat eine eigene Ausprägung der Muttersprache (Idiolekt), die sich stark mit der jeweiligen Gruppensprache (Soziolekt) und - etwas schwächer - mit den Varietäten der anderen Muttersprachler deckt.

Um Sprache untersuchen zu können, werden verschiedene Aspekte definiert, die unabhängig voneinander studiert werden. Die diachronische Sprachwissenschaft untersucht den Wandel der natürlichen Sprachen und stellt Verwandschaftsbeziehungen zwischen Sprachen auf. Die synchronische Sprachwissenschaft betrachtet nur die heutigen Sprachen. Diese wird wiederum in verschiedene Aspekte unterteilt. Ferdinand de Saussure unterschied

**parole** - das Sprechen, die Äußerungen, wie sie tatsächlich gesprochen und gehört werden,

**langue** - die Sprache, die Äußerungen wie sie - gemäß dem Urteil von Sprechern/Hörern für korrekt gehalten werden,

faculte de **language** - die Sprachfähigkeit, die (zu welchem Anteil?) Menschen angeboren ist.

(F. de Saussure "Cours de linguistique generale" deutsch: Grundfragen der Allgemeinen Sprachwissenschaft, Berlin, 1967[2])

Während de Saussure die Sprache (langue) von dem Sprechen (parole) unterschied, um Versprecher, Pausen, abgebrochene Sätze usw. nicht bei der Untersuchung der Sprache berücksichtigen zu müssen, trennt Chomsky Sprachkompetenz von Sprachverwendung (Performanz), wobei die Sprachfähigkeit in die Sprachkompetenz einbezogen wird:

**Kompetenz** - die Kenntnis des Sprechers/Hörers von seiner Sprache

**Performanz** - der aktuelle Gebrauch der Sprache in einer konkreten Situation

*"Für den Linguisten ebenso wie für das Kind, das die Sprache erlernt, besteht das Problem, aus den Daten der Sprachverwendung heraus das zugrunde liegende Regelsystem zu bestimmen, über das der Sprecher-Hörer verfügt und das er in der aktuellen Sprachverwendung in Gebrauch nimmt. ... Die Grammatik einer Sprache versteht sich als Beschreibung der immanenten Sprachkompetenz des idealen Sprecher-Hörers. Weiter, wenn die Grammatik durchweg explizit ist, wenn sie - mit anderen Worten - nicht auf die Intelligenz des Lesers baut, sondern wenn sie gerade eine explizite Analyse dessen liefert, was dieser von sich aus zum Verstehen der Grammatik beiträgt, dann können wir sie (...) eine* generative *Grammatik nennen."*
(N. Chomsky "Aspects of the Theory of Syntax", 1965, deutsch Frankfurt a.M. 1972, S. 14, 15)

Die faculte de language von de Saussure bezieht Chomsky in die Sprachkompetenz ein. Er geht davon aus, daß Sprachen nur erlernbar sind (und das aus verrauschten Daten), weil Menschen eine angeborene Fähigkeit zum Erlernen von Zeichensystemen und insbesondere Sprachen besitzen. Sein Ziel ist, "das System von Prinzipien, Bedingungen und Regeln" zu beschreiben, "die Eigenschaften aller menschlichen Sprachen sind", weil sie biologisch vorgegeben, angeboren sind. Dieses System, von Chomsky *Universale Grammatik* genannt, ist für ihn das "Wesen der menschlichen Sprache" (N. Chomsky "Reflections on Language" 1975, deutsch Frankfurt a.M. 1977, S. 41).

In der KI ist die Dominanz der Syntax über die anderen Ebenen der Sprache (s.u.) nie akzeptiert worden. Die enge Bindung von Kompetenz an Grammatik, wie sie bei Chomsky auch in dem angeführten Zitat deutlich wird, vernachlässigt insbesondere den Handlungs- und Kommunikationsaspekt der Sprache. Die KI strebt nicht an, eine Universale Grammatik operational zu beschreiben. Damit hat sie zugleich - bisher - den Spracherwerb aus den Betrachtungen ausgeschlossen. Die Forschungen zu maschinellem Lernen und zu natürlichsprachlichen Systemen sind in der KI - bisher - weitgehend getrennt.

### 1.3 Ebenen der Sprache

Bei allen Zeichensystemen können verschiedene Untersuchungsgegenstände unterschieden werden:

**Def.: Syntax** betrifft die Beziehung der Zeichen untereinander. Bei der natürlichen Sprache wird die Fügung von Buchstaben oder Lauten zu Wörtern und vor allem die Fügung von Wörtern zu Sätzen untersucht.

**Def.: Semantik** betrifft die Beziehung von Zeichen zu Gegenständen und Sachverhalten. Dabei wird sowohl betrachtet, unter welchen Bedingungen Aussagen wahr sind, als auch worauf die Aussagen sich beziehen.

**Def.: Pragmatik** betrifft ursprünglich die Beziehung von Zeichen zu ihren Verwendern, bei der natürlichen Sprache zu Sprecher/Hörer. Heute wird meist die Beziehung der Zeichen zu ihrem Gebrauchskontext (Sprecher, Hörer, Raum, Zeit, Vorwissen und Überzeugungen von Sprecher/Hörer) untersucht. Dabei wird betrachtet, unter welchen Bedingungen eine Äußerung angemessen ist und gelingen kann.

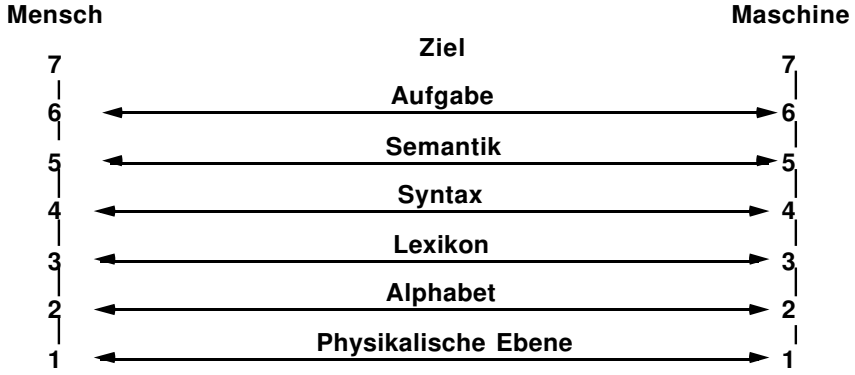
In der natürlichen Sprache trifft man auf jeder Ebene auf Ambiguitäten, d.h. Mehrdeutigkeiten, die vom Hörer aufgelöst und auch von einem NLS behandelt werden müssen. Eine syntaktische Ambiguität finden wir im Satz: Er sah das Mädchen mit dem Fernrohr. Wer hat das Fernrohr? Semantische Ambiguitäten finden wir häufig bei Wörtern (Schloß - zum Schließen oder Gebäude?), aber auch bei Sätzen: Hans will eine Künstlerin heiraten. Ist es eine bestimmte Frau, die er heiraten will, und sie ist Künstlerin? Will er irgendeine Frau heiraten, Hauptsache, sie ist Künstlerin? Pragmatische Ambiguitäten sind nicht am einzelnen Satz festzumachen. So ist z.B. der Satz Du, dich meine ich! je nach Gestik mal mehrdeutig und mal eindeutig. Ein Zettel an einer Bürotür, auf dem steht Bin in 1 Stunde zurück ist mehrdeutig, da wir den Zeitpunkt des Schreibens nicht kennen. Der Satz Hast du eine Uhr? kann durchaus wörtlich gemeint sein, obwohl er häufig eine Frage nach der Uhrzeit bedeutet.

## 1.4 Kommunikation

Sprache ist ein Kommunikationsmittel unter anderen. Unwillkürliches Verhalten wie Stimmqualität, Weinen, Lachen, Versprecher, Mimik, Sitzverteilung und interpersonale Distanz gelten insofern nicht als Kommunikationsmittel, als sie (und nur wenn sie) nicht intentional eingesetzt werden. Verhaltensweisen, die nicht kodiert werden, sondern nur für sich selbst stehen, zählen nicht zur Kommunikation. Es gibt aber Gestik und Mimik, die einem kulturspezifischen Kode entstammt und somit zur non-verbalen Kommunikation gerechnet wird wie z.B. Handschlag, Kopfnicken, Heranwinken, sich melden. Die Sprache zeichnet sich gegenüber der nonverbalen Kommunikation durch ihre Unendlichkeit aus: allen bereits generierten wohlgeformten Sätze einer Sprache läßt sich immer ein weiterer wohlgeformter Satz hinzufügen.

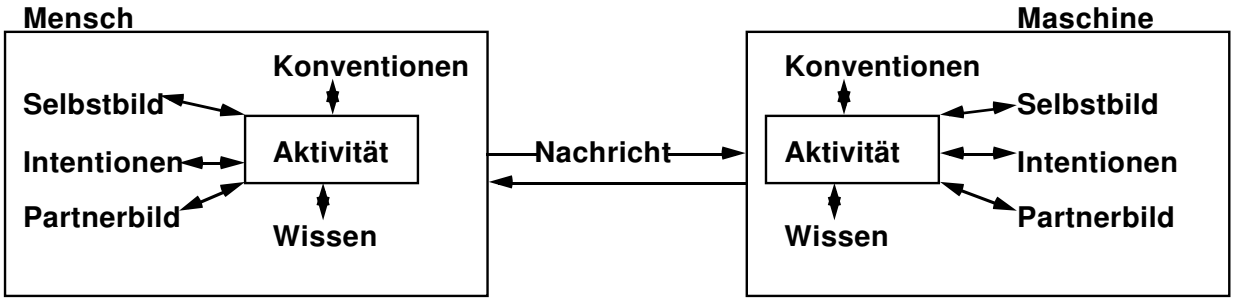
**Def.: Kommunikation** ist ein zielgerichteter, intentionaler Prozeß, bei dem ein Sender eine Nachricht, die in einem Sender und Empfänger bekannten Zeichensystem kodiert ist, über einen Kanal an einen Empfänger sendet.

Der Prozeß wird im ISO-OSI-Modell in Schichten eingeteilt, die insbesondere den Kommunikationskanal betreffen. Modelle menschlicher Kommunikation beschäftigen sich hauptsächlich mit dem Zeichensystem und inwieweit es gleichermaßen von Sender und Empfänger beherrscht sein muß (Beispiel: aktiver und passiver Wortschatz in einer Fremdsprache). Auch wird der Handlungsaspekt der Nachrichtenübermittlung ausführlich in der Pragmatik behandelt: welche Intention verfolgt der Sprecher, welche Handlung möchte er mit seiner Äußerung (Nachricht) durchführen, welche Reaktionsmöglichkeiten hat der Empfänger... Die Mensch-Maschine Kommunikation betrachtet zum einen Abstraktionsebenen wie z.B. im Modell von Nielsen ( J. Nielsen "A Virtual Protocol Model for Computer-Human Interaction", in: *Int. Journal for Man-Machine Studies*, 1986, 24, 301 - 312):




Die Zielebene bezieht sich auf die wirkliche Welt - alle anderen Ebenen beziehen sich auf Repräsentationen der Welt. Die Aufgabenebene beschreibt Objekte und Operationen, die auf der semantischen Ebene genauer gefaßt werden. Die 4. Ebene betrifft die möglichen Befehle an die Maschine bzw. Ausgaben der Maschine. Auf der 3. Ebene (die eigentlich die Syntax angibt - in der allgemeinen Bedeutung von Syntax) werden die Kommandos genauer spezifiziert. Die 2. Ebene betrifft die Umsetzung der eingegebenen Kommandos und die 1. Ebene den (Hardware-) Prozeß.

Als Kommunikationsmodell wird gern das von Susanne Maaß angeführt (S. Maaß "Why System Transparency?" in: Green et al. (eds) "The Psychology of Computer Use", London, 1983, S. 19 - 28):




Dies Modell hat allerdings den Nachteil, daß die Konventionen, das Wissen und der Kode der Nachricht nicht als wechselseitig bekannt (zumindest in Teilen) vorausgesetzt, sondern einzeln Sender und Empfänger zugeordnet wird. Dies widerspricht dem Wesen von Konventionen. Verfügen Sender und Empfänger nicht zumindest teilweise über denselben Kode, ist Kommunikation nicht möglich. Schafft eine Nachricht kein wechselseitiges Wissen, ist Kommunikation sinnlos.

Wichtig für die Kommunikation ist neben dem Kanal (z.B. Schallwellenübertragung über eine Analogleitung) und dem Kode (z.B. natürliche Sprache) auch das Medium (z.B. Telefon). Bei natürlicher Sprache ist ein großer Unterschied zwischen gesprochener und geschriebener Sprache zu beobachten. Dies ist nicht der Unterschied zwischen Kompetenz und Performanz, da sowohl ein Schreibereignis wie auch eine Äußerung beide der Performanz zuzuordnen sind. Vielmehr liegen die Unterschiede in der Tätigkeit, mit der die Äußerung vollbracht wird, sowie im räumlich-zeitlichen Verhältnis von Sprecher und Hörer begründet. Eine kleine Tabelle illustriert die Unterschiede:

| | Raum (Sprecher, Hörer) | Zeit (Sprecher, Hörer) | Tätigkeit |
|---|---|---|---|
| Gespräch (face to face) | = | = | sprechen / hören |
| Telefon | ≠ | = | sprechen / hören |
| Anrufbeantworter | ≠ | ≠ | sprechen / hören |
| privater Brief | ≠ | ≠ | schreiben / lesen |

| | | | |
|---|---|---|---|
| Geschäftsbrief | ≠ | ≠ | tippen / lesen |
| e-mail | ≠ | fast gleich? | tippen / lesen |
| Systembedienung | = | = | tippen / ausführen<br>klicken / ausführen |
| Graffiti | = | ≠ | sprayen / lesen (gucken) |

Die Tätigkeit bestimmt die Form der Äußerungen ganz wesentlich mit: mit einer Spraydose werden keine längeren Texte produziert, mit einem Textverarbeitungssystem werden keine Betonungen durch aufwärts oder abwärts geschwungene Zeilen vorgenommen. Da die meisten NLS der Systembedienung entsprechen, fehlen Markierungen, die in gesprochener Sprache bei zeitlicher Gleichheit auftreten (Pausen, Ausrufe, äh, hm, naja ...). Auch muß beachtet werden, daß das Tippen vielen Menschen nur von Geschäftsbriefen her vertraut ist, die einen bestimmten, formellen Stil aufweisen, der einem natürlichsprachlichen Auskunftssystem nicht angemessen ist.

### 1.5 Vorgehensweise beim "linguistic engineering"

Die Beschreibung natürlicher Sprache kann mit verschiedenen Zielsetzungen vorgenommen werden. Damit gehen unterschiedliche Kriterien zur Beurteilung der Beschreibung einher. Die in der Linguistik entwickelten Kriterien sind (Chomsky "Aspekte der Syntaxtheorie" 1965, deutsch Frankfurt a.M.,1972):

**Def.:** Eine Darstellung ist **beobachtungsadäquat**, wenn sie alle im Hinblick auf eine Fragestellung wichtigen Eigenschaften eines Phänomens oder zahlreicher herausgegriffener Phänomene in ihrer jeweiligen Gliederung erfaßt. Dies läßt zu, daß ad hoc für diese gerade herausgegriffenen Phänomene bestimmte Beschreibungsverfahren gewählt werden, die bei anderen Phänomenen vielleicht nicht mehr anwendbar sind.

**Def.:** Eine Darstellung ist **beschreibungsadäquat**, wenn sie sich auf ein System von Verfahren stützt, das ein ganzes Feld systematisch zusammengehöriger Phänomene zu erfassen erlaubt. Nicht nur die tatsächlich beobachteten Phänomene, sondern auch alle potentiell beobachtbaren Phänomene innerhalb dieses Feldes sollen erfaßt werden können.

*"Die Berechtigung, derart ein ganzes Feld systematisch zusammengehöriger möglicher Erscheinungen abzustecken und sich damit auf allgemeine Gesetzmäßigkeiten oder allgemeine Regeln zu berufen, ergibt sich nicht allein aus Beobachtungen und Phänomenbeschreibungen,... Sie ergibt sich aus einem bestimmten erklärenden und theoretischen Anspruch und aus der allgemeinen Problemlage, wie sie von einer Wissenschaft gegenwärtig gesehen wird."*
(Dieter Wunderlich "Grundlagen der Linguistik", Reinbek 1974, S.87f)

**Def**.: Eine Grammatiktheorie ist **erklärungsadäquat**, wenn sie auf der Basis von primären Daten (Äußerungen) eine beschreibungsadäquate Grammatik auswählen kann.

Beim *linguistic engineering* geht es -- im Gegensatz zu linguistischen Entwicklungen -- nicht darum, eine natürliche Sprache in ihrer Gesamtheit adäquat zu beschreiben, sondern darum, für eine bestimmte Anwendung eine Benutzerschnittstelle zu entwickeln, die natürlichsprachlich ist. Es geht also um die beobachtungsadäquate Darstellung eines Sprachausschnitts. Dazu muß festgestellt werden,

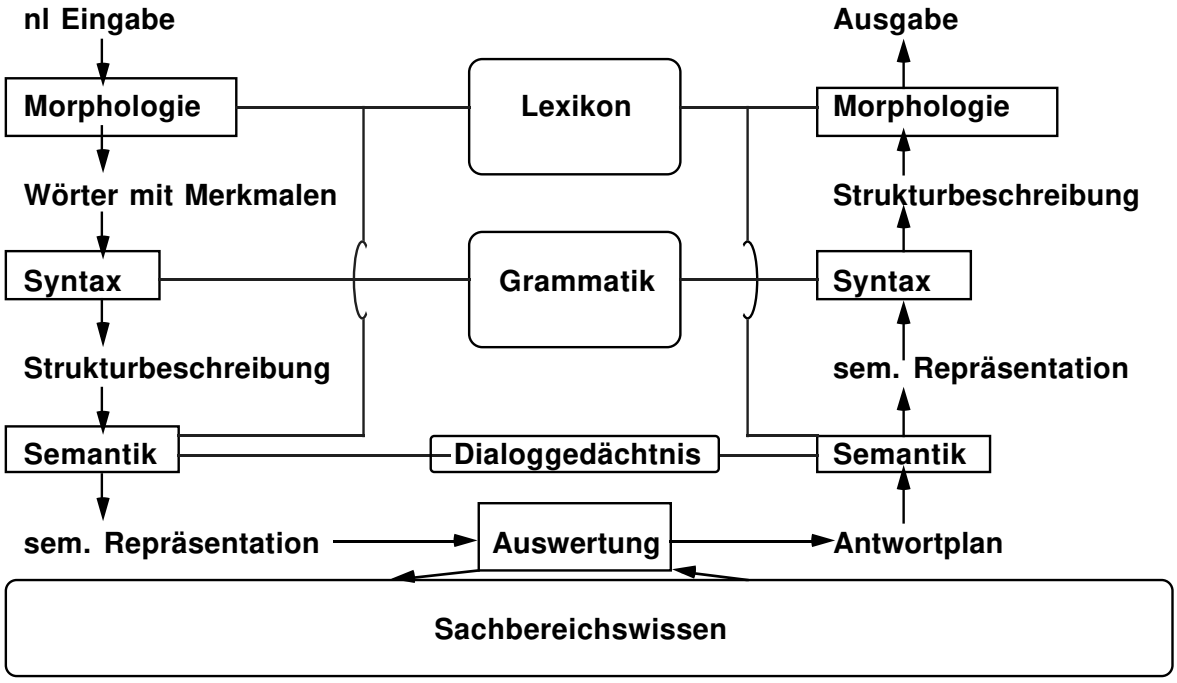
- welche Kommunikationssituation bei der Anwendung vorliegt (zeitlich-räumliches Verhältnis, anzunehmende Intentionen der Benutzer, anzunehmendes Vorwissen der Benutzer, anzunehmendes Systembild der Benutzer, Sachbereich, über den kommuniziert wird, gesprochene/geschriebene Sprache,...),
- was für Sätze in der Anwendung vorkommen können,
- welche formalen Kategorien zu ihrer Beschreibung notwendig sind,
- in welche Repräsentationssprache natürlichsprachliche Äußerungen letztlich übertragen werden sollen und

- wie diese Repräsentation vom System ausgewertet werden soll.

Dieser grobe Rahmenentwurf wird dann feiner unterteilt, die entsprechenden Komponenten entwickelt und getestet. Insbesondere empfiehlt es sich, bereits bei dem Rahmenentwurf eine Menge von Sätzen festzulegen, die später zum Testen verwendet werden. Wird die Wiederverwendung sprachverarbeitender Programme für verschiedene Anwendungsbereiche angestrebt, so wird auch das Ziel höher gesteckt: die Modellierung des Sprachausschnitts soll dann beschreibungsadäquat sein.

### 1.6 Architektur eines NLS

Die Architektur von NLS kann mit dem folgenden Bild angegeben werden, dessen einzelne Komponenten wir im folgenden genauer kennenlernen werden:




Dabei geben die Kästchen Komponenten, die nicht umrandeten Bezeichnungen Zwischenergebnisse bzw. Ein- und Ausgabe und die gerundeten Kästchen Wissensquellen an.

### 1.7 Warum Prolog?

Prolog beinhaltet einen Formalismus, DCG (definite clause grammar), der speziell für die Entwicklung von Lexikon und Grammatik entwickelt wurde. Es beinhaltet damit bereits eine eingebaute syntaktische Analysekomponente. Die Basisoperation moderner Formalismen für die Sprachverarbeitung, die Unifikation, ist als Termunifikation fester Bestandteil von Prolog. Außerdem ist Prolog leicht zu lernen - zumindest kann man sehr schnell ein Programm schreiben, das in etwa das tut, was man sich vorgestellt hat.

Ein **Prolog-Programm** besteht aus einer endlichen Menge von Klauseln. Klauseln wiederum bestehen aus einer endlichen Menge von negativen und maximal einem positiven Literal. Ein Literal ist ein einzelnes Prädikat oder die Negation eines Prädikates. Wir nennen ein Literal negativ, wenn es die Negation eines Prädikats ist, andernfalls nennen wir es positiv. Eine **Anfrage** ist eine Klausel ohne positives Literal. Die Klausel in Mengenschreibweise

{¬ mutter (X, Y), ¬ vater (Y, Z), grossvater (X, Z)}

entspricht der Formel

mutter (X, Y) & vater (Y, Z) --> grossvater (X, Z)

und wird in Prolog geschrieben

```
grossvater (X, Z) :- mutter (X, Y), vater (Y, Z).
```

Dabei heißt das positive Literal Klauselkopf, das negative Klauselkörper. Eine Anfrage wäre z.B.:

```
:- grossvater(tom,Z).
```

Desweiteren gibt es **Fakten**. Das sind Klauseln, die nur aus einem positiven Literal bestehen. Gibt man nun die Fakten ein

```
mutter (tom, lisa).  vater (lisa, udo).
```

so antwortet Prolog auf die Anfrage mit

```
Z = udo.
```

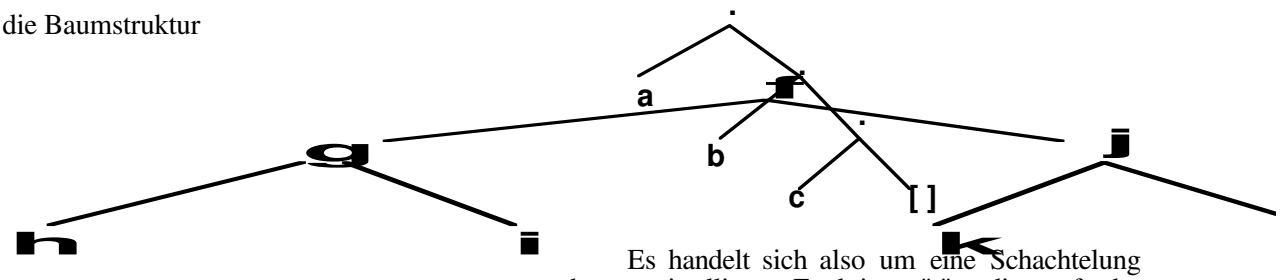
**Literale** sind positive oder negative Prädikate mit einer bestimmten Stelligkeit. Derselbe Prädikatname aber mit verschiedener Anzahl von Argumenten wird nicht als dasselbe Prädikat betrachtet. Ein Prädikat hat als Argumente Terme.

Variablen, Zahlen, Konstante und Funktionen sind **Terme**. Das sind alle Terme, wobei Zahlen auch Konstante sind. Variablen werden mit großen, Konstante wie auch Funktionen und Prädikate mit kleinen Buchstaben geschrieben. Konstante sind im System stets nur einmal vorhanden. Treten sie an verschiedenen Stellen im Programm auf, so wird an diesen Stellen stets auf ein und denselben Eintrag in der Symboltabelle verwiesen. Funktionen repräsentieren nicht den Wert eines Ausdrucks, sondern ein Datenobjekt.

Zum Beispiel steht

```
f( g( h, i), j(k, l)
```

für die Baumstruktur




Listen sind auch Funktionen:.(a, .(b, .(C, []) Sie werden aber übersichtlicher geschrieben als Liste in eckigen Klammern.

So entspricht die Liste `[a, b, c]` der folgenden Baumstruktur:

Es handelt sich also um eine Schachtelung der zweistelligen Funktion ".", die auf der obersten Ebene die beiden Argumente `a` und die Listenfunktion hat. Diese oberste Ebene wird durch die Schreibweise `[a | Rest]` deutlich gemacht.

Um Funktionen in einer angenehm lesbaren Weise notieren zu können, beispielsweise in Infix- oder Postfix-Schreibweise, kann das vordefinierten dreistelligen Prädikat op verwendet werden. Das erste Argument gibt die Präzedenz einer solchen definerten Funktion an (eine Zahl zwischen 1 und 1200). Die Präzedenz legt fest, wie Terme in Operatorschreibweise als Terme interpretiert werden, d.h. Operatoren mit kleinen Präzedenzen bilden Unterstrukturen von Operatoren mit höheren Präzedenzen. Das zweite Argument gibt den Operatyp an, dieser spezifiziert die Art des Operators (Präfix, Infix, Postfix) und die Assoziativität. Ein `x` besagt, daß alle in diesem Argument vorkommenden Operatoren eine kleinere Präzedenz als der Operator `x` haben müssen. Das letzte Argument gibt in Hochkommata das Zeichen für den Operator an. So definieren z.B. die folgenden Anweisungen eine Infixschreibweise für Addition und Multiplikation:

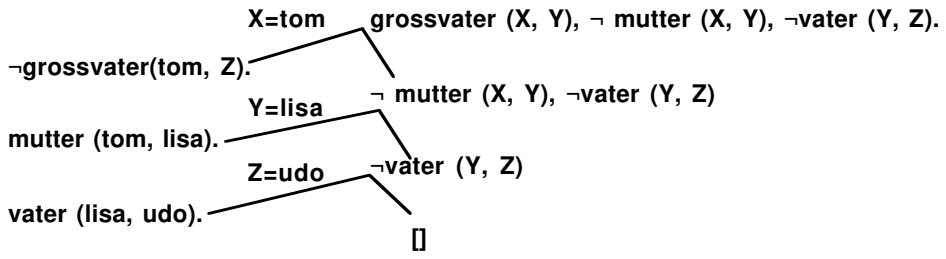
```
:- op (600, yfx, '+')    :-op (400, yfx, '*')
```

Nach dieser Anweisung kann statt `+( * ( 2,3), *(2, 4))` einfach geschrieben werden:

```
2 * 3 + 2 * 4
```

Die Verarbeitung der Programme erfolgt in Prolog nach dem Prinzip der Resolution. Die **Resolution** ist eine Schnittregel zusammen mit einer Substitution, die Variablen so unifiziert, daß geschnitten werden kann. Die Schnittregel schneidet solche Literale heraus, die bis auf das Negationszeichen übereinstimmen. Um Literale, die dasselbe Prädikatssymbol haben, übereinstimmend zu machen, werden sie unifiziert. Die **Unifikation** substituiert Terme in entsprechenden Literalen so, daß die Literale dann gleich sind. Damit das nicht zu unerwünschten Ergebnissen führt, haben verschiedene Klauseln verschiedene Variablen und für die Unifikation verwendet man den allgemeinsten Vereinheitlicher (most general unifiers). Eine Substitution s heißt **allgemeinster Unifikator** von Termen, wenn er die Terme vereinheitlicht und jeder andere Vereinheitlicher s' durch eine weitere Substitution t (also eine Spezialisierung) gewonnen werden kann: s'=

st. In unserem Großvaterbeispiel wird `X` mit `tom`, `Y` mit `lisa` und `Z` mit `udo` unifiziert. Die Schnittregel verwendet die Anfrage, um den Klauselkopf zu unifizieren und dann wegzuschneiden, und die Fakten, um die Literale im Klauselkörper zu unifizieren und dann wegzuschneiden. Bleibt kein Literal aus dem Klauselkörper übrig, ist die Resolution beendet und die Anfrage wird durch Ausgabe der Substitution beantwortet. Gelingt dies bei den gewählten Fakten / Klauseln nicht, werden die nächsten passenden Fakten / Klauseln ausgewählt (Rückziehverfahren, backtracking). Die Auswertungsstrategie von Prolog ist top-down, depth first.




Zur Einführung in Prolog empfehlen sich hier die folgenden Bücher:

Ivan Bratko"Prolog - Programmierung für Künstliche Intelligenz", Bonn, 1987

Michael C. Covington "Natural Language Processing for Prolog Programmers", Englewood Cliffs, 1994

Roland Seiffert, Thomas Stürmer "Grundlagen der KI-Programmierung", in: Görz (ed) "Grundlagen der Künstlichen Intelligenz", Bonn, 1995[2]

## 2. Syntax

Syntax bezieht sich meistens auf die Syntax von Sätzen, obwohl auch Wörter und Texte unter dem syntaktischen Aspekt untersucht werden. Eine Grammatik legt fest, welche Sätze wohlgeformt sind und welche nicht. Eine Grammatik ist ein Automat, der alle wohlgeformten Sätze erkennt und keine anderen bzw. nur wohlgeformte Sätze erzeugt und zwar alle. Da es unendlich viele Sätze gibt, muß die Grammatik rekursiv sein, wenn wir sie in endlich vielen Regeln schreiben wollen. Während bei formalen Sprachen das Alphabet E aus Zeichen besteht und die Halbgruppe E* aus Wörtern, besteht E bei natürlichen Sprachen aus Kategorien (Konstituenten) und E* aus Sätzen.

Um eine Grammatik schreiben zu können, brauchen wir also

- ein Vokabular (Alphabet), mit dem wir Kategorien (Klassen von Wörtern und umfangreicheren Satzteilen) bezeichnen können,
- Regeln, die uns angeben, wie Kategorien in Unterkategorien unterteilt werden (immediate dominance)
- Regeln, die uns angeben, in welcher Reihenfolge Kategorien auftreten können (linear precedence),

wobei die letzten beiden Punkte mit denselben Regeln dargestellt werden können. Die genaue Einordnung der natürlichen Sprachen in die Chomsky-Hierachie ist noch strittig. Diese gruppiert die Sprachklassen nach der Mächtigkeit, wobei die oberen die unteren enthalten:

rekursiv aufzählbare Sprachen (Typ 0)
#
kontextsensitive Sprachen (Typ 1)
#
indizierte Sprachen
#
kontextfreie Sprachen (Typ 2)
#

reguläre Sprachen (Typ 3)

Das Wortproblem, d.h. ob ein Ausdruck in einer Sprache enthalten ist, ist bei Sprachen des Typs 0 aufgrund des Halteproblems nicht algorithmisch lösbar. Das des Typ 1 ist zwar algorithmisch lösbar, allerdings wachsen die Kosten exponientiell. Das Wortproblem bei den Typ 2 Sprachen ist algorithmisch gut lösbar und bei den Typ 3 Sprachen sogar optimal lösbar.

Die natürlichen Sprachen können nicht mit einer regulären Grammatik beschrieben werden, so ist beispielweise im Englischen der Satz:

A doctor (whom a doctor)$^m$ (hired)$^m$ hired another nurse.

wohlgeformt, allerdings nicht mit einer regulären Grammatik zu erkennen. Natürliche Sprachen sind nicht immer kontextfrei, da Sprachkonstrukte wie $a^m b^n c^m d^n$ in einigen natürlichen Sprache möglich sind, die nicht von einer kontextfreien Grammatik erkannt werden können (G. Gazdar, C. Mellish "Natural Language Processing in PROLOG" Workingham, England,1989). Es werden nun Sprachklassen zwischen kontextfreien und kontextsensitiven Sprachen untersucht, wie z.B. die indizierten Sprachen, die rekursive Merkmale verwenden.

Da unser Interesse nicht darin besteht, eine natürliche Sprache vollständig zu beschreiben, können wir diese sprachtheoretischen Fragen hier beiseite lassen.

## 2.1 Satzteile

Was sind nun diese Kategorien, Konstituenten oder Phrasen, aus denen Sätze zusammengesetzt sind? Es gibt bestimmte Teile in einem Satz, die nur zusammen verschoben werden können (Verschiebeprobe). Diese Teile bilden eine Konstituente. Nicht wohlgeformte Sätze werden durch einen Stern markiert.

Die schwarze Katze jagt den kleinen Hasen. Den kleinen Hasen jagt die schwarze Katze.
Jagt die schwarze Katze den kleinen Hasen?

*Die schwarze jagt Katze kleinen den Hasen. *Die kleinen Hasen jagt den schwarze Katze.

Hier haben wir die drei Konstituenten: Die schwarze Katze, jagt, den kleinen Hasen. Diese Konstituenten können wir nun weiter aufteilen. Die Ersetzungsprobe setzt für jeden Bestandteil einer Konstituente etwas anderes ein und stellt fest, ob das Ergebnis ein wohlgeformter Satz ist.

Die graue Katze  Die verfressene Katze  Die Katze  Omas schwarze Katze  Die schwarze Hündin ... jagt den kleinen Hasen

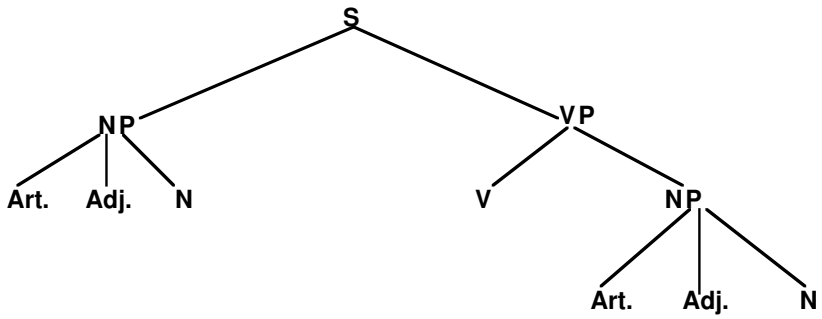
*Der schwarze Katze  *Das schwarze Katze  *Katze  *Bringen schwarze Katze ... jagt den kleinen Hasen

Wir können uns auch fragen, in welchen Kontexten jagt vorkommen kann. Auf diese Weise erhalten wir die Distribution von Teilsätzen oder Wörtern. Alle Wörter oder Teilsätze, die im selben Kontext vorkommen können, sind distributionsäquivalent. Im Beispiel sind {schwarze, graue, verfressene}, {Die, Omas}, {Katze, Hündin} distributionsäquivalent und bilden damit jeweils die Extension einer Kategorie. Durch die Ersetzungsprobe mit dem leeren Wort erfahren wir auch, daß Die Katze zusammen vorkommen muß. Wir teilen also die Konstituente Die schwarze Katze weiter auf in die Konstituenten Die Katze und schwarze. Da Die auch noch vor anderen Wörtern als Katze vorkommt, teilen wir die Konstituente Die Katze in Die und Katze. Analog verfahren wir mit den kleinen Hasen. Wenden wir Verschiebeprobe und Ersetzungsprobe auf sehr viele Beispiele, große Texte, an, so erhalten wir viele extensionale Kategorien, die sich möglicherweise überschneiden. Diese extensionalen Kategorien werden dann benannt oder durch Merkmale beschrieben. Dies wollen wir hier natürlich nicht durchführen - schließlich gibt es bereits linguistisches Wissen über Kategorien! Im Beispiel können wir {Die schwarze Katze, Die graue Katze, Die verfressene Katze, Die Katze, Omas schwarze Katze, Die schwarze Hündin,...} als Nominalphrase (NP) bezeichnen, {Die} als Artikel, {schwarze, graue, verfressene} als Adjektive, und {Katze, Hündin} als Nomen (N).

Das Beispiel zeigt, daß der Satz (S) als oberste umfassendste Konstituente oder Kategorie oder Phrase aus weiteren Kategorien besteht, die wiederum aus weiteren Kategorien bestehen, und so fort bis die nicht mehr zerlegbare Kategorie (hier: eine Wortklasse) erreicht ist. Eine Kategorie dominiert die Kategorien, in die sie zerlegt wird. Im Beispiel dominiert S die Konstituenten NP, Artikel, Adjektiv, N... Von diesen dominiert der Satz nur die NP direkt (immediate dominance).

Das Beispiel zeigt außerdem, daß manche Kategorien in fester Reihenfolge stehen: so muß der Artikel vor dem Nomen stehen, auf das er sich bezieht - egal wieviel dazwischen steht. Das Verb ist im Deutschen in Aussagesätzen die zweite Konsituente (Verbzweitstellung). Dies illustriert die *linear precedence*.

Wir erhalten als syntaktische Struktur für unseren Beispielsatz:




Hier ist die Abfolge (von links nach rechts) und die Dominanzrelation in der einen Struktur wiedergegeben. Es ist die Beschreibung für eine Fülle von Sätzen, allerdings ist sie auch noch etwas zu grobkörnig, denn diese Struktur deckt auch nicht wohlgeformte Sätze ab wie etwa

*Die grüne Idee schläft das rote Hasen.

Um solche Sätze auszuschließen, brauchen wir noch feinere Kategorien.

Jede Schulgrammatik bietet ein Vokabular an Kategorien an. Hier ist eine kurze Übersicht:

---

**Artikel** definit (bestimmt) singular: der, die, das, plural: die, indefinit (unbestimmt): ein, eine, plural: -

**Quantor** alle, einige, jeder, relativ viele, fast alle, drei, ...

**Adjektiv** singular: roter, rote, rotes, plural: roten, Komparativ: größere ..., Superlativ (und Elativ): größter...

**Substantiv (Nomen)** Eigennamen, Kontinuativa: Wasser, Luft, Gerechtigkeit, Gerundium eines Verbs: Laufen hält fit, normale N: Haus, Häuser, Maler

**Pronomen** Possessivpronomen: mein, dein, sein..., Personalpronomen: ich, mir, mich, du, dir, dich..., Demonstrativa: dieser, dieses, diese

**Verb** intransitiv (ohne Objekt): schläft, transitiv (Akkusativobjekt): jagt, ditransitiv (Akkusativ- und Dativobjekt): etwas jemandem geben, Kopula: ist, Modalverb: könnte, möchte, sollte, Hilfsverb: wird, ist, war

**Adverb** zu einem Verb: zügig, langsam, zu einem Adjektiv: sehr, etwas, zu einem Satz: freundlicherweise, leider

**Präposition** in, auf, oben, am

**Konjunktion** und, oder, obwohl, aber, weil, jedoch, indem

Weitere, feinere Kategorien werden durch Merkmale gebildet. Die klassischen Merkmale sind:

**Kasus**: Nominativ, Genitiv, Dativ, Akkusativ,

**Genus**: feminin, masculin, neutrum,

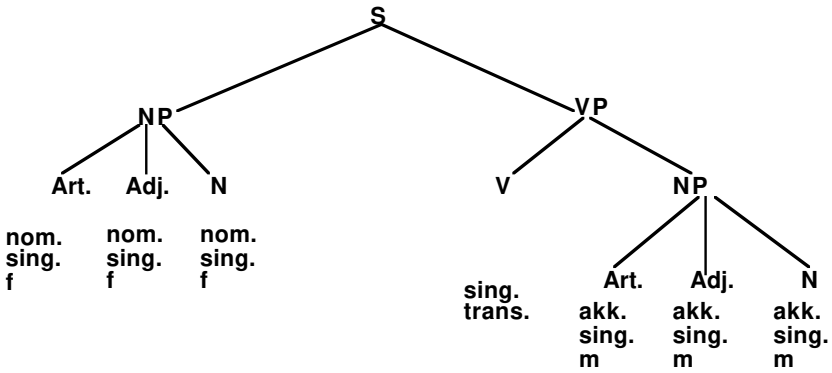
**Numerus**: Singular, Plural,

**Person**: 1., 2., 3.

---

So ist die Klasse aller Substantive im Akkusativ eine Unterkategorie (Subkategorie) von N. Ebenso bilden alle Artikel im Akkusativ eine Unterkategorie der Artikel. Allerdings werden diese Merkmale, da sie verschiedene Wortarten verfeinern, nicht als eigene, benannte Wortarten notiert. Darüberhinaus müssen Merkmale innerhalb und zwischen Konstituenten übereinstimmen: alle Wörter in einer NP haben denselben Kasus, Numerus und Genus; die NP im Nominativ und das Verb der VP haben denselben Numerus und dieselbe Person. Diese Regeln lassen sich leichter ausdrücken, wenn die Unterkategorien durch Merkmale und nicht durch eigene Kategoriennamen ausgedrückt werden. Die Subkategorisierung der Wortarten erfolgt also durch Merkmale.

Eine Strukturbeschreibung unseres Beispiels mit Merkmalen ist:




Die syntaktischen Rollen der Schulgrammatik wie Subjekt, Prädikat und Objekt sind hier durch die Dominanzrelation dargestellt: das Subjekt ist die NP, die direkt von S dominiert wird, das Objekt ist die NP, die unmittelbar von der VP dominiert wird.

## 2.2 Parsing in Prolog

**Def.:** Ein **Parser** ist ein Algorithmus, der einen Satz als Eingabe nimmt und entscheidet, ob er bezüglich einer gegebenen Grammatik wohlgeformt ist und ihm - falls ja - eine Strukturbeschreibung zuordnet.

Mit Prolog können wir leicht einen nichtdeterministischen Top-Down Parser programmieren. Ein Top-Down Parser beginnt mit dem Satzsymbol S, ersetzt dies durch NP und VP, ersetzt jene wieder usw. bis die zum Satz passende Struktur gefunden ist. Da es verschiedene Regeln gibt, die eine Konstituente in ihre Nachfolgekonstituenten überführt, sind Parser mit einer top-down Strategie nichtdeterministisch - sie müssen eine Regelauswahl rückgängig machen können und die nächste Regel probieren (backtracking). Wir brauchen lediglich die Grammatikregeln als Prolog-Klauseln zu schreiben!

**Grammatik:**

S --> NP VP

NP --> Det N

Det --> die

N --> Katze

N --> Ratte

VP --> V NP

V --> jagt

**Prolog-Klauseln:**

```
s(L1,L):- np(L1,L2),vp(L2,L).

np(L1,L):- det(L1,L2),n(L2,L).

det([die|L],L).

n([katze|L],L).

n([ratte|L],L).
```

```
vp(L1,L):- v(L1,L2),np(L2,L).                    v([jagt|L],L).
```

Dabei ist `L1` die aktuell zu verarbeitende Eingabe, `L2` der durch eine Konstituente erkannte (abgedeckte) Teil dieser aktuellen Eingabe, `L` der noch zu verarbeitende Rest der Eingabe. Die Verarbeitung eines Eingabesatzes ist geglückt, wenn `¬s(L1,[])` zum Widerspruch mit der Grammatik führt, wobei `L1` der Eingabesatz ist. Diese Listen werden auch **Differenzlisten** genannt.

### 2.2.1 Beispiel

Nehmen wir die oben angegebene Grammatik in Prolog und verfolgen den Beweispfad für den Nachweis der Wohlgeformtheit des Satzes Die Katze jagt die Ratte.

Beweisziele — zum Beweis verwendete Regeln, Fakten:

```
:- s([die, katze, jagt, die ratte],[]).          s (L1, L):- np(L1, L2), vp(L2,L).

np([die,katze,jagt,die ratte],_1),vp(_1,[])       np(L1,L) :- det(L1,L2), n(L2,L).

det([die,katze,jagt,die,ratte],_2),n(_2,_1),vp(_1,[])              det([die|L],L).

 det([die|katze,jagt,die,ratte],[katze,jagt,die,ratte]) erkannt!

n([katze,jagt,die,ratte],_1),vp(_1,[])                              n([katze|L],L).

 n([katze|jagt,die,ratte],[jagt,die,ratte]) erkannt!

vp([jagt,die,ratte],[])                              vp(L1,L) :- v(L1,L2), np(L2,L).

v([jagt,die,ratte],_3),np(_3,[])                                    v([jagt|L],L).

 v([jagt|die,ratte],[die,ratte]) erkannt!

np([die,ratte],[])                             np(L1,L) :- det(L1,L2),n(L2,L).

det([die,ratte],_4),n(_4,[])                                        det([die|L],L).

 det([die|ratte],[ratte]) erkannt!

n([ratte|[]],[])                                    n([katze|L],L). nicht erkannt!

n([ratte|[]],[])                                    n([ratte|L], L).

 n([ratte|[]],[]) erkannt! kein weiteres Ziel mehr – Eingabe wohlgeformt!
```

### 2.2.2 DCG

Die Prolog-Notation wird schnell unübersichtlich. Daher bietet Prolog eine andere Notation an, die dichter an der Schreibweise der Grammatikregeln ist:

`s --> np, vp.` oder `det --> [die].`

Eine Grammatik in Prolog ist eine **definite clause grammar** (DCG), da immer nur eine Konstituente (ein Kopfliteral) auf einmal expandiert werden kann. Wir können also keine kontextsensitiven Regeln schreiben. Die DCG-Notation ist stets von der Form:

`NT --> W`, wobei `W` eine oder eine durch Kommata getrennte Folge der folgenden Möglichkeiten ist:

ein nichtterminales Symbol NT,

ein terminales Symbol,

das leere Zeichen (Wort ) [ ],

ein Prolog-Aufruf in geschweiften Klammern `{write ('NP gefunden!')}`.

Da Prolog die DCG-Notation intern sofort in die oben angeführte Prolog-Notation überführt, sieht man beim Debugging die Prolog-Darstellung. Die DCG-Notation unterscheidet sich von der Prolog-internen Notation dadurch, daß die DCG-Notation die Liste der Eingabekette und die Liste der noch zu verarbeitenden Wörter nicht zeigt. Bei Quintus-Prolog sieht man außerdem eine leicht abweichende Darstellung bei den terminalen Symbolen, die mit der Eingabe abgeglichen werden:

`n --> [katze]` ist intern `n (L1, L) :- 'C' (L1, katze, L).`

wobei `'C'` ein Prädikat ist, das ein Element vor eine Liste hängt bzw. eine Liste in das erste Element und den Rest zerlegt. So wird `katze` und `L` zu `L1 = [katze| L]`.

**Warnung**! Es führt in eine Endlosschleife, dasselbe nichtterminale Symbol auf der linken und als erstes nichtterminales Symbol auf der rechten Seite einer Regel zu haben, d.h. in anderen Worten: links-rekursive Regeln lassen Prolog nicht anhalten. Wie aber soll man die Struktur von beispielsweise Die Katze fängt die Ratte und die Maus finden? Die Regel NP --> NP Konj NP ruft sich selbst auf ad infinitum, ohne jemals die Eingabe zu verkürzen. Hingegen ist die Regel NP --> X NP harmlos, da durch X die Eingabe verkürzt wird und dann die zweite Regelanwendung mit einer anderen Eingabe erfolgt. Covington gibt einen Ausweg an:

```
snoop(X, Y, X, Y). % Dies ist pures Prolog. Deshalb hat es zwei Argumente mehr
s--> np,vp.
np --> snoop(A,B), % als es hier, in der DCG-Form hat!
       {on(und,A),append(C,[und|D],A),append(C,D,B)}, % Hier passiert es!
       np, np.
np --> det, n.
vp --> v,np.
det --> [die].
n--> [katze].
n-->[ratte].
n-->[maus].
v-->[jagt].
```

Wir wissen ja, daß bei der Übersetzung von DCG nach Prolog zwei Argumente angehängt werden, die aktuell zu verarbeitende Eingabe A und die hinterher noch verbleibende Eingabe B. Durch die Argumentstruktur werden diese beiden Listen mit den beiden Argumenten, die in der DCG-Notation sichtbar sind, unifiziert. Mit diesem Trick haben wir auch in der DCG-Form Zugriff auf die aktuelle Eingabe (A) und können sie in einem Prolog-Programm verarbeiten. Gleichzeitig haben wir Zugriff auf die weiterzugebende Restliste (B). Das Prolog-Programm testet, ob das Wort `und` in der Eingabe vorkommt. Hier heißt dies Testprädikat `on`, es kann auch `member` heißen. `append` setzt aus zwei Listen (den ersten beiden Argumenten) eine dritte zusammen. Hier wird es benutzt, um `und` aus der Eingabekette zu entfernen. Mit der Restliste `B` wird dann weitergearbeitet. Bei der Eingabe `[die, katze, jagt, die, ratte, und, die, maus]` ist `A` zunächst der ganze Satz, `B` noch ungebunden. In dem Prolog-Programm der `np`-Klausel wird `B` mit `[die, ratte, die, maus]` unifiziert. Damit ist die Kette verkürzt und durch zweimaliges Anwenden der zweiten `np`-Klausel wird sie problemlos abgearbeitet.

### 2.2.3 Aufbau von Strukturbeschreibungen

Bisher haben wir die Grammatik nur dazu benutzt, eine Eingabe als wohlgeformt oder nicht zu erkennen. Wir wollen aber eine Strukturbeschreibung herstellen. Dazu nutzen wir die Möglichkeit von Prolog aus, mit Strukturen zu arbeiten, die noch gar nicht bekannt sind. Wir halten für die Struktur, die wir hinterher haben wollen, eine Argumentstelle frei. Zunächst steht dort nur eine Variable. Ist die Eingabe aber erfolgreich abgearbeitet, ist die Variable durch die richtigen Terme ersetzt.

Wenn das Ziel `s(s(NP,VP))` bewiesen werden soll, sind `NP` und `VP` noch nicht instanziiert. Die nächste Regel unifiziert `NP` mit `np(D,N)`, so daß das neue Ziel ist: `s(s(np(D,N),VP))`. Danach wird `D` mit `det(die)` unifiziert, `N` mit `n(katze)`, schließlich `VP` mit `vp(V, NP)` und `V` mit `v(jagt)`. Am Ende haben wir die Strukturbeschreibung als Substitutionsliste, mit der der Beweis gelang.

```
s(np(det(die),n(katze)),vp(v(jagt), det(die),n(ratte)))
```

Diese geschachtelte Struktur entspricht genau dem Baum, den wir als Strukturbeschreibung haben wollten.

**Einfache DCG-Notation:**

```
s --> np, vp.

np --> det, n.

det --> [die].

n --> [katze]

n --> [ratte].

vp --> v, np.

v --> [jagt].
```

**DCG mit Strukturaufbau:**

```
s(s(NP,VP)) --> np(NP),vp(VP).

np(np(D,N)) --> det(D), n(N).

det(det(die)) --> [die].

n(n(katze)) --> [katze].

n(n(ratte)) --> [ratte].

vp(vp(V, NP)) --> v(V), np(NP).

v(v(jagt)) --> [jagt].
```

### 2.2.4 Syntax in DCG mit Merkmalen

Die einzigen Argumente von Prädikaten, die für Satzteile stehen, waren bisher die aufzubauenden Strukturen. Damit haben wir die Merkmale noch nicht berücksichtigt, die die Kongruenz zwischen NP und VP, die beide von S dominiert werden und die Übereinstimmung innerhalb einer NP sowie den Unterschied zwischen intransitiven, ditransitiven und transitiven Verben ignoriert. Jetzt sollen Merkmale als Argumente der Konstituentenprädikate eingeführt werden. Dann können wir die Kongruenz im DCG- (oder Prolog-) Format sehr einfach ausdrücken, indem wir die Unifikation ausnutzen: soll ein Merkmal bei verschiedenen Konstituenten einer Regel denselben Wert annehmen, so wird dieselbe Variable eingetragen. Soll ein Merkmal einen bestimmten Wert annehmen, so wird dieser als Konstante eingetragen.

```
s -->  np(kasus:nom,person:3,numerus:Numerus),
       vp(person:3,numerus:Numerus,subkat:VT).

np(kasus:Kasus,person:P,numerus:Numerus) -->
                                    det(kasus:Kasus,numerus:Numerus,genus:G),
                                    n(kasus:Kasus,numerus:Numerus,genus:G).

vp(person:P,numerus:Numerus,subkat:vi) --> v(person:P,numerus:Numerus,subkat:vi).

vp(person:P,numerus:Numerus,subkat:vt)  -->  v(person:P,numerus:Numerus,subkat:vt),
                                             np(kasus:akk,person:P2,numerus:N2).

vp(person:P,numerus:Numerus,subkat:vd)  -->  v(person:P,numerus:Numerus,subkat:vd),
                                             np(kasus:akk,person:P2,numerus:N2),
                                             np(kasus:dat,person:P3,numerus:N3).

det(kasus:nom,numerus:sing,genus:f) --> [die].

n(kasus:nom,numerus:sing,genus:f) --> [katze].

v(person:3,numerus:sing,subkat:vi) --> [schlaeft].

v(person:3,numerus:sing,subkat:vt) --> [jagt].
```

Mit diesen Regeln (bei denen der Übersichtlichkeit halber der Strukturaufbau weggelassen ist - er muß vor die Merkmale eingefügt werden!) wird ausgedrückt, daß die NP unmittelbar unter S im Nominativ steht, die VP unmittelbar unter S in der dritten Person ist, NP und VP im Numerus übereinstimmen. Innerhalb der NP müssen Kasus, Numerus und Genus übereinstimmen. Die Subkategorisierung der Verbtypen ist durch das Merkmal `subkat` angegeben. Die Subkategorisierung der Nomina erfolgt hier direkt bei der Überführung in Wörter.

## 2.3 Unifikationsbasierte Grammatik

Der Einsatz von Merkmalen hält die Satzbaupläne übersichtlich und verringert die Anzahl von Kategorien und Grammatikregeln beträchtlich. Statt einer Fülle von verschiedenen Kategorien haben wir einige wenige, die aber durch Merkmale spezialisiert sind. Wenn die Merkmale einen bestimmten Wert haben, dann ist die Regel, in der dieses Merkmal vorkommt, genauso speziell wie eine Regel mit einer spezialisierten Kategorie. Andere Regeln aber, die die allgemeinere Kategorie verwenden, können direkt verwendet werden - der Merkmalswert wird durch die Unifikation an die andere Regel weitergegeben.

```
s -->  np(kasus:nom,person:3,numerus:Numerus),
       vp(person:3,numerus:Numerus,subkat:VT).

np(kasus:Kasus,person:P,numerus:Numerus) -->
                                    det(kasus:Kasus,numerus:Numerus,genus:G),
                                    n(kasus:Kasus,numerus:Numerus,genus:G).
```

Die erste Regel könnte auch lauten:

```
s --> npnom(numerus:Numerus), vp3(numerus:Numerus, VT).
```

Es könnten dann aber die zweite Regel und die entsprechenden VP-Regeln nicht direkt verwendet werden.

Wenn eine Regel (wie z. B. die erste) die Übereinstimmung eines Merkmals (hier: Numerus) bei verschiedenen Konstituenten durch Unifikation erzwingt, so ersetzt diese Regel so viele speziellere Regeln wie das Merkmal Werte hat. Merkmale sind also eine gute Idee.

Das beste an Merkmalen ist jedoch, daß sie alle mit einer einzigen Operation behandelt werden können, nämlich der Unifikation. Bei der Unifikation braucht man keine Reihenfolge zu beachten, in der Merkmale ihren Wert bekommen. Ob zuerst der Numerus der NP und dann der der VP bestimmt wird, oder umgekehrt, braucht nicht festgelegt zu werden. Wann immer `Numerus` mit einem bestimmten Wert unifiziert wird, schlägt das auf die anderen Vorkommen von `Numerus` in derselben Regel durch. Es wird eben nicht ein Wert weitergereicht, sondern gefordert, daß an verschiedenen Stellen dasselbe vorkommen soll - was immer es gerade sei. Gegenüber früheren Verfahren, die die Reihenfolge, mit der Merkmale Werte bekommen, behandeln mußten, ist dies ein großer Vorteil. Seit dem Ende der 70er Jahre gibt es deshalb verschiedene Ansätze zur Beschreibung syntaktischer Strukturen mithilfe von Merkmalen, die per Unifikation behandelt werden. Diese Ansätze werden unter dem Begriff der unifikationsbasierten Grammatik zusammengefaßt.

**Def.:** Eine **unifikationsbasierte Grammatik** ist eine Grammatik, die zur Behandlung syntaktischer Strukturen Merkmale und ihre Werte verwendet, wobei die einzige Operation, die Merkmalen Werte zuweist, die Unifikation ist.

Merkmale werden zu Strukturen zusammengefaßt. Innerhalb einer Merkmalsstruktur werden die Merkmale durch ihren Namen identifiziert. Folglich sind

{person: 2, numerus:sing} und {numerus:sing, person: 2}

gleichbedeutend.

Zwei Merkmalsstrukturen werden unifiziert, wenn die Werte desselben Merkmals sich nicht widersprechen. Die allgemeinste Unifikation zweier Merkmalstrukturen ist ihr Supremum in einem Verband, der als Bottom `fail` und direkt darüber die atomaren Merkmalsstrukturen, als Top `true` und direkt darunter die Variablen hat. Dies hat eine Reihe von Konsequenzen:

Ist ein Merkmal in einer Merkmalsstruktur nicht angegeben, so "stört" es die Unifikation nicht. Die Unifikation von

{person: 2, numerus: sing} und {numerus: X, kasus:nom} ergibt

{person: 2, numerus: sing, kasus:nom}.

Ist aber ein anderer Wert angegeben, so schlägt die Unifikation natürlich fehl. So scheitert die Unifikation von

{person: 2, numerus: sing} und {numerus: plural, kasus:nom}.

Ist dieselbe Variable bei zwei Merkmalen eingetragen, so ist die Gleichheit der Merkmalswerte erzwungen. Zum Beispiel ergibt die Unifikation von

{sem:belebt, kasus:nom} und {sem: X, obj1sem: X}

{sem:belebt, kasus:nom, obj1sem: belebt}.

Merkmale müssen nicht atomar sein, d.h. der Wert eines Merkmals kann wiederum eine Merkmalsstruktur sein.

{sem: {klasse: tier, essbar:ja}, syn: {numerus: plural, kasus:nom}}.

Wir können dann mehrere Merkmale auf einmal unifizieren:

{sem: {klasse: tier, essbar:ja}, syn: X}, {sem: Y, syn:X} ergibt

{sem: {klasse: tier, essbar:ja}, syn: X}.

Jedes Merkmal existiert genau einmal. Alle Verwendungen des Merkmals zeigen auf dasselbe Datenobjekt. Deshalb wird in linguistischen Texten oft eine Merkmalsstruktur mit einer Zahl in einem Kästchen geschrieben, wobei die Zahl mit Kästchen in verschiedenen Konstituenten auftreten kann.

### 2.3.1 Übersichtliche Repräsentation für Merkmalsstrukturen

Bisher haben wir im DCG-Formalismus die Merkmalsstrukturen als Argument der Konstituenten behandelt. Dies führt zu recht unübersichtlichen Klauseln, da alle Merkmale - auch wenn sie gar nicht benötigt werden - ihre Position in der Liste der Argumente haben. Es gibt deshalb andere Notationen wie die von PATR-II oder GULP. PATR-II schreibt die Gleichheitsbedingungen von Merkmalsstrukturen hinter die Ersetzungsregel.

S --> NP VP
<NP person> = <VP person>
<NP numerus> = <VP numerus>
<NP kasus> = nom

GULP schreibt zwischen Merkmal und Wert einen Doppelpunkt, zwischen zwei Merkmalsstrukturen zwei Punkte.

```
s --> np (person: P..numerus:X..kasus:nom), vp (person:P..num:X).
```

Eingeschachtelte Merkmalsstrukturen sehen dann so aus:

```
(sem: (klasse:tier..essbar:ja)..syn:X).
```

Die Vorteile von PATR-II oder GULP sind, daß nur diejenigen Merkmale, die auch benötigt werden, aufgeschrieben sind, und die Merkmalsstrukturen nur ein Argument der Konstituentenprädikate sind. Die GULP-Darstellung ist sehr nah an Prolog, muß aber noch in das DCG-Format übersetzt werden. Dabei werden die in einer Regel vorkommenden Merkmalsstrukturen unifiziert und in die richtige Position der Liste aller Merkmale gesetzt. Covington schlägt ein Prolog-Programm vor, das ein DCG-Programm mit Merkmalstrukturen in GULP-Notation liest, darin nach Merkmalsstrukturen sucht, und diese in die korrekte interne Darstellung überführt.

Dazu wird zunächst festgelegt, wieviele Merkmale es geben soll und in welcher Reihenfolge sie in der Prolog-Darstellung vorkommen. Nehmen wir an, wir hätten lediglich Kasus, Person, Numerus als Merkmale. Dann verwendet die reine DCG-Darstellung nachher eine Liste der Länge 3 für die Merkmalsstrukturen. Die folgenden Fakten setzen ein Merkmal an die richtige Position der internen Liste. Die produzierte Liste der Merkmale wird mit `g_` bezeichnet.

```
g_schema (kasus:X, [X,_,_]).

g_schema(person:X, [_,X,_]).
```

```
g_schema(numerus:X, [_,_,X]).
```

Natürlich müssen die Infix-Operatoren deklariert werden:

```
:- op(600, xfy,':').

:- op(600, xfy,'..').
```

Die Übersetzung von GULP-Ausdrücken erfolgt unter Verwendung von `g_schema` durch `g_translate`.

```
% g_translate (+Merkmalsstruktur, - g_ (List))
% übersetzt Merkmalsstrukturen in die interne Repräsentation g_(List)

% Fall 1: Merkmalsstruktur ist Variable oder atomar

g_translate(X,X) :- (var(X) ; atomic(X)), !.

% Fall 2: Merkmalsstruktur ist ein Merkmal-Wert-Paar

g_translate(M: W,g_(Liste)) :-  !,
 g_translate(W,NeuW),
 g_schema(M:NeuW, Liste).

% Fall 3: Eine Folge von Merkmal-Wert-Paaren

g_translate(First..Rest,g_(Liste)) :- !,
 g_translate(First,g_(Liste)),
 g_translate(Rest,g_(Liste)).

% Fall 4: Eine Struktur

g_translate(Struktur, Ergebnis) :-
 Struktur = ..[Funktor|Args], !,
 g_translate_aux(Args,NeuArgs),
 Ergebnis = .. [Funktor|NeuArgs].

% Hilfsprädikat, das alle Elemente einer Liste abarbeitet.

g_translate_aux([T|Terme], [NeuT|NeuTerme]) :-
 g_translate_aux(T,NeuT),
 g_translate_aux(Terme, NeuTerme).

g_translate_aux([],[]).
```

Wir können nun also übersichtlichere Regeln schreiben wie z.B.

```
s(s(NP,VP))-->np(NP,person:P..numerus:X..kasus:nom),vp (VP,person:P..numerus:X).
```

und übersetzen diese in das normale DCG-Format:

```
s(s(NP,VP))-->np(NP,g_([P,X,nom])),vp (VP,g_([P,X,_])).
```

## 3. Lexikon

Wir haben in den Beispielen für die syntaktische Analyse den speziellsten (untersten) Kategorien direkt ein Wort zugeordnet.

```
v(v(schlaeft),person:3..numerus:sing..verbtyp:vi) --> [schlaeft].
```

Diese Regel ist ein Lexikoneintrag, wenn auch ein sehr einfacher. Im folgenden betrachten wir die Lexikonkomponente genauer.

**Def.:** Ein **Lexikon** ist eine Liste von Wörtern, die jedem Wort bestimmte Eigenschaften zuordnet. Die Aufgaben des Lexikons umfassen je nach Anwendung: Definition aller möglichen Wörter, Zuordnung von Eingabezeichen zu Wörtern und/oder Bereitstellung der notwendigen Information für Syntax (und Semantik).

Eine Möglichkeit ist die, die wir oben gewählt haben:

**Def.:** Ein Lexikon, das alle Wortformen enthält, wird als **Vollformenlexikon** bezeichnet. Es ordnet den Wortformen direkt alle Merkmale zu.

Ein durchschnittliches Wörterbuch enthält ca. 100 000 Wörter, wobei keine flektierten Wörter enthalten sind. Sollen auch alle möglichen Varianten eines Wortes enthalten sein, beispielsweise neben spielen auch spiele, spielst, spielt usw., so vervielfacht sich ihre Anzahl. Dabei haben sehr viele Wörter eine regelhafte Flexion. So können wir z.B. regelmäßige Verben zerlegen.

spiel+e, spiel+st, spiel+t, spiel+en, ge+spiel+t, spiel+te, spiel+test, spiel+te, spiel+ten, spiel+tet

Stamm, Suffix (Endung) und Präfix (z.B. ge) sind hier die kleinsten bedeutungstragenden Einheiten.

**Def.: Morpheme** sind die kleinsten bedeutungstragenden Einheiten eines Wortes.

**Def.: Morphologische Analyse** heißt der Prozeß, der Wörter in ihre Morpheme zerlegt. Im weiteren Sinne bezeichnet die morphologische Analyse die Zerlegung von Wörtern in Bestandteile.

Während die Flexionsmorpheme selbst keine selbständige Einheit darstellen, sind den Wortstämmen spezielle, über die Grammatik hinausgehende Bedeutungen zugeordnet.

**Def**.: Ein **Lexem** ist eine selbständige lexikalische Einheit. Hier sind vor allem die Stämme eines Wortes gemeint.

Interessant (und schwierig zu behandeln!) sind im Deutschen die zusammengesetzten Verben, wie etwa ausspielen. Hier wird aus im normalen Aussagesatz an das Ende der VP gesetzt. Das Partizip enthält jetzt ge in der Mitte, als Infix: aus+ge+spiel+t. Der Infinitiv mit zu ist bei diesen Verben in das Wort hineingezogen: aus+zu+spiel+en.

Auch Substantive können zerlegt werden: Kind, Kind+es, Kind+e, Kind+er, Kind+ern. Hier sind der Stamm (Kind) und die Suffixe, die Numerus und Kasus anzeigen, die kleinsten bedeutungstragenden Einheiten des Wortes.

Bei einigen Wörtern wird die Flexion durch einen Vokalwechsel realisiert: Haus, Häus+er, Maus, Mäus+e. Bei Verben kommt ein Vokalwechsel häufig vor: nehm+en, nahm, ge+nomm+en.

Manchmal ist die Flexion nicht durch hinzugefügte Zeichen realisiert. So ist der Plural von Maler immer noch Maler. Wir fassen dies auf als Maler+0.

Speziell im Deutschen besteht das Problem der zusammengesetzten Wörter (Komposita). Während das Englische die Wortgrenzen beibehält und komplexere NPs konstruiert, werden die Substantive im Deutschen zu einem Wort zusammengefügt. Wir haben also nicht nur die Flexionsmorpheme abzutrennen, sondern obendrein möglicherweise verschiedene Lexeme zu separieren.

Schwein+e+schnitzel, Bahn+hof+s+halle

Hier haben wir einmal das Lexem Schwein und dann das Lexem schnitzel, die mit dem Morphem für Plural verbunden sind, sowie die drei Lexeme Bahn, hof und halle, wobei Bahn und hof enger zusammengehören als das mit dem Morphem für den Genitiv angeklebte halle. Es macht manchmal einen Unterschied, ob ein Genitiv-Morphem anwesend ist, oder nicht:

Land+s+mann, Land+mann

Manche Buchstaben scheinen nur für das leichtere Sprechen eingefügt. Wie erklärt sich das g in

ge+g+essen ?

Es ist eine Frage, wie fein wir unterteilen wollen. Bei der feinen Zerlegung in Morpheme müssen die Merkmale, die den Morphemen zugeordnet sind, so verknüpft werden, daß die Merkmale für das gesamte Wort herauskommen. Gerade bei semantischen Merkmalen ist das kein leichter Prozeß, denn es gibt viele verschiedene Relationen (für, aus, von,...) zwischen den Lexemen:

Schweineschnitzel - Schnitzel aus Schwein(efleisch), Jägerschnitzel - Schnitzel für Jäger

Welches ist die Relation zwischen Bahn und Hof? Welche Grundbedeutung für diese Lexeme müssen wir annehmen, damit ihre Fügung Sinn macht? Vermutlich ist das Wort Hof für Gehöft, Garten, Bezirk, Tempel (mit Dach), abgeleitet aus der Grundbedeutung Anhöhe, hier verallgemeinert worden zu besonderer Platz. Ein Bahnhof ist also ein besonderer Platz für die Bahn. Soll vielleicht lieber Bahnhof als ein Lexem behandelt werden? Schließlich verstehen die meisten Bahnhof direkt als einen bestimmten Ort, an dem Züge halten, um Passagiere ein- und aussteigen zu lassen. Problematisch ist auch die Grundbedeutung von Morphemen wie be oder ver. Was bewirkt die Zusammensetzung von ver und antwort in verantworten oder Verantwortung? Wenn wir diese Probleme ausklammern wollen, dann werden wir etwa Verantwortung+s+bewußt+sein der Zerlegung in Ver+antwort+ung+s+ be+wußt+sein vorziehen.

Ein weiteres Problem sind die getilgten Buchstaben bei Komposita. Ein Beispiel ist Schif(f)+fahrt. Durch Zerlegen des Wortes kommen wir hier nicht zum Ergebnis. Wir müssen berücksichtigen, daß derselbe Konsonant hier nicht dreimal hintereinander vorkommen darf. Allerdings soll die Rechtschreibreform gerade diese Regel abschaffen, so daß dann das leicht zerlegbare Wort Schifffahrt korrekt ist.

Wenn wir kein Vollformenlexikon benutzen wollen und nicht an der Entscheidung, was ein mögliches und was ein unmögliches Wort des Deutschen ist, interessiert sind, dann können wir unsere Definition des Lexikons spezialisieren:

> Eine Lexikonkomponente soll uns für ein Wort, wie es in einer Äußerung vorkommt, (genauer: für eventuell auch verstreute Teile eines Wortes) eine Zerlegung liefern und dann für jeden Teil der Zerlegung die für Syntax und Semantik wichtigen Merkmale angeben.

Wir haben keinen Prozeß vorgesehen, der die Wortbildung über die Flexion hinaus behandelt. Damit haben wir uns dafür entschieden, nur Lexeme und Flexionspartikel als Bestandteile eines Wortes zu betrachten und nicht alle Morpheme, in die das Wort zerlegbar wäre. Ein gängiges Kompositum wird hier als ein Lexem behandelt. Dies ist nicht angemessen, wenn wir Zeitungstexte analysieren wollen. Dort sind etwa 30% der Wörter Namen oder spontane Komposita! Es ermöglicht uns aber, eine direkte Zuordnung von syntaktischen/semantischen Merkmalen und den Teilen, in die die Lexikonkomponente ein Wort zerlegt hat, vorzunehmen. Wir haben hier zwei Prozesse vorgesehen:

- Eine morphologische Analyse, die ein Wort in ein Lexem und seine Flexionsmorpheme zerlegt, und
- ein Lexikon, das Lexemen und Flexionsmorphemen syntaktische und semantische Merkmale zuordnet.

Für die Realisierung als Programm müssen wir noch beachten, daß ein Wort nicht direkt zerlegt werden kann, sondern daß nur eine Folge von Buchstaben zu Lexemen und Morphemen zusammengefaßt werden kann. Als erstes brauchen wir also

- einen Prozeß, der aus einem Wort eine Folge von Buchstaben macht.

Da Covington (1994) ein Prolog-Programm für die Überführung von Wörtern in Buchstaben angibt, gehe ich auf diesen Prozeß hier nicht ein.

### 3.1 Morphologische Analyse

Nehmen wir an, wir hätten die Wörter in ihre Buchstaben zerlegt und zwischen ihnen das Zeichen "-" als Trennsymbol eingefügt. Wir können dann den DCG-Formalismus auch dazu verwenden, Wörter in Stamm und Suffix zu zerlegen. Hier ist ein sehr einfaches Beispiel, das die Vorgehensweise illustriert:

```
% DCG für Lexikonkomponente
% Aufruf mit
```

```
% phrase(s(S), [d,u,-,f,l,i,e,g,s,t,-]).
%
% Operatordeklaration
:- op(500,xfx,':').

% wort überführt in Stamm und Suffix
% ohne Merkmalsstrukturen
wort(Stamm, Kat, Suffix, pers:P,num:Num,temp:T)-->
      stamm(Stamm,Kat),
      suffix(Suffix, pers:P,num:Num,temp:T).

% stamm überführt Stamm in Buchstabenfolge
stamm(flieg, vi) --> [f,l,i,e,g].

% suffix ordnet einem Flexionspartikel Merkmale zu
suffix(e, pers:1, num:sing, temp:praes)-->[e, -].
suffix(st, pers:2, num:sing, temp:praes)-->[s,t, -].

% Zum Testen nötige Grammatik
s(s(NP,VP)) -->
      np(NP, pers:P, num:Num, kas:nom),
      vp(VP, pers:P,num:Num,temp:T).

      np(np(N), pers:P, num:Num, kas:nom) -->
      pron(N, pers:P, num:Num, kas:nom).

vp(vp(V), pers:P,num:Num,temp:T) -->
      v(V,pers:P,num:Num,temp:T).
v(v(Stamm, Suffix),pers:P,num:Num,temp:T)-->
      wort(Stamm, Kat, Suffix,pers:P,num:Num,temp:T).

% Zum Testen nötige zusätzliche Lexikoneinträge
% Dies ist nur ein Hack!
pron(pron(ich), pers:1, num:sing, kas:nom) --> [i,c,h, -].
pron(pron(du), pers:2, num:sing, kas:nom) --> [d,u, -].
```

Die Analyse von `[d,u,-,f,l,i,e,g,s,t,-]` liefert als Ergebnis

```
s(np(pron(du)),vp(v(flieg,st))).
```

Selbst dieses kleine Beispiel zeigt bereits einige wesentliche Punkte der Wortzerlegung.

Im Gegensatz zum Vollformenlexikon sind hier Informationen, die für eine Klasse von Wörtern (hier: Verben eines Flexionstyps) gleich sind, nur einmal angegeben. Wir haben die Flexionssuffixe zu einer Tabelle zusammengefaßt, die als Regeln mit dem Prädikat `suffix` gespeichert sind. Entsprechende Einträge für die anderen Zeiten müssen noch erstellt werden. Da nicht alle Verben dieselben Suffixe haben, muß das Prädikat `suffix` als weiteres Argument die Angabe einer Flexionsklasse erhalten. Die Flexionsklasse gehört dann in den Lexikoneintrag des Verbstammes. Die wort-Regel stellt die Übereinstimmung der Flexionsklasse in `stamm` und `suffix` her.

Die Trennung zwischen Lexikoneinträgen, morphologischer Analyse und Parsing muß nicht durch einen Wechsel des Formalismus' angezeigt werden. Wir haben hier stets den DCG-Formalismus verwendet. In dem Beispiel stellen `suffix-`, `stamm-` und `pron`-Regeln die Lexikoneinträge dar. Die `wort`-Regel ist die morphologische Analyse.

Die Merkmale, die im Lexikon für eine bestimmte Wortform ihren Wert erhalten (hier: durch `suffix`), werden an die Grammatik hochgereicht (hier: durch die wort-Regel). Wir haben hier dem Wortstamm nur ein Merkmal zugeordnet. Wie bereits das Beispiel der Pronomen zeigt, müssen auch Lexemen Merkmale zugeordnet werden. Grammatik und Lexikon müssen also hinsichtlich der Merkmale abgestimmt werden: alle von der Grammtik benötigten Merkmale müssen im Lexikon bereitgestellt werden. Darüberhinaus müssen die Merkmale im Lexikon mit der semantischen Komponente abgestimmt werden. Das Lexikon vermittelt sowohl syntaktische als auch semantische Informationen.

Stellen wir uns eine größere Grammatik und ein umfangreicheres Lexikon (`suffix` und `stamm`) vor, so können wir ein Problem der top-down Strategie von Prolog erkennen: erst beim Wort werden Merkmale mit konkreten Werten versehen, die dann die Anwendung bestimmter Grammatikregeln ausschließen. Dies führt

zu beträchtlichem Backtracking! Hätten wir einen schnelleren Zugang zur Information des Lexikons (wie bei bottom-up Parsern), könnten wir gleich auf einen Ausschnitt der Grammatikregeln fokussieren.

Die Zerlegung erfolgt über die Unifikation in den `suffix`- und `stamm`-Regeln. Es kann verschiedene mögliche Zerlegungen geben. Obendrein kann dasselbe Suffix mit verschiedenen Merkmalen vorkommen. Auch dies führt zu Backtracking - jetzt im Lexikon.

Das kleine Beispiel behandelt weder durch Infix markierte Formen noch den Vokalwechsel. Das Verb fliegen im Beispiel weist überdeutlich darauf hin: schließlich heißt der Stamm im Imperfekt flog!

### 3.2 Lexikoneinträge

Lexikoneinträge ordnen Morphemen oder Wörtern Merkmale zu. Diese Merkmale werden von Grammatik (und Semantik) gebraucht. Wir haben gesehen, daß Merkmale zum einen die Flexionsart betreffen, zum anderen den Kontext, in dem ein Wort stehen kann. Der Zusammenhang zwischen Grammatik und Lexikon wird vielleicht am deutlichsten bei der Subkategorisierung der Verben. Jedes Verb hat eine Nominalphrase im Nominativ als Subjekt. Darüber hinaus gibt es obligatorische und fakultative Ergänzungen sowie freie Ergänzungen. Obligatorische Ergänzungen müssen in jedem Falle im Satz vorkommen, wenn das betreffende Verb vorkommt. Die Ergänzungen des Verbs werden auch der **Verbrahmen** genannt.

Ich erwerbe ein Haus. * Ich erwerbe.

Fakultative Ergänzungen gehören zu dem Verb, müssen aber nicht vorkommen.

Ich berichte. Ich berichte ihm. Ich berichte ihm mein Erlebnis.

Ich berichte ihm über meine Reise. Ich berichte ihm von meiner Reise.

Ich berichte ihm, wer es gewesen ist.

Hier sind die NP im Dativ, die NP im Akkusativ und die Präpositionalphrase bzw. der eingebettete Satz fakultative Ergänzungen für berichte.

Freie Ergänzungen sind nicht spezifisch für ein bestimmtes Verb. So kann am Morgen - jedenfalls unter syntaktischen Gesichtspunkten - mit jedem Verb im selben Satz vorkommen. Wir können nun für jede Subkategorie der Verben eigene Grammatikregeln schreiben.

```
%intransitives Verb
vp(person:P..numerus:Numerus..subkat:(typ:vi))-->
 v(person:P..numerus:Numerus..subkat:(typ:vi)).

%transitives Verb
vp(person:P..numerus:Numerus..subkat:(typ:vt))-->
 v(person:P..numerus:Numerus..subkat:(typ:vt)),
       np(kasus:akk..person:P2..numerus:N2).

%ditransitives Verb
vp(person:P..numerus:Numerus..subkat:(typ:vd))-->
 v(person:P..numerus:Numerus..subkat:(typ:vd)),
       np(kasus:dat..person:P3..numerus:N2),
       np(kasus:akk..person:P2..numerus:N3).

%Dativverb
vp(person:P..numerus:Numerus..subkat:(typ:dv))-->
 v(person:P..numerus:Numerus..subkat:(typ:dv),
 np(kasus:dat..person:P3..numerus:N2).

%Satzergänzung - mit Markierung, daß der Satz eingebettet ist.
vp(person:P..numerus:Numerus..subkat:(typ:vs) -->
 v(person:P..numerus:Numerus..subkat:(typ:vs)),
 np(kasus:dat..person:P3..numerus:N2),
 s(compl:1).
```

```
%mit Präpositionalphrase
vp(person:P..numerus:Numerus..subkat:(typ:pv..prep:PR))-->
 v(person:P..numerus:Numerus..subkat:(typ:pv..prep:PR)),
 np(kasus:dat..person:P3..numerus:N2),
 pp(prep:PR..kasus:K).

pp(prep:PR..kasus:K)-->
pr(prep:PR..kasus:K),
np(kasus:K..person:P..numerus:N).
```

Im Lexikoneintrag für den Verbstamm wird angegeben, zu welcher Klasse das Verb gehört. Wird ein Verb unterschiedlich verwendet (wie etwa `berichten`), so gibt es mehrere Lexikoneinträge. Auch können Merkmale eingesetzt werden, um die Präpositionen anzugeben, die zu einem Verb gehören. Hier habe ich die Präsposition nicht über eine `wort`-Regel oder als Stamm angegeben, da es hier nur darum geht, zu illustrieren, wie Grammatik und Lexikon bei der Subkategorisierung von Verben zusammenwirken.

```
stamm(bericht, subkat: (typ:pv..prep:ueber)--> [b,e,r,i,c,h,t].
stamm(bericht, subkat:(typ:pv..prep:von)--> [b,e,r,i,c,h,t].
pr(prep:ueber..kasus:akk)--> [ü,b,e,r].
pr(prep:von..kasus:dat)--> [v,o,n].
```

Wir können aber auch die Subkategorie des Verbs als Merkmalsstruktur schreiben und im Lexikon als Werte dieser Merkmalsstruktur eine Liste aller fakultativen und obligatorischen Ergänzungen angeben. Dies bedeutet, daß wir nicht mehr die Kategorie als Prädikat, sondern als Merkmal schreiben. In der Grammatik schreiben wir eine Variable für diese Merkmalsstruktur. Im Lexikon wird nun bei jedem Verb angegeben, welche Ergänzungen es haben kann.

```
p(kat:(typ:vp)..subkat:Y)--> p(kat:(typ:vp)..subkat:[X|Y], p(kat:X).

p(kat:(typ:s)) --> p(kat:(typ:np..kasus:nom)), p(kat:(typ:vp)..subcat:[]).
```

Lexikoneinträge für berichten sehen dann etwa so aus[1]:

```
p(kat:vp..subkat:[])-->[bericht]. %intransitive Verwendung

p(kat:vp..subkat:[(typ:np..kasus:dat)])-->[bericht]. % als Dativverb

                                                      %ditransitive Verwendung
p(kat:vp..subkat:[(typ:np..kasus:dat),(typ:np..kasus:akk)])-->[bericht].

p(kat:vp..subkat:[(typ:np..kasus:dat),(typ:s)])-->[bericht]. % mit Satzergänzung
```

Die erste Regel entspricht einer Regel VP --> VP, X. Dabei wird das X nach und nach an die Elemente der Subkategorisierungsliste des Verbs gebunden. Diese Liste enthält hier direkt die Ergänzungen des Verbs. Dies geschieht, bis die Liste leer ist. Erst dann wird die zweite Regel, die S --> NP, VP entspricht, angewandt. Der Lexikoneintrag für das Verb bietet gleich den ganzen Verbrahmen, nicht nur das Verb selbst. Es ist also ein Eintrag für eine Verbalphrase.

**Achtung!** Dies Verfahren führt bei einem Parser mit top-down Strategie in eine Endlosschleife! Wenn wir Prolog weiterhin als unseren Parsing-Formalismus verwenden wollen, können wir diese Alternative nicht wählen. Sie ist angemessen, wenn wir einen bottom-up Parser implementieren.Das können wir natürlich auch in der Programmiersprache Prolog tun.

### 3.2.1 Lexikonzugriff

Wie speichern wir all die Wörter und wie greifen wir möglichst effizient auf sie zu? Wenn wir stets die gesamten Wortstämme speichern, so ist sicherlich der Zugriff relativ schnell, wenn das Wort das erste Argument eines Prädikats ist, weil Prolog auf die erste Argumentposition indexiert. Aber jedes Wort ist ein

[1]Da ich nur das Prinzip der Interaktion zwischen Grammatik und Lexikon deutlich machen möchte, habe ich hier die Umsetzung in eine Buchstabenfolge ebenso wie den Aufbau der Strukturbeschreibung weggelassen.

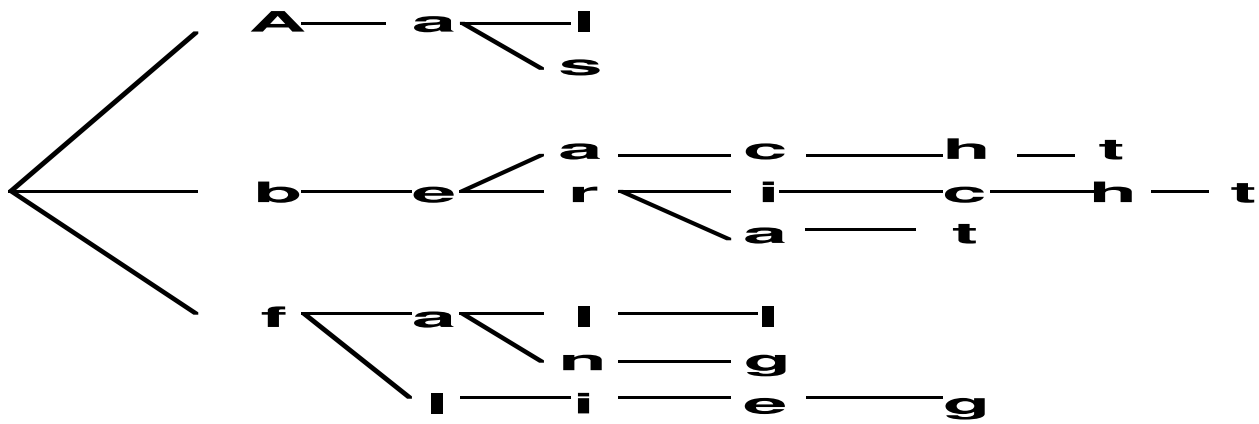
Eintrag in der Symboltabelle von Prolog. Da bis zum Auffinden des richtigen Wortes und der richtigen Struktur auch viele Fehlversuche von Prolog unternommen werden und es zu Backtracking kommt, wird die Symboltabelle mit den Wörtern all der fehlgeschlagenen Versuche überfüllt. Daher ist die Repräsentation, die wir oben angeführt haben,

```
stamm(bericht, subkat: (typ:pv..prep:ueber)--> [b,e,r,i,c,h,t].
```

nicht praktikabel für ein realistisch großes Lexikon. Stattdessen verwenden wir einen sogenannten ltree. Dieser Buchstabenbaum ist so tief wie die größte Anzahl von Buchstaben eines Wortes und so breit wie die Anzahl der Buchstaben. Dabei kann er alle Wörter einer Sprache aufnehmen.

**Def.:** Der **Buchstabenbaum** (ltree) ist ein Baum, dessen Kanten Buchstaben so verbinden, daß ein Pfad durch den Baum zu einem Lexem führt.

Als kleiner Ausschnitt aus den Wortstämmen sei hier der Buchstabenbaum für die Stämme der Wörter Aal, Aas, berichten, beraten, beachten, fallen, fangen, fliegen gezeigt:




Als Blätter können dann die Lexikoneinträge mit den Merkmalsstrukturen der Wortstämme angefügt werden. Ein Buchstabenbaum ist eine Liste von Zweigen. Jeder Zweig ist wiederum eine Liste. Wir schreiben den bildlich dargestellten Baum also als:

```
ltree( [ [A, [a,   [l, Aal], [s, Aas]]],

         [b, [e,   [a, [c, [h, [t, beacht]]]],

                   [r,   [a, [t, berat]],

                         [i, [c, [h, [t, bericht]]]]],

         [f, [l,   [i, [e, [g, flieg]]],

             [a,   [n, [g, fang]],

                   [l, [l, fall]]]] ]).
```

Covington gibt ein Prolog-Programm an, das diese Datenstruktur verwendet, um zu einem Lexikoneintrag zu gelangen. Ich habe hier lediglich das Lexem selbst als Lexikoneintrag angegeben. Tatsächlich werden an der Stelle natürlich die Merkmalsstrukturen angegeben.

## 4. Semantik

Bisher haben wir lediglich die Struktur eines Satzes formalisiert. In diesem Kapitel geht es um die Bedeutung von Sätzen.

- Auf der Grundlage der Strukturbeschreibung einer Eingabekette und bestimmter Angaben im **Lexikon** können wir eine semantische Beschreibung herstellen.

- Diese kann möglicherweise Ambiguitäten enthalten. Zur Auflösung von Mehrdeutigkeiten (Disambiguierung) wird der sprachliche Kontext sowie die Äußerungssituation hinzugezogen. Die Erfüllbarkeit der semantischen Beschreibung wird über dem **begrifflichen Wissen** getestet. Auch wird das begriffliche Wissen verwendet, um die semantische Beschreibung zu ergänzen.

- Schließlich soll die semantische Beschreibung ausgewertet werden, indem sie auf das aktuelle **Weltwissen** bezogen wird.

Dies ist der Ablauf bei der Analyse von Sätzen. Er muß keinesfalls strikt sequentiell abgearbeitet werden! Bei der Generierung wird zunächst aufgrund des Weltwissens und der Äußerungssituation eine semantische Beschreibung erstellt, die dann in eine Strukturbeschreibung überführt wird. Auch ist die Gliederung mehr eine den Gegenstandsbereich der Semantik strukturierende als ein tatsächlicher Ablauf von verschiedenen Systemkomponenten.

Mit der Bedeutung von Sätzen hat sich die Logik seit Jahrhunderten auseinandergesetzt. Allerdings ging es nicht darum, den Prozeß zu beschreiben, der eine Äußerung (oder ihre Struktur) in eine semantische Beschreibung überführt. Dieser Prozeß blieb allein dem "Logik-Anwender" überlassen. Vielmehr wurde ein Satz direkt mit seiner logischen Darstellung identifiziert und es ging um die Bedeutung dieses Ausdrucks in logischer Darstellung. Was ist Bedeutung? Gottlob Frege hat den Unterschied zwischen Sinn und Bedeutung eingeführt.

**Def.:** Die **Bedeutung** eines Ausdrucks ist seine Interpretation. Die Interpretation bildet
- eine Konstante auf ein Element aus dem Universum (universe of discourse),
- ein Prädikat auf ein Tupel, gebildet aus Elementen des Universums,
- eine Formel auf einen Wahrheitswert ab.

**Def.:** Der **Sinn** einer Äußerung ist das, wovon sie tatsächlich handelt.

Wenn wir dem Satz Hoffentlich regnet es morgen nicht keinen Wahrheitswert zuordnen können, so hat er keine Bedeutung -- sehr wohl aber Sinn! Derselbe Unterschied wird durch das Begriffspaar ausgedrückt:

**Def.:** Die **extensionale Bedeutung** (Referenz) eines Ausdrucks ist das Objekt, auf den er sich bezieht, oder (bei Sätzen / Formeln) der Wahrheitswert . Das Objekt ist der **Referent** oder das Referenzobjekt.

**Def.:** Die **intensionale Bedeutung** eines Ausdrucks ist eine Beschreibung oder ein qualitativer Sachverhalt.

Die Intension entspricht dem Sinn, die Extension entspricht der Bedeutung. Ein Beispiel mag den Unterschied verdeutlichen:

der Mörder von Schmitt drückt intensional aus, daß jemand den Tod von Schmitt vorsätzlich verursacht hat, wobei wir wissen, was Tod ist und verschiedene Vorstellungen davon haben, wie jemand den Tod einer Person verursachen kann. Eine Menge von Bildern, Geschichten, Erlebtem wird durch so einen Ausdruck hervorgerufen. Ein sprachlicher Ausdruck ist in unserem Weltwissen und unserer Erfahrung verankert. Der Begriff des Mörders hat definierende Eigenschaften. Diese sind durch unser sprachliches Wissen gegeben. So können wir beispielweise nicht sagen:

> Müller ist der Mörder von Schmitt und Schmitt lebt und selbst, wenn er tot wäre, hätte Müller nichts damit zu tun.

Extensional kann der Mörder von Schmitt sich auf Müller beziehen. Die Extension von der Mörder von Schmitt und Müller ist dann gleich. Aber natürlich ist die Intension nicht gleich, denn Müller hat vermutlich noch ganz andere Eigenschaften außer der, ein Mörder zu sein. Jemand, der irrtümlich Schulze für

den Mörder von Schmitt hält, kann mit der Mörder von Schmitt auch Schulze meinen. Für ihn ist die Extension von der Mörder von Schmitt und Schulze gleich, nicht aber die von Schulze und Müller .

Ein anderes Beispiel, das den Unterschied zwischen Extension und Intension zeigt, ist der Ausdruck

der Bundeskanzler der Bundesrepublik Deutschland.

Extensional bezieht er sich auf Helmut Kohl. Intensional bezieht er sich aber auf ein Amt mit bestimmten Rechten und Pflichten, das auf bestimmte Weise vergeben wird. Die extensionale Gleichheit ist zeitlich befristet. Eine intensionale Gleichheit wäre allgemeingültig.

Ein Prinzip, das ebenfalls auf Frege zurückgeführt wird, ist das der Kompositionalität.

**Def.:** Das Prinzip der **Kompositionalität** besagt, daß die Bedeutung eines Ausdrucks sich aus der Bedeutung seiner Teile ergibt.

**Def.:** Das Prinzip der **Substitution** besagt, daß gleichbedeutende Teile eines Audrucks ausgetauscht werden können, ohne daß sich die Bedeutung des Ausdrucks verändert.

Manfred Pinkal (in Görz a.a.O.) gibt überzeugende Beispiele dafür an, daß diese Prinzipien bei natürlichsprachlichen Sätzen nicht angemessen sind. Ich gebe hier lediglich zwei dieser Beispiele wieder. Das erste zeigt, daß die extensionale Gleichheit sich auf einen Zeitraum beschränkt, der verschieden von einem Zeitraum der Intension sein kann.

Helmut Kohl ist der Bundeskanzler.
Der Bundeskanzler hat immer die Richtlinienkompetenz.
*Helmut Kohl hat immer die Richtlinienkompetenz.

Während Helmut Kohl für das Zeitintervall seiner Lebensdauer und das Amt des Bundeskanzlers für die Zeitdauer der Bundesrepublik bestehen, ist die extensionale Gleichheit dieser beiden Intensionen auf ein Zeitintervall irgendwo in dem Schnitt dieser beiden Intervalle beschränkt. Die Richlinienkompetenz rührt von der Definition des Amtes her, nicht notwendigerweise von der Person Helmut Kohl.

Ein anderes Beispiel zielt nicht auf den zeitlichen Verlauf ab, sondern auf die Einstellungen gegenüber Sachverhalten.

Es ist nicht schön, daß Peter ununterbrochen arbeitet.
Peter hat Erfolg genau dann, wenn er ununterbrochen arbeitet.
*Es ist nicht schön, daß Peter Erfolg hat.

Die extensionale Gleichheit der Zustände, in denen Peter Erfolg hat und in denen er ununterbrochen arbeitet, bedeutet keine intensionale Gleichheit. Die Intension des vielen Arbeitens kann negativ, die Intension des Erfolgs positiv bewertet werden.

Da gerade die zeitliche Beschränkung, die Einstellung (z.B.: schön finden) und Modalitäten (z.B. notwendig, möglich) nur intensionsgleiche, nicht aber extensionsgleiche Ausdrücke substituierbar machen, nennt man temporale Ausdrücke, Einstellungsprädikate und Modalausdrücke auch intensionale Ausdrücke. Sie können in der klassischen Prädikatenlogik nicht dargestellt werden, sondern werden in speziellen Logiken behandelt (z.B. mögliche-Welten-Logik, Modallogik).

Das Prinzip der Kompositionalität legt nahe, daß Schritt um Schritt der Aufbau einer semantischen Beschreibung erfolgen könnte. Dies ist aber nicht der Fall, weil die syntaktischen Strukturen nicht direkt semantischen Strukturen entsprechen. Syntaktisch ähnliche Sätze haben semantisch sehr unähnliche Strukturen und umgekehrt.

| Peter arbeitet. | Jemand arbeitet. | Alle arbeiten hier. |
|---|---|---|
| `arbeitet(peter)` | `æX arbeitet(X)` | `ÆX\|an_uni(X)->arbeitet(X).` |

Quantoren sind überhaupt ein Problem: die natürliche Sprache bietet sehr viele Quantoren, während die Prädikatenlogik nur zwei zur Verfügung stellt.

## 4.1 Prädikatenlogik als SRS

Wir brauchen einen Formalismus für die semantische Beschreibung wie wir auch einen Formalismus für die Darstellung (und Erzeugung) einer syntaktischen Strukturbeschreibung brauchten. Ein solcher Formalismus heißt semantische Repräsentationssprache.

**Def.:** Eine **semantische Repräsentationssprache** (SRS) ist ein Formalismus, in dem die Bedeutung natürlichsprachlicher Sätze (oder Texte) dargestellt wird.

Eine Beschreibung in SRS soll

- bei der Analyse aus syntaktischen Beschreibungen und Lexikoneinträgen (sowie möglicherweise anderen Quellen) konstruiert werden können,
- bei der Generierung die Konstruktion einer Strukturbeschreibung erlauben,
- in Hinblick auf konkrete Sachverhalte ausgewertet werden können.

Die SRS muß es erlauben, Inferenzen zu ziehen und allgemeine Aussagen mit konkreten Sachverhalten in Beziehung zu setzen. Der Formalismus muß also eine Ableitungsrelation haben und eine Interpretation von einer Aussage im Formalismus auf den Sachbereich, über den gesprochen wird.

Wir behalten alle guten Gründe bei, die gegen die Prädikatenlogik als SRS sprechen, und verwenden sie dann doch in der Vorlesung als SRS, wenn auch mit der Erweiterung der Lambda-Abstraktion. Die Gründe hierfür sind, daß sich auf diese Weise am leichtesten (d.h. ohne weitere Vorarbeiten) vorführen läßt, wie man von syntaktischen Strukturen zu Ausdrücken in SRS kommt, wie man diese durch Ableitungen ergänzt und auf Erfüllbarkeit hin untersucht, und wie man sie dann auswertet.

### 4.1.1 Lambda-Abstraktion

Lambda ist ein Operator, der Variablen bindet - genauso, wie Quantoren Variablen binden. Er führt eine Variable als Platzhalter ein. Auf diese Weise können wir die Bedeutung von Eigenschaften repräsentieren, bevor wir wissen, wer oder was diese Eigenschaft hat. So können wir

schwarz schreiben als `l X schwarz(X).`

Damit bezeichnet der Lambda-Ausdruck die Eigenschaft schwarz als diejenigen Dinge, für die gilt, daß sie schwarz sind. Dazu mußten wir keinen All-Quantor einführen. Erfahren wir dann, daß das Schwarze eine Katze ist, so ergänzen wir den Ausdruck zu:

`l X schwarz(X) & katze(X).`

Da wir vorher keinen All-Quantor verwendet haben, brauchen wir an der vorherigen Formel nichts zu ändern, sondern mußten sie nur erweitern. Die Lambda-Darstellung erlaubt uns, den Zusammenhang zwischen verschiedenen Beschreibungen herzustellen. Das Schwarze und die Katzeneigenschaft betreffen denselben Referenten. So können wir auch darstellen, welche Aussagen verschiedene Referenten betreffen sollen. Wenn wir an die verschiedenen Verbkategorien denken, so sollen verschiedene Referenten mit den verschiedenen NPs bezeichnet werden:

Er gibt ihr ein Buch.
`l(X,(männlich(X),l(Y,(weiblich(Y),l(Z,(buch(Z),geb(X,Y,Z))))).`

Das ditransitive Verb `geben` hat drei Ergänzungen, die jeweils durch eine Lambda-Variable gekennzeichnet sind. Diese Variable tritt in allen übrigen Beschreibungen des betreffenden Referenten auf. Bei unserem Beispiel Die schwarze Katze fängt die Ratte muß also `schwarz` und `katze` für dasselbe X, `ratte` jedoch für eine andere Variable gelten. Die Behandlung von Kopulaverben (das Verb `sein`), die ja zwei Phrasen gleichsetzen (Helmut Kohl ist der jetzige Bundeskanzler, Der Stuhl ist ein Möbel), wird dadurch auch erleichtert: eventuell auf ein Zeitintervall befristet, werden hier zwei Aussagen an dasselbe Referenzobjekt gebunden.

Während LISP auf dem Lambda-Kalkül aufgebaut ist, muß in Prolog eine Funktion für Lambda eingeführt werden. Sie sollte zweistellig sein, so daß an erster Stelle die Variable, an zweiter die Charakterisierung oder Aussage steht. Die Aussage kann wiederum einen Lambda-Ausdruck enthalten. Für Prolog hat diese Funktion, hier `l` genannt, keine besondere Bedeutung. Insbesondere heißt dies, daß die Unifikation sich wie üblich verhält. Gegeben der Ausdruck

`s(l (X, schwarz(X)), l (Y, frosch(Y)))`, wird die Anfrage `s(l(A,B), l(A,C))` beantwortet:

```
A = _116, B= schwarz(_116), C=frosch(_116).
```

Wir können jetzt in Form von Lambda-Ausdrücken die Bedeutung einiger syntaktischer Kategorien beschreiben. Dabei geben wir für den Sinn meist nichts anderes an als das Wort selbst, ggf. mit einem Zusatzzeichen versehen, das angibt, daß nicht das Wort, sondern sein Sinn gemeint ist. So wird als intensionale Bedeutung von Hans etwa `Hans#` und als Sinn von `geb` etwa `geb#` notiert. Eigennamen können als Konstante oder auch als Lambda-Ausdruck repräsentiert werden. Die Semantik des Verbs ist aber nicht auf den Sinn beschränkt, sondern beinhaltet -- wie wir bei `geb` gesehen haben -- die Ergänzungen. So können wir bei Verbergänzungen angeben, welche Eigenschaften die Referenten der Ergänzungen haben müssen. Beispielsweise muß der Referent für das Subjekt bei `geb` menschlich sein, der Referent für das Dativobjekt belebt, der Referent für das Akkusativbjekt unterliegt dort keinen Einschränkungen. Es können auch mehrere semantische Merkmale für ein Wort angegeben werden. So können wir für `Junggeselle` mehr angeben als nur `junggeselle#`, nämlich

```
l(X,(heiratsfähig(X),unverheiratet(X), männlich(X)).
```

Eine kleine Tabelle zeigt, welche groben Regeln wir für den Zusammenhang von Syntax und Semantik bei verschiedenen Kategorien annehmen.

| **Kategorie** | **Lambda-Ausdruck** | **Prologbeispiel** |
|---|---|---|
| Adjektiv | `l (X,ADJ(X))` | `l (X,schwarz(X))` |
| Nomen | `l (X,N(X))` | `l (X,katze(X))` |
| Eigennamen | `l (X,NAME(X))` | `l (X,müller(X))` |
| Nomen mit Adjektiv | `l (X,(ADJ(X),N(X)))` | `l ((X,schwarz(X),katze(X)))` |
| transitives Verb | `l (X, l(Y,V(X,Y)))` | `l (X,l(Y,jag(X,Y)))` |
| ditransitives Verb | `l(X, l(Y, l(Z,V(X,Y,Z))))` | `l(X, l(Y,l(Z,geb(X,Y,Z))))` |
| Kopulaverb | `l (X, l(Y, ='(X,Y)))` | `l(X, l(Y, X=Y))` |

### 4.1.2 Aufbau der semantischen Beschreibung

Nun haben wir also eine -- bekanntermaßen ungenügende -- semantische Repräsentationssprache (SRS) ausgewählt, nämlich Prolog mit der Funktion Lambda. Wie kann nun für einen Satz der entsprechende Ausdruck in SRS aufgebaut werden? Prinzipiell gibt es zwei Möglichkeiten: Eingabe in die Komponente zum Aufbau eines SRS-Ausdrucks ist die syntaktische Strukturbeschreibung und das Lexikon; oder der SRS-Ausdruck wird während des Parsing aufgebaut, parallel zur syntaktischen Beschreibung. Natürlich gibt es auch Mischformen, in denen zunächst ein Teil der syntaktischen Analyse durchgeführt und semantisch behandelt wird, dann der nächste Teil etc. Beides läßt sich durch den DCG-Formalismus realisieren.

Wir nehmen als Beispiel wieder den Satz Die Katze jagt die Ratte. und modifizieren die bereits bekannte Grammatik für den Aufbau eines SRS-Ausdrucks. Dabei sind die syntaktischen Merkmale nur angedeutet, also keinesfalls vollständig und nicht in Form einer Merkmalsstruktur angegeben. Es soll

lediglich illustriert werden, daß die syntaktische Strukturbeschreibung (mit Merkmalen) und der SRS-Ausdruck von derselben Regelmenge aufgebaut werden können.

```
s( s(NP,VP), sem(l(X,(PN,PV))))-->
      np(NP,kas:nom,num:Num, sem(l(X,PN)) ),
      vp(VP,num:Num,subkat:vt, sem(l(X,PV)) ).

np( np(D,N),kas:K,num:Num, sem(l(X,PN)))-->
      det(D,kas:K,num:Num,sem(_) ),
      n(N,kas:K,num:Num,sem(l(X,PN))).

vp( vp(V,NP),num:Num,subkat:vt, sem(l(X,l(Y,(PO,PVT))))) -->
      v(V, num:Num,subkat:vt, sem(l(X,l(Y,PVT)))),
      np(NP,kas:akk,num:N1, sem(l(Y,PO))).

det(det(die),kas:N, num:sing,sem(_)) --> [die].
n(n(katze),kas:nom, num:sing, sem(l(X,katze(X)))) --> [katze].
n(n(ratte),kas:akk,num:sing, sem(l(X,ratte(X)))) --> [ratte].
v(v(jagt),num:sing,subkat:vt,sem(l(X,l(Y,jagt(X,Y)))) ) --> [jagt].

:-  phrase(s(Syn, Sem), [die, katze, jagt, die, ratte])
```

liefert

```
Syn = s(np(det(die), n(katze)), vp(v(jagt), np(det(die), n(ratte)))),

Sem = sem(l(_1777, (katze(_1777),l(_2721, (ratte(_2721),jagt(_1777, _2721))))))
```

d.h.:

```
Sem = sem(l(X, (katze(X), l(Y, (ratte(Y), jagt(X,Y))))))
```

Wir haben nun in den Lexikoneinträgen Lambda-Ausdrücke, die den Rahmen für ein Wort in der semantischen Repräsentation angeben. Insbesondere wird der Verbrahmen, d.h. die notwendigen Ergänzungen eines Verbs, durch einen Lambda-Ausdruck angegeben. Die Angaben zu den Mitspielern des Verbs werden durch die Lamda-Ausdrücke zusammengesammelt. Zusätzlich können dort semantische Merkmale stehen. So kann z.B. das Verb jagt fordern, daß beide beteiligten Referenten belebt sein müssen. Es wird also zusammengestellt, welche Anforderungen das Verb an seine Ergänzungen stellt, und welche Eigenschaften die jeweiligen NPs tatsächlich haben.

```
s( s(NP,VP), sem(l(X,(PN,PV))))-->
      np(NP,kas:nom,num:Num, sem(l(X,PN)) ),
      vp(VP,num:Num,subkat:vt, sem(l(X,PV)) ).

np( np(D,N),kas:K,num:Num, sem(l(X,PN)))-->
      det(D,kas:K,num:Num,sem(_) ),
      n(N,kas:K,num:Num,sem(l(X,PN))).

vp( vp(V,NP),num:Num,subkat:vt, sem(l(X,l(Y,(PO,PVT))))) -->
      v(V, num:Num,subkat:vt, sem(l(X,l(Y,(PO,PVT))))),
      np(NP,kas:akk,num:N1, sem(l(Y,PO))).

det(det(die),kas:N, num:sing,sem(_)) --> [die].

n(n(katze),kas:nom, num:sing, sem(l(X,belebt(X)))) --> [katze].

n(n(ratte),kas:akk,num:sing, sem(l(X,belebt(X)))) --> [ratte].

n(n(lampe),kas:akk,num:sing, sem(l(X, not belebt(X)))) --> [lampe].

v(v(jagt),num:sing,subkat:vt,sem(l(X,l(Y,(belebt(Y),(belebt(X),jagt(X,Y)))))))
                                                              -->[jagt].
```

Hier ist nun das semantische Merkmal `belebt` eingeführt, das bei der Lampe negiert ist. Damit der Zusammenhang zwischen der Charakterisierung des Objekts in dem Lexikoneintrag für das Verb und in der

Beschreibung der von der VP abhängigen NP hergestellt wird, wurde jetzt die Variable `PO` bei beiden Literalen der VP-Klausel angegeben, so daß die Unifikation erzwungen wird.

```
:-  phrase(s(Syn, Sem), [die, katze, jagt, die, ratte]).
```

liefert

```
Syn = s(np(det(die), n(katze)), vp(v(jagt), np(det(die), n(ratte)))),

Sem=sem(l(_1593,(belebt(_1593),l(_2088,(belebt(_2088),belebt(_1593),
                                                    jagt(_1593,_2088))))))
```

und

```
:-  phrase(s(Syn, Sem), [die, katze, jagt, die, lampe]).
```
liefert

```
no
```

### 4.1.3 Begriffliches Wissen

Wir haben eben einen semantisch nicht wohlgeformten Satz durch die Unifikation von einem semantischen Merkmal in der Anforderung des Verbs und in der betreffenden NP ausgeschlossen. Die Gleichheit der Eigenschaften ist aber nicht ganz das, was wir wollen. Der Nachteil dieses Vorgehens ist zum einen, daß nur jeweils ein semantisches Merkmal behandelt werden kann. Der noch wichtigere Nachteil ist aber, daß die Beziehung zwischen `belebt` und `menschlich` nicht berücksichtigt wird! Wir wissen aber:

```
belebt(X) :- menschlich(X).
```

Obwohl also `menschlich(X)` und `belebt(X)` nicht dieselbe Aussage darstellen, sind sie doch -- im Gegensatz zu `belebt(X), not belebt(X)` -- zusammen erfüllbar. Wir müssen also Wissen über semantische Beziehungen -- z.B. in Form von Regeln bzw. Prolog-Klauseln -- repräsentieren und die offenen Formeln innerhalb der Lambda-Ausdrücke auf Erfüllbarkeit testen. Wir brauchen dabei noch keinen Rückgriff auf das Weltwissen, sondern betrachten nur, was von der Wortbedeutung impliziert wird. Wir bauen uns also eine Wissensbasis auf, die Zusammenhänge zwischen semantischen Merkmalen enthält. Diese Wissensbasis wird ergänzt durch die semantischen Merkmale in Lexikoneinträgen. Da diese während der Verarbeitung eines Satzes aufgebaut werden, müssen sie für Prolog als dynamische Datenstrukturen deklariert werden.

```
:- dynamic belebt/1.
:- dynamic mensch/1.
:- dynamic katze/1.
:- dynamic ratte/1.
:- dynamic lampe/1.
:- dynamic mann/1.
```

Als vorgegebenes begriffliches Wissen nehmen wir einmal nur die oben schon genannte Regel an. Natürlich könnte auch im Lexikon nur `katze(X)` stehen und dann im begrifflichen Wissen angegeben werden:

```
belebt(X) :- katze(X).
belebt(X):- mensch(X).
```

Jetzt müssen wir überprüfen, ob die semantischen Merkmale zueinander passen. Im Gegensatz zu den syntaktischen Merkmalen handelt es sich hierbei um Aussagen über eine Variable. Um die Erfüllbarkeit dieser Ausdrücke zu überprüfen, gehen wir so vor:

1. Die Lambda-Variablen werden skolemisiert, d.h. verschiedene Variablen werden durch verschiedene Konstanten ersetzt. Jede Konstante beginnt mit `refo_` für "Referenzobkjekt".

2. Alle positiven semantischen Merkmale, die bei der Verarbeitung eines Satzes aufgesammelt wurden, werden in die Wissensbasis für begriffliches Wissen dynamisch eingetragen.

3. Alle negativen semantischen Merkmale werden in positiver Form als Anfrage an die Wissensbasis gestellt - schlägt die Anfrage nicht fehl, so besteht ein Widerspruch zwischen zwei Merkmalen desselben Referenten. Schlägt die Anfrage fehl, so ist der Ausdruck erfüllbar.

```
%konsistent(Lambda-Ausdruck)
%liefert Erfolg oder schlägt fehl und gibt eine Meldung aus.
konsistent(OL):-
      copy_term(OL,L), % Den Orginal-Ausdruck nicht verändern,
                       %damit auch die Formel mit Variablen noch zur Verfügung steht!
      init_gensym(refo_),
      parse_lambda(L,Pos,Neg),%da Lambda-Ausdrücke komplex sein können, müssen sie
                              %analysiert werden.
      assert_all(Pos),        %Alle positiven semantischen Merkmale
                              %werden eingetragen in die Wissensbasis --
      !,                      %dies braucht nur einmal zu geschehen.
      call_all(Neg).          %Alle negativen Merkmale werden in positiver Form
                              %als Anfrage an die Wissensbasis gestellt.

%parse_lambda zerlegt einen komplexen Lambda-Ausdruck in positive und negative
%semantische Merkmale und skolemisiert die Variablen.
parse_lambda(l(V,A),P,N):-
      gensym(refo_,V),        %Skolemisierung
      parse_lambda(A,P,N).

parse_lambda((A,B),P,N):-
      parse_lambda(A,P1,N1),
      parse_lambda(B,P2,N2),
      append(P1,P2,P),        %Erzeugen einer Liste von positiven sem. Merkmalen
      append(N1,N2,N).        %Erzeugen einer Liste von negativen sem. Merkmalen

parse_lambda(not(P),[],[P]). %Die negativen Merkmale kommen in die zweite Liste.

parse_lambda(P,[P],[]).%Die positiven Merkmale kommen in die erste Liste.

%Eintragen einer ganzen Liste von semantischen Merkmalen -- hier:positive!
assert_all([]).
assert_all([FirstP|RestP]):-
      assert(FirstP),
      assert_all(RestP).

%Anfragen aller in einer Liste gesammelten Ausdrücke -- hier: negative sem. Merkmale!
call_all([]).
call_all([FirstN|RestN]):-
      call(\+FirstN),         %Schlägt der skolemisierte Ausdruck wirklich fehl?
      !,                      %Falls nein, soll kein backtracking versucht werden,
      call_all(RestN).
call_all([FirstN|_]):-        %sondern diese Klausel:
      write('Wiederspruch: '),%Meldung, dass sem. Merkmale einander ausschließen
      write(not(FirstN)),nl,
      !, fail.                %Fehlschlag und keine weiteren Versuche!
```

Nun müssen wir diesen Erfüllbarkeitstest in die Grammatik einfügen. Wir wollen prüfen, ob alle Mitspieler des Verbs, wie sie von der syntaktischen Analyse ermittelt werden, auch den semantischen Anforderungen entsprechen, die das Verb an sie stellt. Die Objekte bei ditransitiven Verben bzw. das Objekt bei transitiven Verben werden an letzter Stelle der VP-Klausel geprüft. Das Subjekt wird an letzter Stelle der S-Klausel geprüft. Da es sich bei `konsistent` um ein Prolog-Programm handelt, das nicht dem DCG-Formalismus entspricht, muß das Literal in geschweifte Klammern gesetzt werden.

```
s( s(NP,VP),      sem(l(X,(PN,PV))))-->
      np(NP,kas:nom,num:Num, sem(l(X,PN)) ),
      vp(VP,num:Num,subkat:vt, sem(l(X,PV)) ),
      {konsistent(l(X,(PN,PV)))}.
np( np(D,N),kas:K,num:Num, sem(l(X,PN)))-->
      det(D,kas:K,num:Num,sem(_) ),
      n(N,kas:K,num:Num,sem(l(X,PN))).

vp( vp(V,NP),num:Num,subkat:vt, sem(l(X,l(Y,(PO,PVT))))) -->
```

```
      v(V, num:Num,subkat:vt, sem(l(X,l(Y,PVT)))),
      np(NP,kas:akk,num:N1, sem(l(Y,PO))),
      {konsistent(l(X,l(Y,(PO,PVT))))}.

det(det(die),kas:N, num:sing,sem(_)) --> [die].
det(det(der),kas:nom,num:sing,sem(_)) --> [der].
n(n(katze),kas:nom, num:sing, sem(l(X,(katze(X),belebt(X)))))--> [katze].
n(n(ratte),kas:akk,num:sing, sem(l(X,(ratte(X),belebt(X)))))--> [ratte].
n(n(mann),kas:nom, num:sing, sem(l(X,(maennl(X),erwachsen(X),mensch(X)))))--> [mann].
n(n(lampe),kas:akk,num:sing, sem(l(X,not(belebt(X)))))--> [lampe].

v(v(jagt),num:sing,subkat:vt,sem(l(X,l(Y,(belebt(Y),(belebt(X),jagt(X,Y)))))))-->
                                                                        [jagt].
```

Wenn der Eingabesatz semantisch wohlgeformt ist, erhalten wir eine semantische Beschreibung wie bereits gehabt. Die Skolemterme sind nicht sichtbar, weil `copy_term` das Original mit Variablen aufbewahrt hat.

```
%:-  phrase(s(Syn, Sem), [die, katze, jagt, die, ratte])
%N°1  Syn = s(np(det(die), n(katze)), vp(v(jagt), np(det(die), n(ratte)))),
%Sem = sem(l(_1299, ((katze(_1299),belebt(_1299)),
%           l(_1419, ((ratte(_1419),belebt(_1419)),
%           belebt(_1419),belebt(_1299),jagt(_1299, _1419))))))
```

Wenn der Eingabesatz nicht wohlgeformt ist, weil die semantische Beschränkung, die das Verb in seinem Verbrahmen einem Mitspieler auferlegt, nicht eingehalten wurde, gelingt die Verarbeitung des Eingabesatzes nicht und es wird eine Meldung ausgegeben. In dieser Meldung tritt der Skolemterm auf, mit dem die Anfrage scheiterte.

```
:-  phrase(s(Syn, Sem), [der, mann, jagt, die, lampe])
Wiederspruch: not belebt(refo_1)
no
Dabei ist refo_0 der Skolemterm fuer mann  und refo_1 der fuer die Lampe.
```

Die Vereinbarkeit von `mensch(X)` und `belebt(X)` ist notwendig, damit der zweite Erfüllbarkeitstest, der in der S-Klausel NP und VP untersucht, gelingt.

```
:-  phrase(s(Syn, Sem), [der, mann, jagt, die, ratte])
N°1   Syn = s(np(det(der), n(mann)), vp(v(jagt), np(det(die), n(ratte)))),
Sem = sem(l(_1297, ((maennl(_1297),erwachsen(_1297),mensch(_1297)),
           l(_1424, ((ratte(_1424),belebt(_1424)),
           belebt(_1424),belebt(_1297),jagt(_1297, _1424)))))
```

Hier ist also aus `mensch(refo_0)` auch `belebt(refo_0)` abgeleitet worden!

Die interne Darstellung sieht man, wenn man `copy-term` auskommentiert:

```
:-  phrase(s(Syn, Sem), [die, katze, jagt, die, ratte])
N°1   Syn = s(np(det(die), n(katze)), vp(v(jagt), np(det(die), n(ratte)))),
Sem = sem(l(refo_0, ((katze(refo_0),belebt(refo_0)),
           l(refo_1, ((ratte(refo_1),belebt(refo_1)),
       belebt(refo_1),belebt(refo_0),jagt(refo_0, refo_1))))))
```

Wir haben hier eine kleine Wissensbasis für das begriffliche Wissen aufgebaut. In realen NLS wird für diese Aufgabe oft ein Termsubsumtionssystem eingesetzt. Damit kann der Begriff katze zum Beispiel mit all seine notwendigen semantische Merkmalen definiert werden. Zwei Begriffe sind miteinander verträglich, wenn ihre Schnittmenge nicht notwendigerweise leer ist. Im Lexikon verbleibt dann nur noch der Begriff -- seine notwendigen Eigenschaften sind in der TBox abgelegt. So würde im Lexikon für `mann` nur noch `mann(X)` stehen. In der TBox sind die definierenden Eigenschaften (`erwachsen`, `maennl`, `mensch`) repräsentiert. Die Inferenzen der TBox werden zur Erfüllbarkeitsprüfung und zur Ergänzung der Semantik genutzt.

Der Grund, warum wir die semantische Wohlgeformtheit vor der Auswertung prüfen, ist, daß wir nicht unnötig eine womöglich aufwendige Datenbankanfrage stellen wollen. Außerdem wollen wir mit unserer Grammatik keine semantisch nicht wohlgeformten Sätze beschreiben (und damit auch generieren!)

### 4.1.3 Behandlung von Quantoren

In den Beispielen zum Aufbau einer semantischen Beschreibung haben wir den bestimmten Artikel nicht behandelt, sondern seine Semantik durch eine anonyme Variable ausgedrückt. Gerade die Quantoren sind aber in der natürlichen Sprache von großer Bedeutung. Die einfachen Quantoren der Prädikatenlogik lassen sich leicht übersetzen:

| | |
|---|---|
| alle Menschen sind sterblich | Æ X `mensch (X) --> sterblich (X)` |
| eine Katze jagte eine Ratte | æ X `katze(X)` & (æ Y `ratte(Y) & jagt(X,Y))` |

Statt `alle` kann auch `jeder` mit dem Allquantor übersetzt werden. Die Aussage, die die quantifizierte Menge einschränkt, wird die **Restriktion** genannt. Dies ist hier: `mensch(X), katze(X), ratte(X)`. Die Aussage über diese quantifizierte Menge wird **Skopus** genannt. Dies ist hier: `sterblich (X), jagt(X,Y)` und (æ `Y ratte(Y) & jagt(X,Y))`.

Im allgemeinen ist ein Quantor etwas, das eine Restriktion und einen Skopus hat. Wir können dies mit einem Operator `q` ausdrücken, der aus einem Lambda-Ausdruck einen Lambda-Ausdruck oder eine Formel aufbaut:

```
q( q( (l(X,Restriktion), l(X,Skopus)), Formel).
```

Dies soll heißen: aus Restriktion und Skopus wird eine Formel aufgebaut, die ihrerseits ein Quantor-Ausdruck ist. Wir können jetzt folgende Lexikoneinträge schreiben:

```
det(det(eine),kas:N,num:sing,sem(q(q(l(X,R),l(X,Sk)), some(X,R,Sk))) --> [eine].

det(det(jede),kas:N,num:sing,sem(q(q(l(X,R),l(X,Sk)), all(X,R,Sk))) --> [jede].
```

Um die Lambda-Ausdrücke herum konstruieren wir eine quantifizierte Formel. Obwohl `q` eine Funktion ist, hat es hier die Rolle eines formelbildenden Operators. Was nun in dem Prolog-Programm ein Term darstellt, ist eigentlich die Realisierung eines Ausdrucks einer Logik höherer Stufe.

```
s( s(NP,VP),sem(Sem)) -->
      np(NP,kas:nom,num:Num, sem(q(l(X,Sk),Sem) ) ),
      vp(VP,num:Num,subkat:vt, sem(q(X,Sk) )).

np( np(D,N),kas:K,num:Num, sem(q(l(X,Sk),Quant)))-->
      det(D,kas:K,num:Num,sem( q(q(l(X,R),l(X,Sk)),Quant) ) ),
      n(N,kas:K,num:Num,sem(l(X,R))).

vp( vp(V,NP),num:Num,subkat:vt, sem(l(X,Quant))) -->
      v(V, num:Num,subkat:vt, sem( l(X,l(Y,Sk)) ) ),
      np(NP,kas:akk,num:N1, sem( q( l(Y, Sk),Quant) ) ).

v(v(jagt),num:sing,subkat:vt,sem(l(X,l(Y,jagt(X,Y)))) ) --> [jagt].
v(v(jagt),num:plur,subkat:vt,sem(l(X,l(Y,jagt(X,Y)))) ) --> [jagen].

det(det(eine),kas:N, num:sing,sem( q(q(l(X,R),l(X,Sk)),some(X,R,Sk)) ) ) --> [eine].
det(det(alle),kas:N, num:plur,sem( q(q(l(X,R),l(X,Sk)),all(X,R,Sk)) ) ) --> [alle].
n(n(katze),kas:nom, num:sing, sem(l(X,katze(X)))) --> [katze].
n(n(katze),kas:nom, num:plur, sem(l(X,katze(X)))) --> [katzen].
n(n(ratte),kas:akk,num:sing, sem(l(X,ratte(X))))--> [ratte].
n(n(ratten),kas:akk,num:plur, sem(l(X,ratte(X))))--> [ratten].

:-  phrase(s(Syn, Sem), [eine, katze, jagt, alle, ratten])
```

liefert

```
N°1   Syn = s(np(det(eine), n(katze)), vp(v(jagt), np(det(alle), n(ratten)))),
      Sem = sem(some(_1608,katze(_1608), all(_2135,ratte(_2135), jagt(_1608,_2135))))

:-  phrase(s(Syn, Sem), [alle, katzen, jagen, eine, ratte])
N°1   Syn = s(np(det(alle), n(katze)), vp(v(jagen), np(det(eine), n(ratte)))),
      Sem = sem(  all(_1317, katze(_1317),
                  some(_1417, ratte(_1417), jagt(_1317, _1417))))
```

Eine der Schwierigkeiten bei der Behandlung der Quantoren ist, daß sie zwar lokal bei einer NP auftreten, jedoch Geltung für den gesamten Satz haben. Wir haben hier als semantische Beschreibung des gesamten Satzes eine quantifizierte Formel, deren Quantor aus der direkt von S dominierten NP herrührt. Auch in der VP wird der Quantor der von VP dominierten NP nach oben gezogen und enthält nun die Aussage des Verbs als Skopus.

Wir haben nun die Behandlung des unbestimmten Artikels und der Quantoren `all` und `some` gesehen. Der bestimmte Artikel hat die Bedeutung, daß ein bereits eingeführter Referent und nicht etwa ein neues Objekt gemeint ist, und daß der Referent eindeutig zu bestimmen ist. Er weist somit über die Satzgrenze hinaus.

In meiner Straße steht ein kleines Haus, gleich neben dem Park.
Das Haus ist rosa gestrichen.

Hier ist zu bemerken, daß sich in einer Straße mehrere Häuser befinden, aber nur ein Park. Deshalb wird das Haus mit dem unbestimmten Artikel eingeführt, der Park aber gleich mit dem bestimmten. Da im nachfolgenden Satz dasselbe Haus gemeint ist, kann hier der bestimmte Artikel verwendet werden. Auf solche satzübergreifenden Phänomene wird später eingegangen.

Die Bedeutung der bestimmten und unbestimmten Artikel ist mehrdeutig:

Der Löwe frißt Fleisch. Ein Löwe ist ein Fleischfresser.

Hier wird sowohl der bestimmte wie auch der unbestimmte Artikel generisch gebraucht, d.h. er bestimmt eine Klasse und kann daher mit dem All-Quantor übersetzt werden. Fast derselbe Satz erhält aber eine andere Bedeutung, wenn Zeit- oder Ortsangabe die Aussage auf einen speziellen Zeitpunkt oder Ort einschränken:

Der Löwe frißt gerade Fleisch.

Quantoren wie relativ viele, fast alle, ziemlich wenige... sind deshalb so kompliziert, weil sie eine Annahme über die übliche Quantität einer Bezugsmenge beinhalten. Ziemlich wenige Blätter am Baum können immer noch viel mehr sein als relativ viele Kinder in der Familie. Programme für solche Quantoren müssen also zunächst die mögliche Kardinalität der Menge, die durch die Restriktion gegeben ist (Anzahl der Blätter bei Bäumen, Anzahl der Kinder bei Familien), ermitteln und dann einen Prozentsatz davon bestimmen, für den auch die Aussage im Skopus gelten muß.

Während die semantische Beschreibung in Bezug auf das begriffliche Wissen verborientiert war, ist die semantische Beschreibung in Hinblick auf die Quantoren NP-orientiert. Wir können beide Beschreibungen nebeneinander aufbauen, indem wir für einen Satz die syntaktische Strukturbeschreibung, die offene Formel für den begrifflichen Gehalt und die geschlossene Formel für die Quantifizierung ausgeben. Der Klauselkopf der S-Klausel lautet dann:

```
s( s(NP,VP),sem1(l(X,(PN,PV))),sem2(Sem)) --> ...
```

### 4.1.4 Auswertung der semantischen Beschreibung

Wir haben nun zwei semantische Beschreibungen erstellt und mit der einen die Erfüllbarkeit festgestellt. Mit der anderen wollen wir nun die Satzsemantik über dem Weltwissen auswerten. Die Formel soll also nun nicht nur erfüllbar sein, sondern auch tatsächlich gelten. Wir nehmen als gegeben an eine Wissensbasis, die das Weltwissen der Hörers darstellt. In dieser Wissenbasis stehen Grundfakten und Regeln. Der Einfachheit halber nehmen wir an, daß dieselben Prädikate verwendet werden, wie im Lexikon für den Sinn angegeben. Ansonsten brauchen wir noch eine Übersetzungstabelle zwischen Lexikonangaben und Wissensbasisangaben.

Eine kleine Beispielwissensbasis ist:

```
katze(felix).     katze(fanni).     mensch(udo).
ratte(ulf). ratte(ina). maennl(udo).
jagt(felix,ulf).                    erwachsen(udo).
jagt(felix,ina).  jagt(fanni,ina).  jagt(udo,ulf).
```

Wir sehen, daß eine Katze, nämlich Felix, alle Ratten jagt, daß eine Ratte, nämlich Ina, von allen Katzen gejagt wird, und daß ein Mann eine Ratte, nämlich Ulf, jagt. Für die Auswertung nehmen wir die aufgebaute quantifizierte Formel.

```
all(_1317, katze(_1317), some(_1417, ratte(_1417), jagt(_1317, _1417)))
```

Wir wollen nun erst alle Instanzen in der Wissensbasis suchen, für die die Restriktion gilt. Dann wollen wir nachsehen, ob für alle oder einige von diesen auch die Aussage im Skopus gilt. Wenn diese wiederum quantifiziert ist, müssen wir auch dort die Instanzen für die Restriktion suchen.

Wir können in Prolog für die Quantoren kleine Programme schreiben:

```
%all (-X, +Ziel1, +Ziel2)  z.B.all (X, mensch(X), sterblich(X))
all(X, Ziel1, Ziel2) :-
not (Ziel1, not (Ziel2, write(Ziel1),nl)),     %die Negation wirft
                                               %Variablenbindungen weg!
not(not(Ziel1)), !.                            %Präsuppositionsüberprüfung

%some(-X,+Ziel1, +Ziel2)
some(_, Ziel1, Ziel2):-
      not( not( (Ziel1,
                 Ziel2,
                 write(Ziel1),nl))),!.
```

Diese Programme werden über dem Weltwissen ausgewertet und liefern wahr, wenn alle bzw. eine Lösung für die Ziele dort gefunden werden. Die Präsuppositionsüberprüfung realisiert die unterschiedliche Auffassung in der formalen Logik und in natürlicher Sprache von der Wahrheit einer Implikation, bei der der Antezedens falsch ist. Während in der Logik alle Einhörner sind grün als wahre Aussage betrachtet wird, ist in der natürlichen Sprache eine Voraussetzung für die Wahrheit, daß es ein Einhorn gibt, der Antezendens (bzw. die Restriktion) also wahr ist.

Da Prolog die Variablenbindungen aus einem negierten Literal wegwirft, wird sie geschrieben. Das Ergebnis soll aber gar nicht alle Bindungen enthalten, sondern nur den Wahrheitswert. Deshalb wird der positive Ausdruck als doppelte Negation formuliert.

Wir schreiben nun ein kleines Programm zur Auswertung. Es ruft zuerst die Satzanalyse auf, nimmt dann den quantifizierten semantischen Ausdruck und wertet ihn über der Wissensbasis mithilfe der Programme `all` und `some` aus.

```
%satz(+Eingabe,_,-Quant)
satz(Eingabe,_,Quant):-
phrase(s(Syn,_,sem(Quant)),Eingabe),
call(Quant),!.
```

Das Verhalten dieses Programms ist in etwa das, was wir erwarten. Es schreibt die Restriktionen aus und gibt als Ergebnis den gelungenen quantifizierten Ausdruck aus.

```
:-  satz([eine, katze, jagt, alle, ratten], Q)
```
liefert:
```
ratte(ulf)
ratte(ina)
katze(felix)
N°1 Q = some(_1025, katze(_1025), all(_1125, ratte(_1125), jagt(_1025, _1125)))
```

Hier wird die Katze Felix gefunden, die alle Ratten (nämlich Ulf und Ina) jagt.

```
:-  satz([eine, katze, jagt, eine, ratte], Q)
```
liefert:
```
ratte(ulf)
katze(felix)
N°1 Q = some(_1327, katze(_1327), some(_1427, ratte(_1427), jagt(_1327, _1427)))
```

Hier wird nur eine Katze und eine Ratte gefunden - mehr war nicht gefragt!

```
:-  satz([alle, katzen, jagen, eine, ratte], Q)
```
liefert:

```
ratte(ulf)
katze(felix)
ratte(ina)
katze(fanni)
N°1 Q = all(_1286, katze(_1286), some(_1386, ratte(_1386), jagt(_1286, _1386)))
```

Es war nicht gefordert, daß es dieselbe Ratte sein muß, die von allen Katzen gejagt wird. Daher wird für jede Katze nur eine Ratte gefunden, die von ihr gejagt wird.

```
:-  satz([ein, mann, jagt, eine, ratte], Q)
```
liefert
```
ratte(ulf)
mann(udo)
N°1 Q = some(_1668, mann(_1668), some(_1768, ratte(_1768), jagt(_1668, _1768)))
```

Hier wurde auch das begriffliche Wissen benötigt, um Udo zu finden.

Ein vollständiges NLS nimmt das Ergebnis der Auswertung (die Antwortbasis) als Ausgangspunkt für die Antwortgenerierung. Aus der Antwortbasis wird eine semantische Beschreibung geformt. Dies ist hier die Beschreibung, die an Q gebunden wurde mit der Bindung der betreffenden Variablen an `udo` und `ulf`. Für diese semantische Beschrebung muß dann eine passende syntaktische Struktur gefunden und eine natürlichsprachliche Ausgabe erzeugt werden. Bei der Antwortgenerierung wird berücksichtigt, was nicht verbalisiert werden muß, da der Hörer es schon weiß. Wie auf das Wissen des Hörers eingegangen wird, ist ein Phänomen, das der Pragmatik zugerechnet wird.

## 5. Pragmatik

Levinson (a.a.O.) beschreibt ausführlich die Probleme, den Begriff "Pragmatik" zu definieren.

- Beschäftigt sich die Pragmatik nur mit Performanz-Phänomenen oder gibt es auch auf der Pragmatik-Ebene eine Kompetenz?

Wenn wir pragmatische Prinzipien aufdecken wollen, so sollen dies Prinzipien sein, die von Hörern und Sprechern einer Sprache auch verwendet werden bzw. adäquat beschreiben, welche Äußerungen in welchen Kontexten für angemessen gehalten werden. Ein typisches Performanzproblem wie etwa ein Versprecher oder ein abgebrochener oder falsch zuende gebrachter Satz sind aber nicht kontext-spezifisch.

- Beschäftigt sich Pragmatik mit der Äußerungsbedeutung, während Semantik sich mit der Satzbedeutung beschäftigt?

Dies ist eine Abschwächung der Performanzbindung der Pragmatik, insofern als die Äußerungsbedeutung, nicht jedoch die Äußerungsproduktion (abgebrochen, versprochen,...) beschrieben werden soll.

- Beschäftigt sich die Pragmatik mit allen Kontexten, die zur Desambiguierung von Sätzen notwendig sind, mit solchen, die in der Struktur der Sprache kodiert sind, oder mit allen, die zum Verstehen von Äußerungen notwendig sind?

Gerade hier kommt die KI in linguistischen Diskursen vor. Während die Chomsky-Schule so wenig als möglich von der Welt oder dem Wissen über die Welt in eine linguistische Theorie einbeziehen will, wird in der KI alles, was zum Sprachverstehen an Weltwissen notwendig ist, in die Betrachtung einbezogen. Das Problem ist dabei die ungeheure Fülle an Weltwissen (Enzyklopädie-Problem). Ein Ausweg ist dabei die mittlere Position. Ein anderer besteht darin, die Prinzipien des Zugriffs auf das Weltwissen zu untersuchen, nicht jedoch das Weltwissen selbst zu repräsentieren. Ich schlage vor, die Prinzipien des Zugriffs auf die Welt selbst und die Prinzipien der Übermittlung von Weltwissen durch Sprache einzubeziehen -- gerade dies ist bisher noch nicht geschehen.

- Was ist ein Kontext und was ist irrelevantes Beiwerk einer Situation?

- Beschäftigt sich Pragmatik mit der Angemessenheit von Sätzen, wo die Semantik sich auf die Wahrheitsbedingungen von Sätzen konzentriert?

Falls man diese Definition befolgt, nimmt man eine pragmatische Kompetenz an. Das Problem ist, wie man aus den beobachtbaren Äußerungen auf die Richtlinien der Angemessenheit kommt. Schließlich verhalten sich Menschn nicht immer angemessen!

- Ist Pragmatik eine Abbildung von Kontext und Satz auf eine Aussage (Proposition)? Oder ist Pragmatik eine Abbildung von Kontext und Satz auf einen neuen Kontext, der nämlich durch die Äußerung erzeugt wird?

Diese letztere Idee von Gazdar (zitiert in Levinson a.a.O., S.31) kommt meiner Beantwortung auf die Frage nahe, wie Welt und Weltwissen einzubeziehen sei.

Während also eine klare Definition nicht gelingt, bleibt doch eine Reihe von Themen, bei denen sich die meisten sicher sind, daß sie zum Gegenstandsbereich der Pragmatik gehören: Deixis, Präsuppositionen, Konversationsimplikatur, Sprechhandlungen und Dialog- oder Textstruktur. Genau diese Bereich werden hier kurz vorgestellt.

Die besondere Bedeutung der Pragmatik für NLS und überhaupt Mensch-Maschine-Kommunikation ergibt sich daraus, daß ein System, das mit Menschen kommuniziert und nicht nur eine Kommunikation von Mensch zu Mensch vermittelt oder die Wohlgeformtheit bzw. Wahrheit eines Satzes feststellt, eine Rolle (Sprecher/Hörer) in der Kommunikationssituation einnimmt. Menschen sind pragmatische Prinzipien unbewußt. Daher können sie nicht anders, als sie anzuwenden. Ein System muß so konstruiert sein, daß es den BenutzerInnen erlaubt, ihr eigenes, unbewußtes Verhalten beizubehalten.

## 5.1 Deixis

*Ein Lehrer schickt seinen Schüler mit einem Zettel zur Lehrersgattin: "Sende dir durch diesen Jungen deine Pantoffeln!" Als der Schüler erstaunt fragt, wieso der Lehrer "deine Pantoffeln" geschrieben habe, antwortet der: " Nu, wenn ich `meine Pantoffeln´ geschrieben hätte, so hätte sie gelesen `meine Pantoffeln´und hätte mir ihre geschickt. So liest sie `deine Pantoffeln´und schickt mir also meine."*
So ähnlich auch in Levinson (a.a.O., S.70).

**Deixis** bezieht sich auf sprachliche Ausdrücke, die auf den Kontext Bezug nehmen. Es geht also um sprachliche Ausdrücke für Person, Ort und Zeit einer Äußerung. Dabei gibt es solche Ausdrücke, die nur im Zusammenhang mit Gesten bei zeitlicher Gleichheit und Sichtbarkeit (evtl.über ein Medium) verwendet werden:hier, da drüben, in einer Stunde, der da,... Andere Ausdrücke können auch in einem Buch, also bei sprachlich präsentiertem Kontext verwendet werden: diese Stadt, ihr, du, vor 10 Jahren, ... Hier reicht es, zu wissen, wer mit wem spricht, wann und wo. Wie können die Konventionen, die die natürliche Sprache festlegt, in multi-medialer Kommunikation am besten angenähert werden? Wenn die Maus eine Geste darstellt, wie wird dann ein begleitender Text geschrieben? Wie muß der sprachliche Ausdruck sein, wenn eine WWW-Seite aus drei Feldern bestehen würde: ein Feld zeigt, worüber gesprochen wird (z.B. einen neuen Roboter), ein Feld zeigt jemanden, der den Roboter vorstellt, das dritte Feld zeigt einen Text über den Roboter, den man ausschneiden und selbst weiterverwenden kann.

Die sprachlichen Mittel, Personen zu bezeichnen, die an einer Kommunikation beteiligt sind, sind in verschiedenen Sprachen sehr unterschiedlich. Es kann ein den Sprecher einschließendes und ein ausschließendes wir geben. Es können Rangunterschiede zwischen Sprecher und Hörer ausgedrückt werden. Mindestens Sprechereinschluß -- bei erster Person (+) und dritter Person (-) --, Adressanteneinschluß -- bei zweiter Person (+) und dritter Person (-) -- sind wichtige Merkmale zur Beschreibung von Personalpronomina.

Bezeichnungen einer Person können unterschiedlich sein, je nachdem, ob die Person direkt angesprochen oder über sie gesprochen wird. So ist im Griechischen z.B. die Anredeform von Jannis kurz Janni. Vokative sind NPs, die auf den Adressaten der Äußerung referieren, aber nicht syntaktisch in den Satz eingebunden sind. *He Sie,* Sie haben gerade mein Auto zerkratzt!

Die unterschiedlichen Medien führen zu unterschiedlicher Selbstreferenz. In direktem Gespräch stelle ich mich vor: " Ich bin Katharina." Am Telefon hingegen:"Hier ist Katharina Morik". Welche Konventionen gelten bei e-mail, Bildtelefon, WWW? Welche Anredeformen gelten bei Mensch-Maschine-Kommunikation?

Außer den AdressatInnen kann es die MithörerInnen und Äußerungsziele geben, außer den SprecherInnen von ihnen verschiedene Äußerungsquellen. Wer oder was ist die Äußerungsquelle bei einem NLS? Wer ist das Äußerungsziel bei einer mailbox-Diskussion?

Neben der Personeneixis gibt es die Zeitdeixis und die Raumdeixis. Während Ortsangaben wie Das Museum liegt direkt neben dem Dom am Hauptbahnhof unabhängig vom Kontext der Äußerung sind, ist Das Museum ist nur 200m von hier entfernt wahr oder falsch je nachdem, wo sich der Sprecher während der Äußerung befindet. Auch die Verben kommen und gehen könnne nur in Bezug auf den Kontext ausgewählt werden. Sie nehmen auf den Sprecherstandort Bezug.

## 5.2 Präsuppositionen

*"Moische, was soll eigentlich das `p´ im Namen `Hannah´?" - "Im Namen `Hannah´ist doch gar kein `p´drin!" - "Wieso ist keines drin?" - "Was soll denn ein`p´ im Namen `Hannah´?" - "Das frage ich ja eben!"*

Es gibt verschiedene Definitionen, was Präsuppositionen eigentlich sind. Hier will ich die folgende Definition verwenden:

**Def.: Präsuppositionen** sind diejenigen Annahmen P, die ohne Wissen über die Welt und ohne kommunikative Konventionen durch einen Satz S dergestalt impliziert werden, daß gilt:

S --> P & ¬ S --> P

So gilt z.B. bei dem Satz

Herr Müller hat aufgehört, die CDU zu wählen.

ebenso wie bei dessen Negation

Herr Müller hat nicht aufgehört, die CDU zu wählen.

daß Herr Müller (früher) die CDU gewählt hat.

Auch Befehle haben Präsuppositionen. So präsupponiert der Befehl

Öffne bitte die Tür!

daß die Tür geschlossen ist. Wenn der Adressat feststellt, daß sie bereits geöffnet ist, so ist nein! vielleicht nicht die richtige Antwort, sondern eher

Aber sie ist bereits geöffnet!

Auch Fragen präsupponieren etwas, nach dem nicht gefragt wird. Bei Frage und Antwort ergibt sich ein besonderes Problem, falls die Präsupposition verletzt ist: weder die Antwort nein noch die Antwort ja sind wahr.

Ist der König von Hamburg kahlköpfig? - Nein.
- Ja.
- Es gibt keinen König von Hamburg!

Nur die letzte Antwort ist richtig. Leider ist in der formalen Logik jedoch über etwas nicht Existentes jede Aussage wahr. Es muß also eigens die Existenzpräsupposition zu der Satzbedeutung hinzugefügt werden, so daß ein Satz mit einer Präsuppositionsverletzung falsch wird. Allerdings ist dies Vorgehen nicht angemessen, da der Grund für die Ungültigkeit des Satzes nicht erkennbar ist: war die Präsupposition verletzt oder war die Aussage falsch? Präsuppositionen sind Aussagen, die wahr sein müssen, damit es überhaupt eine wahre Antwort geben kann! Ein Beispiel für Datenbankanfragen führt auch zum Problem der Präsuppositionsverletzung. Nehmen wir an, der Benutzer möchte wissen, welche Mitarbeiter einen bestimmten Kurs besucht haben. Die Datenbank hat zwei Tabellen: Mitarbeiter und Kurse. Jeder Mitarbeiter ist durch einen Namen eindeutig gekennzeichnet, jeder Kurs durch eine Nummer.

| Mitarbeiter | | Kurse | |
|---|---|---|---|
| Name | Teilnahme | Nr | Titel |
| Müller127 | 13 | 13 | Compilerbau |
| … | … | … | … |

Der Benutzer möchte alle diejenigen Mitarbeiter ausgegeben haben, die den Kurs Betriebssysteme besucht haben.

Falls die Frage in der Form gestellt wird

```
SELECT Mitarbeiter.Name FROM Mitarbeiter,Kurse
 WHERE Kurs.Titel = Betriebssystem  AND  Mitarbeiter.Teilnahme = Kurs.Nr,
```

dann werden keine Mitarbeiter ausgegeben, auch wenn es gar keinen Kurs Betriebssysteme gab. Dies ist eine irreführende Antwort, weil der Benutzer dann glaubt, keiner der Mitarbeiter hätte den Kurs besucht. Auch hier muß also vor der eigentlichen Anfrage erst geprüft werden, ob es einen Kurs mit dem Titel "Betriebssysteme" gibt.

Aus diesen Gründen müssen wir das Auswertungsprogramm noch ergänzen, daß es zurückliefert, welches Literal der Klausel fehlgeschlagen ist. Dadurch kann der Antwortgenerierung die Information übergeben werden, die sie braucht, um für Präsuppositionsverletzungen und nicht erfüllte Aussagen unterschiedliche Antworten auszugeben. Man beachte, daß die Präsuppositionsbehandlung bei der Auswertung der semantischen Beschreibung vorgenommen wird und nicht bereits bei der Überführung in die semantische Repräsentation. Obwohl natürlich Wörter (Artikel oder andere Quantoren, Fragewörter, Verben, Adjektive) die Präsuppositionen einführen, ist die Präsuppositionsüberprüfung auf der Ebene der Pragmatik und nicht auf der der Syntak oder gar Morphologie anzusetzen. Sie gehört auch nicht in die Semantik, da es um das Weltwissen von Sprecher und Hörer geht. Präsuppositionen nehmen Bezug auf die Überzeugungen von Sprecher und Hörer.

## 5.3 Konversationsimplikatur

*Moische kauft second-hand einen Pelzmantel. Zuhause stellt er fest, daß Flöhe im Pelz sind. Empört geht er zum Händler: "In dem Pelzmantel , den Sie mir verkauft haben, sind Flöhe!" - "Nona, das Bolschoi-Ballett werd´ich Ihnen hineinsetzen!"*

In dem Buch "Handlung, Kommunikation, Bedeutung", von Georg Meggle 1979 bei Suhrkamp herausgegeben, finden sich in deutscher Übersetzung 4 Aufsätze von Paul Grice. Dieser Wissenschaftler hat durch seine Schriften eine intensive Diskussion ausgelöst, wie es überhaupt kommt, daß Kommunikation mithilfe natürlicher Sprache möglich ist, wobei wir auch nicht-wörtliche Bedeutungen von Sätzen verstehen. Er verankert dabei das sprachliche Verhalten in allgemeinen Handlungsmaximen, die auch nichtsprachliches Verhalten steuern. Dies bedeutet, daß die Maximen nicht willkürliche Konventionen sind, die man auch ändern könnte, sondern daß Menschen ihnen unterworfen sind. Dies wiederum hat erhebliche Konsequenzen für die Gestaltung aller Mensch-Maschine-Systeme. Wenn Menschen sich unwillkürlich an die Handlungs-maximen halten bzw. jedes Verhalten im Lichter der Maximen interpretieren, so muß ein System sie ebenfalls befolgen, damit BenutzerInnen nicht in die Irre geführt werden.

Intentionale Kommunikation charakterisiert Grice so:

S meinte z durch das Aussprechen von U, wenn und nur wenn

1) S beabsichtigte, daß U im Rezipienten H einen Effekt z hervorrufe,

2) S beabsichtigte, daß 1) einfach dadurch erreicht würde, daß H diese Intention erkenne.

Die Intention einer Äußerung ist erfüllt, sobald sie erkannt ist. Das Erkennen ist die Intention. Die Kommunikation glückt, wenn im gegenseitigen Wissen (mutual knowledge) von Hörer und Sprecher die Intention des Sprechers vermerkt ist. Die Kommunikation ist nicht gelungen, wenn eine andere als die intendierte Bedeutung vm Hörer verstanden wurde. Es ist dann das Wissen des Höreres über die Intention des Sprechers und das Wissen des Sprechers über seine Intention verschieden. Mit dem Effekt ist z.B. nicht

das Befolgen eines Wunsches gemeint, sondern lediglich das Erkennen des Wunsches -- der dann auch abgelehnt werden kann.

Wie aber hängt nun der Effekt z mit der wörtlichen Bedeutung von U zusammen? Oft ist der zusammenhang nicht unmittelbar, sondern verlangt einige Schlußfolgerungen.

**Def.:** Ein Satz, geäußert in einem Kontext, impliziert andere Aussagen. Diese Implikationen werden **Konversationsimplikationen** genannt. Die Theorie der Konversation, die sich mit diesen Implikationen befaßt, ist die Konversationsimplikatur.

Als übergeordnetes Prinzip formulierte Grice das **Kooperationsprinzip**:

> "Mache deinen Gesprächsbeitrag jeweils so, wie es von dem akzeptierten Zweck oder der akzeptierten Richtung des Gesprächs, an dem du teilnimmst, gerade verlangt wird."
>
> (Paul Grice in Meggle (a.a.O.), S. 248)

Hierbei sind Zweck und Richtung des Gesprächs natürlich bezogen auf den jeweiligen Äußerungskontext. Wichtig ist dabei, daß nur der gegenseitig akzeptierte Zweck und nicht noch heimliche, nur einem Gesprächsteilnehmer bekannte Nebenzwecke berücksichtigt werden.

Als Maximen der Konversation (oder überhaupt Kommunikation) hat Grice die folgenden vier vorgestellt:

**Quantitätsmaxime**:

Mache deinen Beitrag so informativ wie (für die gegebenen Gesprächszwecke) nötig.

Mache deinen Beitrag nicht informativer als nötig.

**Qualitätsmaxime**:

Mache deinen Beitrag so, daß er wahr ist.

Sage nichts, was du für falsch hältst.

Sage nichts, wofür angemenessene Gründe fehlen.

**Relationsmaxime**:

Sei relevant.

**Modalitätsmaxime**:

Vermeide Unklarheit der Formulierung.

Vermeide Mehrdeutigkeiten.

Fasse dich kurz

Sei methodisch -- sage alles der Reihe nach.

Man kann leicht den Eindruck gewinnen, Grice habe es des öfteren mit weitschweifigen Gesprächspartner zu tun gehabt und seine Maximen seien Empfehlungen für gutes Dialogverhalten. Dies ist aber ein Mißverständnis. Die Maximen sind nicht als Empfehlungen für Sprecher gedacht, sondern als Beschreibungen dessen, was Hörer zur Intepretation von Äußerungen heranziehen. So gelingt z.B. die Lüge nur, weil Hörer einem Sprecher unterstellen, daß er/sie die Qualitätsmaxime befolgt. Wäre dies nicht der Fall, könnte keine Lüge gelingen.

Wenn jemand einem anderen hilft, ein Auto zu reparieren und es werden 4 Schrauben benötigt, so wird gemäß der Quantitätsmaxime auch erwartet, daß 4 und nicht 10 Schrauben angereicht werden. Wenn zur

Bearbeitung eines Problems die Information nötig ist, wo sich jemand befindet, so wird erwartet, daß nur die Ortsangabe, nicht jedoch auch noch ein gemeinsames Ferienerlebnis geäußert wird.

Die Relationsmaxime wird herangezogen, um z.B. die folgende Antwort auf die Frage, wo Michael ist, zu verstehen:

Vor Susannes Haus steht ein Motorroller.

Wir brauchen gar nicht zu wissen, daß Michael einen Roller fährt. Wir erfahren es aus der Antwort. Allerdings nur dann, wenn wir davon ausgehen, daß die Tatsache, daß ein Roller vor der Tür steht, etwas mit dem Aufenthaltsort von Michael zu tun hat. Unser allgemeines Weltwissen sagt uns, daß jemand sein Fahrzeug vor er Tür abstellt, wenn er ein Haus betritt. Deshalb ist das Vorhandensein eines Fahrzeugs vor der Tür ein Hinweis darauf, daß der Fahrer im Haus ist. Dies ist allerdings kein zwingender Schluß. Wen aber jemand eben auf die Frage nach Michaels Aufenthaltsort damit antwortet, daß ein bestimmtes Fahrzeug vor einer bestimmten Tür steht, so interpretieren wir:

Michaels Fahrzeug ist ein Motorroller.
Der Sprecher weiß nicht mit Sicherheit, ob Michael bei Susanne ist.

Wir konstruieren einen Zusammenhang zwischen der Frage und der Antwort, weil wir davon ausgehen, daß die Antwort mit der Frage zu tun haben muß. Was sich konstruieren läßt, hängt sicherlich mit unserem Weltwissen zusammen. Der Zusammenhang selbst muß aber nicht dem Hörer bekannt sein, damit er die Äußerung verstehen kann. Dies ergänzt unsere Diskussion, wieviel Weltwissen von vornherein vorhanden sein muß, damit natürliche Sprache verarbeitet werden kann, und wie Weltwissen durch Sprache übermittelt wird.

Wenn auf die Frage nach dem Ort, an dem sich Michael befindet, geantwortet wird:

In Südfrankreich.

so interpretieren wir dies mit Fug und Recht so, daß die Antwortende den genauen Ort nicht weiß oder glaubt, daß es bei der Frage nur darum geht, zu wissen, ob Michael erreichbar ist (und das ist er nicht). Wir wenden bei der Interpretation der Antwort die Modalitätsmaxime an.

Die Erwartung, daß ein Spercher sich an die Maximen hält, wird zum Verstehen seiner Äußerungen herangezogen. Dennoch kann es natürlich sein, daß er sich nicht daran hält. Grice nennt die folgnden 4 Arten, die Maximen nicht zu befolgen:

**Verletzung** der Maximen: Ein Sprecher kann bewußt irreführen oder lügen.

**Aussteigen** aus der Konversation: Ein Sprecher kann sich weigern, zu kooperieren. "Ich sage nichts!"

**Kollision** von Maximen: Manchmal stehen die Maximen Qualität und Quantität im Widerspruch, weil gerade die notwendige Information nicht abgesichert ist. Ein Beispiel, die Maximen dennoch zu befolgen war das Beispiel oben: der Hinweis auf Michaels Roller vor Susannes Tür.

**Ausbeutung** der Maximen: Ein scheinbarer Verstoß gegen Maximen, bei dem aber Verletzung, Aussteigen und Kollision ausgeschlossen ist, führt dazu, daß eine Interpretation gesucht wird, die auf einer tieferen Ebene die Befolgung der Maximen bedeutet.

Gerade die Ausbeutung von Maximen ist der Schlüssel zum Verständnis von Ironie. Nur weil wir von der Geltung des Kooperationsprinzips ausgehen, sind wir bereit, umfangreiche Inferenzen vorzunehmen, wenn offenkundig eine Maxime verletzt ist. Wir wandeln die wörtliche Bedeutung durch direkte Negation oder durch Abstraktion von einigen Merkmalen ab, um so zu einer die Maximen befolgenden Äußerung zu gelangen. Auf diese Weise kann die Äußerung

In den Kaufhäusern lief wieder diese bezaubernde Weihnachtsmusik und hat mich beim Einkauf entspannt.

interpretiert werden als:

Die scheußliche Kaufhausmusik zu Weihnachten ist mir beim Einkaufen auf die Nerven gegangen.

Eine besondere Eignung für Konversationsimplikationen haben die Nachfolgeradjektive. Dies sind Adjektive, die den Normalzustand bezeichnen. Deshalb können sie nur verwendet werden, wenn der Normalzustand zuvor gestört war. Während man also in Einklang mit den Maximen sagen kann

Der Kapitän war heute betrunken.

ist der Eintrag im Logbuch

Der Kapitän war heute nüchtern.

dank der Ausbeutung der Maximen eine kräftige Beschuldigung des Kapitäns. Das Nachfolgeadjektiv nüchtern impliziert vorherige Trunkenheit, da es nicht der Quantitätsmaxime entspricht alle Normalzustände zu verbalisieren. Metaphorische Bedeutungen werden in ähnlicher Weise kreiert:

Königin Victoria war aus Eisen.

Da jeder weiß, daß eine Frau natürlich aus Fleisch und Blut und nicht aus Eisen ist, müssen andere Eigenschaften des Eisens auf Königin Victoria übertragen werden.

Die Formalisierung der Konversationsmaximen ist schwierig. Wenn Inferenzen nicht mehr durch die Wort- und Satzbedeutung beschränkt sind, sondern auch noch deren Gegenteil oder einige Ausschnitte (welche?) umfassen sollen, können sie leicht unendlich werden. Wir müssen also Repräsentationen finden, die nötige Implikationen erlauben, aber auch einige Implikationen abblocken. Einige solcher Repräsentationen werden hier besprochen.

Bei Begriffen kann angegeben werden, ob sie einen Normalzustand angeben. So bezeichnen alle Nachfolgeradjektive Normalzustände für die Dinge, auf die sie sich beziehen können: sauber, nüchtern, heil, gesund, ... Wenn wir die Semantik von Adjektive durch eine Dimension und Extrempunkte repräsentieren, so kann bei den Extrempunkten angegeben werden, welcher davon der Normalzustand ist.

Bei Quantoren und Begriffen können Subsumtionsbeziehungen angegeben werden. So subsumiert der Quantor mindestens 1 die Quantoren mindestens 2 und mindestens 3. Der Quantor einige subsumiert den Quantor alle. Der Begriff Säugetier subsumiert den Begriff Mensch. Der Begriff mögen subsumiert lieben. Der Begriff warm subsumiert heiß. Die Konversationsmaximen fordern, daß der speziellste zutreffende Quantor oder Begriff verwendet wird. Daher folgern wir das Nichtzutreffen aller von dem geäußerten Begriff (Quantor) subsumierten Begriffe(Quantoren). Wenn wir das begriffliche Wissen in Form einer TBox abgelegt haben, ist dies eine einfache Schlußfolgerung.

Leider ist dies noch nicht alles. Wie behandeln wir Äußerungen, die eine Definition verbalisieren, obwohl es einen sprachlichen Ausdruck für das Definierte gibt?

Greife die Klinke, drücke sie herunter und ziehe die Tür zu dir - danke!

Einerseits kann dies unterstellen, der Hörer würde die Definition nicht kennen. Andererseits kann es unterstellen, der Hörer würde die Tür nicht öffnen, wenn er darum gebeten wird. Hier ist die Definition der handlung "Tür öffnen" vollständig angegeben. Ein besonderer Fall liegt vor, wenn die Handlung fast vollständig definiert wird.

A. gab eine Folge von Tönen von sich, wie in den Noten des Rigoletto für Gilda angegeben.

Wieso heißt es nicht einfach: A. sang die Gilda im Rigoletto? Offensichtlich gibt es noch ein Merkmal, das eben nicht erfüllt war, vermutlich der Wohlklang. Unser Problem bei der Formalisierung ist hier: soll jede fast vollständige Definition als Oberbegriff eingetragen werden, so daß dann das allgemeine Verfahren angewandt werden kann? Soll ein Prozeß implementiert werden, der bei jeder Definition ein Merkmal weglassen kann und dadurch einen neuen Begriff formt? Beide Alternativen sind aufwendig! Die zweite Alternative hat immerhin den Vorteil, daß sie nur bei Bedarf, also im Falle des Scheiterns der wörtlichen Bedeutung zur Anwendung kommt. Der Prozeß muß aber beschränkt werden. Tatsächlich gibt es eine versteckte Maxime: sage Unerfreuliches nicht direkt, sondern laß es den Hörer selbst inferieren. Die Bewertung von Aussagen kann also ausgenutzt werden bei der Beschränkung der Suche nach den Merkmalen, die wegzulassen sind. Dies bedeutet allerdings, daß Bewertungen repräsentiert werden müssen. Diese Bewertungen sind obendrein subjektiv, d.h. unterschiedlich bei verschiedenen Menschen. (Ansätze zur

Repräsentation und Ausnutzung subjektiven Wissens werden in Katharina Morik "Überzeugungssysteme der Künstlichen Intelligenz", Tübingen, 1982 dargestellt.)

Wir sehen, daß einerseits die Pragmatik die Semantik entlasten kann: ironische oder metaphorische Lesarten müssen nicht bei der Wortsemantik angegeben werden, sondern können einem pragmatischen Prozeß von Schlußfolgerungen überlassen bleiben. Andererseits stellt dieser Prozeß Anforderungen an die Repräsentation der Semantik: Abstufungen von Begriffen, ihre Bewertung und Normalität müssen angegeben sein.

## 5.4 Sprechhandlungen

In der Einleitung wurde bereits darauf hingewiesen, daß Äußerungen auch Handlungen sind. John Langshaw Austin hat mit seinem Buch "How To Do Things With Words", das 1962 in Oxford und 1972 in deutscher Übersetzung in Stuttgart herauskam, Äußerungen in eine allgemeine Handlungstheorie einzubinden versucht. John Searle führte mit seinem Buch "Speech Acts" (Cambridge 1969, deutsch: Frankfurt a.M. 1971) die Arbeit fort. Obwohl diese Ausformung der pragmatischen Theorie gegenwärtig nicht mehr verwendet wird, soll hier kurz vorgestellt werden, was Sprechakttheorie ist.

### 5.4.1 Sprechakttheorie

Manche Handlungen können überhaupt nur sprachlich durchgeführt werden: wetten, taufen, den Krieg erklären, zum Ritter schlagen, Einspruch erheben, warnen sind Handlungen, die (zumindest unter anderem) sprachlich vollzogen werden. Die entsprechenden Verben sind performative Verben. Aber nicht nur performative Verben stehen für sprachliche Handlungen. Mit einer Äußerung werden immer auf einmal drei verschiedene Handlungen vorgenommen:

**lokutiver Akt**: der Sprecher äußert eine Folge von Wörtern.
Beispiel: Eine Frau sagt zu zwei Kindern, auf die ein Hund zuläuft:"Der Hund beißt nicht!"

**illokutiver Akt**: der Sprecher vollzieht mit seiner Äußerung eine Sprechhandlung.
Beispiel: Die Frau informiert die Kinder darüber, daß der Hund nicht beißt.

**perlokutiver Akt**: mit der Sprechhandlung wird eine bestimmte Wirkung beim Hörer erzielt.
Beispiel: Die Frau beruhigt die Kinder, die vielleicht Angst haben, daß der Hund sie beißen könnte.

Während illokutive Akte durch performative Verben ausgedrückt werden können, ist der Gebrauch von perlokutiven Verben nicht dazu angetan, den entsprechenden Akt auszuführen:

* Ich überrede dich hiermit, daß du morgen kommst.
(Ich möchte dich hiermit überreden, daß du morgen kommst.)

Damit Sprechhandlungen glücken, müssen bestimmte Bedingungen erfüllt sein. Dies sind die Gelingensbedingungen (felicity conditions). Sie betreffen gegenseitiges Wissen, Wünsche, soziale Positionen und Bewertungen von Zuständen. Wir können also Handlungsbeschreibungen erstellen wie z.B. die beiden folgenden:

| Bedingungen | BITTEN | WARNEN |
|---|---|---|
| propositionaler Gehalt | künftige Handlung A von H | künftiges Ereignis E |
| Voraussetzungen | S glaubt, H kann A tun | S glaubt, E wird geschehen |
| | S glaubt, daß H gerade nicht A tut | S glaubt, daß E nicht im Interesse von H ist |
| | S glaubt, daß H nicht ohnehin schon vorhat, A zu tun | S glaubt, daß H nicht weiß, daß E geschehen wird |
| Aufrichtigkeit | S will, daß A von H getan wird | S glaubt, daß E nicht in Hs Interesse ist |

| | | |
|---|---|---|
| essentielle Bedingung | S intendiert, daß A von H getan wird | S intendiert, daß H weiß,<br>daß E geschehen wird und<br>daß S weiß, daß H das Ereignis E negativ bewertet |

Wir sehen bereits an diesem Beispiel, daß sich Sprechakttheorie und Konversationsimplikatur überschneiden. Wir sehen weiterhin, daß performative Verben als Bezeichnungen illokutiver Akte betrachtet werden. Sie geben die illokutive Kraft an, die mit einer Äußerung verbunden ist. Die drei Satztypen Aussage, Frage und Befehl werden wie performative Verben betrachtet als Explikation eines illokutiven Aktes. Eine Äußerung kann einen illokutiven Akt darstellen, ohne daß das performative Verb oder die betreffende Satzart verwendet wird. Sie muß lediglich den propositionalen Gehalt verbalisieren.

Die Erkennung der Handlung, die mit einer Äußerung ausgeführt wird, muß also aus der semantischen Repräsentation (für den propositionalen Gehalt) und dem aktuellen Kontext (z.B. H tut nicht bereits A) auf die Intention von S schließen. Dazu muß eine Repräsentation ähnlich der Tabelle für alle Sprechhandlungen vorliegen. Die abgeleitete Intention von S kann dann in das wechselseitige Wissen von Sprecher und Hörer eingetragen werden. Der illokutive Akt ist erkannt.

Das Problem hierbei ist, daß die Intention von S sich auf das Wissen von S bezieht, das von dem von H verschieden sein kann. Wenn H tatsächlich bereits A tut, sollte die Bitte, die Tür zu schließen, dennoch als solche erkannt werden. Wenn aber das S zu unterstellende Wissen nicht (für H) der Fall ist, ist die Suche in der Tabelle der Sprechhandlungen nicht beschränkt. Anders ausgedrückt: für die Erkennung der Sprechhandlungen verwenden wir ein Wissen, das unter Umständen erst durch die erfolgreiche Erkennung der Sprechhandlung gewonnen wird!

Gerade NLS, die Prozesse wie die Erkennung illokutiver Akte ja operational formalisieren müssen, decken solche Schwierigkeiten bestimmter linguistischer Theorien auf.

### 5.4.2 Das KAMP System

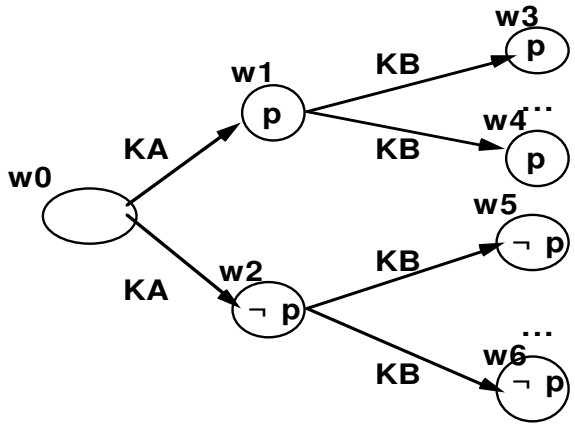
Douglas Appelt hat in seiner Dissertation 1982 versucht, den Zusammenhang zwischen nicht-sprachlichen und sprachlichen Handlungen zu beschreiben. Sprechakte werden wie andere Handlungen repräsentiert und eine Äußerung generiert, wenn zur Erreichung des Ziels ein Sprechakt nötig ist. Sein System KAMP (Knowledge and Modalities Planner) verwendet eine Logik möglicher Welten. Die reale Welt im aktuellen Zustand w0 enthält Aussagen, die gerade gelten. Das Wissen von Hörer und Sprecher wird durch das zweistellige Prädikat `know` repräsentiert, das den Akteur an erster und eine Proposition an zweiter Stelle hat. So bedeutet z.B. `true(w1, know (a, p))`, daß in der Welt `w1` der Handelnde `a` weiß, daß `p`.

Appelt legt nun zwei Kompatibilitätsannahmen zugrunde:

Wenn `a` in `w1` weiß, daß `p`, so sind alle Welten mit `w1` kompatibel, in denen laut `a` ebenfalls gilt, daß `p`.

Wenn `a` in `w1` weiß, daß `p`, so gilt in allen Welten, die mit `w1` kompatibel sind, daß `p`.

Die erste Annahme faßt alle Welten zusammen, in denen `a` weiß, daß `p`.Die zweite Annahme stellt den Zusammenhang von Wissen und tatsächlichem Zustand her: wenn jemand etwas weiß, dann ist es auch so. Nun kann ein Akteur wissen, daß jemand anderes etwas weiß, was er nicht weiß. Wenn `a` nicht weiß, ob `p`, so gibt es zwei Möglichkeiten: `p` gilt oder `p` gilt nicht. Jede Möglichkeit ist in einer anderen Welt abgelegt. Diese Welten sind nicht kompatibel! Wenn `a` glaubt, daß `b` weiß, ob `p`, dann sind von der Welt, in der `p` gilt nur solche Welten erreichbar, in denen `b` weiß, daß `p`. Ein Bild mag dies illustrieren:




Handlungen, ihre Voraussetzungen und Wirkungen werden jeweils als mögliche Welten repräsentiert. Ziele sind in KAMP Welten mit bestimmten Kompatibilitätsbedingungen. So gibt es zum Beispiel eine Handlung `act1`, die `a` tun kann und als Wirkung eine Welt hat, in der `a` weiß, ob `p`.Der Planungsmechanismus verläuft zum einen horizontal: für ein Ziel wird eine Handlung gefunden, deren Vorbedingungen die nächsten Ziele sind, die durch Handlungen erreicht werden, deren Vorbedingungen die nächsten Ziele werden, und so weiter, bis die Vorbedingungen einer Handlung in der aktuellen Situation gegeben sind. Eine Handlung kann expandiert (verfeinert) werden. Ist `act1` beispielsweise eine kommunikative Handlung, so kann sie weiter expandiert werden in eine Frage (nach `p`) und eine Antwort (ob `p`). Dies ist die vertikale Planung: eine Handlung wird in ihre Teilhandlungen zerlegt, die wiederum Vor- und Nachbedingungen haben können. Es wird hierarchisch geplant: zunächst wird horizontal eine Handlungsabfolge gefunden. Dann werden die Handlungen vertikal verfeinert. Eine Kritikkomponente überprüftt, ob der Plan auf der feinsten (ausführbaren) Ebene noch stimmig ist.

Das Beispiel von Douglas Appelt ist die bekannte SRI-Pumpe. Diese Pumpe soll in Kooperation von dem Roboter, Rob, und einem Menschen, John, demontiert werden. In der Startsituation befinden sich Rob und John am selben Ort. Nehmen wir an, Rob hat in seinem Plan für die Pumpendemontage das Ziel, daß die Teile `pu` und `pl` nicht mehr miteinander befestigt sind.

```
w0: loc(john) = loc(rob), attached(pu, pl).
???
wi: ¬ attached(pu,pl).
```

Es gibt eine Handlung, die von einer Welt mit zwei befestigten zu einer Welt mit zwei nicht mehr befestigten Teilen führt. Diese Handlung, `remove`, erfordert einen Akteur, der sich am selben Ort wie die Teile befindet und bereit ist, die Handlung auszuführen.

```
wj: attached(X, Y), loc(A)=loc(Y), wantsTo(A, remove(X,Y))
-remove(A,X,Y)-
wj+1: ¬attached(X,Y)
```

Es gibt eine Handlung, die jemanden dazu bringt, daß er bereit ist, eine Handlung auszuführen, `request`. Für diese Handlung muß der Fragende und der Gefragte am selben Ort sein. In der Nachfolgewelt will der Adressat der `request`-Handlung das, was Gegenstand der `request`-Handlung war.

```
wj: loc (X)=loc(Y)
-request(X, p)-
wj+1: wantsTo(Y,p)
```

Horizontal wurde also geplant, daß Rob zunächst John dazu bringt, daß er die Verbindung von `pu` und `pl` lösen will, und daß dann John diese Handlung ausführt. Dies ist der grobe Plan, der nun verfeinert werden muß. `request` ist ein Sprechakt. Ein illokutiver Akt ist eine Handlung, deren Vorbedingungen das Wissen und die Intention von Sprecher und Hörer betreffen, und die Wissen produziert. Sprechakte sind direkt mit Satztypen assoziiert: Informieren mit Aussagesatz, Aufforderungen mit Befehlssätzen, Fragen mit Fragesätzen. `request` ist direkt mit dem Satztyp Befehlssatz verknüpft. Der propositionale Gehalt ist noch nicht vollständig festgelegt. Bis zu diesem Punkt der Planung ist nur `remove(john,pu,pl)` bekannt. Die Befehlssatzstruktur S --> V NP wird weitergeführt zu `V --> remove`.

Es muß nun geplant werden, wie `pu` und `pl` zu kommunizieren sind. Die Verbalisierung von Dingen muß dergestalt sein, daß der Hörer das gemeinte Ding identifizieren kann. Diesen Schritt nennt Appelt *concept activation*. Nehmen wir an, Rob weiß, daß John die Pumpe kennt. Dann reicht eine definite Nominalphrase zur Verbalisierung aus. `NP --> Det N` und `Det --> the` und `N--> pump`. Aber was ist mit `pl`? Es handelt sich um die Plattform, auf die die Pumpe montiert ist. Da John dies gemäß Rob weiß und da remove keine notwendige Ergänzung hat, braucht `pl` gar nicht verbalisiert zu werden.

Jetzt wird ein Gesamtplan erstellt. Dazu wird das Vorwissen des Hörers genauso berücksichtigt wie der Plan für die zu verbalisierende Handlung. Im Beispiel wird ein Plan erstellt, wie `remove(john,pu,pl)` auszuführen ist. Diese Verfeinerung ist:

```
wj: knowsWhatIs(john, b1), loc(john) = loc(pl), have(john, b1)
-unfasten(john, b1, pu, pl)-
wj+1:¬attached(pu,pl)
```

Auf dieser detaillierteren Ebene wird wieder horizontal geplant. Wenn gemäß Robs Wissen über Johns Wissen klar ist, daß die Voraussetzung `knowsWhatIs(john, b1)` in der aktuellen Welt nicht erfüllt ist, so wird eine Handlung, `inform(john, b1=wr1)`, gefunden, die als Nachbedingung eben dieses Wissen hat. Das Werkzeug `b1` ist ein Schraubenschlüssel (*wrench*). Auch diese Handlung ist wieder ein illokutiver Akt. Dieser Akt ist mit dem Satztyp Aussagesatz assoziiert. Da es viele Schraubenschlüssel gibt, muß dieser gekennzeichnet werden. Eine Möglickeit zu kennzeichnen ist eine Ortsangabe. Der Schraubenschlüssel ist in der Werkzeugkiste.

Der Gesamtplan wird nun geprüft, ob mehrere Handlungen dieselbe Nachbedingung haben (*Subsumtpionstest*). In unserem Beispiel ist dies der Fall: die Verbalisierung des Referenzobjektes `b1` beinhaltet gerade die Informationen, die John braucht, um die `unfasten`-Handlung ausführen zu können. Der Plan wird also verkürzt. Es wird an den bereits teilweise generierten Satz eine Präpositionalphrase gehängt, die das Referenzobjekt `b1` durch Nennung seiner Klassenzugehörigkeit (Schraubenschlüssel) und seines Ortes (in der Werkzeugkiste) kennzeichnet und dabei gleichzeitig die für die Handlung nötige Information ausdrückt. Das KAMP-System gibt den Satz aus:

```
remove the punp with the wrench in the tool-box.
```

Die Verbindung zwischen sprachlichen und nicht-sprachlichen Handlungen ist in KAMP durch die Repräsentation des Wissens von Sprecher und Hörer möglich geworden. Sprachliche Handlungen haben als Vor- und Nachbedingungen Wissen von Hörer und Sprecher. Die Verbindung zwischen sprachlicher Handlung und Äußerung ist jeweils vorgegeben: wenn ein Befehl geäußert werden soll, wird das Satzmuster Befehlssatz gewählt; wenn ein Ort verbalisiert werden soll, wird eine Präpositionalphrase mit Ortspräposition gewählt; wenn ein Werkzeug für eine Handlung verbalisiert werden soll, wird eine Präpositionalphrase mit with gewählt. Die Einführung von Referenten wird sorgfältig geplant: es wird geprüft, ob das Wissen, das durch eine Äußerung entsteht oder durch den Plan für eine nichtsprachliche Handlung gegeben ist, ausreicht, um das Referenzobjekt zu identifizieren.

## 5.5 Dialogstruktur

Bisher haben wir nur einzelne Sätze behandelt. Damit wird auch nur ein Gesprächsteilnehmer -- wenn auch mit seinem Wissen über die anderen Gesprächsteilnehmer -- berücksichtigt. Sprache beruht aber wesentlich darauf, daß mehrere SprecherInnen beteiligt sind. Deshalb soll hier kurz angerissen werden, mit welchen Phänomenen man es zu tun hat, wenn man Dialoge beschreiben will.

### 5.5.1 Sprecherwechsel

Es ist erstaunlich, wie wenig Überlappung es bei Gesprächen gibt. Selbst wenn eine Gruppe von Menschen miteinander redet, beträgt der Anteil, in dem mehr als ein Sprecher das Wort hat, höchstens 5 %. Dabei dauern die Pausen zwischen den Redebeiträgen nur wenige Mikrosekunden (Levinson, a.a.O. S.295). Offensichtlich gibt es ein funktionierendes System für den Sprecherwechsel, das unbewußt ist und von allen Menschen geteilt wird. Dies System ist unabhängig von Blickkontakten. Es beruht auf Einheiten des Gesprächs, die offensichtlich für alle Hörer leicht zu erkennen sind. Sacks, Schegloff und Jefferson haben in ihrer Schrift "A Simplest Systematics foir the Organization of Turn-Taking in Conversation", publiziert 1974 in *Language*, ein einfaches Modell vorgeschlagen. Ein Redebeitrag (turn) ist die kleinste Einheit in einem

Gespräch. Sprecherwechsel können nur nach einem Redebeitrag stattfinden. Hierbei gelten die folgenden Regeln:

- Der aktuelle Sprecher kann den Nachfolgesprecher auswählen, z.B., durch eine Frage mit Anrede. Dann hört der Sprecher auf und der genannte Nachfolger muß etwas sagen. Hier gilt auch sein Schweigen als Redebeitrag.
- Wenn der Sprecher keinen Nachfolger auswählt, kann jeder sich selbst wählen. Wer zuerst spricht, behält das Wort.
- Wenn kein anderer spricht, kann derselbe Sprecher den nächsten Redebeitrag halten.

Wieder gilt wie schon bei den Konversationsmaximen, daß diese Regeln das beobachtete Verhalten beschreiben. Sie sind nicht präskriptiv, sondern deskriptiv. Das Problem besteht darin, die Einheit des Redebeitrags zu definieren. Ein Satz? Ein Sprechakt? Gedankengänge? Sätze sind deshalb eine gute Einheit, weil ihr Ende klar vorhersagbar und deutlich markiert ist durch die Grammatik. Selbst wenn man dem Gedankengang nicht folgen konnte, weiß man, wann der Satz aufhört. Selbst wenn man nicht feststellen konnte, welchen illokutiven oder gar perlokutiven Akt der Sprecher äußern wollte, kann man das Satzende bemerken und sich mit einer Klärungsfrage zu Wort melden.

Es gibt stereotype Sequenzen, die aus mehreren Teilen bestehen. So verlangt eine Frage nach einer Antwort, ein Gruß einen weiteren Gruß, eine Einladung Annahme oder Ablehnung, ... Solche Nachbarschaftspaare sind nicht zwingend. Der erste Teil etabliert aber eine Erwartung. Eine dreiteilige Sequenz ist Aufruf, Antwort und Anliegen. Die ersten beiden Teile etablieren die Verbindung zwischen Sprecher und Hörer.

| | | | |
|---|---|---|---|
| Mama? | John? | klopf, klopf, klopf | Telefonklingeln |
| Ja, was ist? | Ja? | Herein! | Hallo? |
| ich brauche einen Stift! | reichst du mal das Salz? ... | | Hallo! |
| | | | Oh, hallo! |

Erst wenn die Verbindung etabliert wird, wird das erste Thema eingeführt. Oft geht noch eine kurze Sequenz zur Etablierung eines sozialen Kontaktes voraus (Wie geht's?). Diese Sequenz wird kurz gehalten, damit sie nicht als das erste Thema mißverstanden wird. Das erste Thema dominiert das Gespräch, wird als das zu bearbeitende Thema angesehen. Es kann nicht ohne gegenseitiges Akzeptieren verlassen oder beendet werden. Wer das Thema einführte, kann um Beendigung bitten. Ein anderer Gesprächsteilnehmer muß die Bitte akzeptieren. Natürlich werden diese Bitten und ihr Erfüllen implizit geäußert. So wird z.B. festgestellt, daß ein gemeinsamer Plan erstellt wurde. (Dann treffen wir uns also morgen um 1 in der Mensa, ja?) Es wird das Gespräch zu dem ersten Thema zusammengefaßt. (Na, gut, danke dir.) Es wird ein Übergang zu einem anderen Thema vorgenommen. Dies andere Thema kann aufgegriffen werden oder eben nicht. Die themeneinführende Sprecherin ist wie festgenagelt, wenn ihrer Bitte um Beendigung oder Wechsel nicht stattgegeben wird. Es folgen vermutlich immer weitere Bitten. Wird stattgegeben (Ja, gut), wird dies oft auch noch einmal bestätigt.

### 5.5.2 Dialogsteuerung

Die thematische Organisation von Dialogen (und Texten) umfaßt größere Einheiten als die Redebeiträge. In der KI wurde die thematische Organisation und die Erkennbarkeit von (thematischen) Einheiten von Rachel Reichman untersucht. Sie hat die Einbettungen von Sequenzen in andere Sequenzen und die Markierung einer Sequenz durch Hinweiswörter (*clue words*) beschrieben und an vielen Tonbandaufzeichnungen belegt. Thematische Einheiten sind als Stapel (*stack*) organisiert. Ein Einschub wird auf den Stapel gelegt. Sobald er abgearbeitet ist, kommt das darunterliegende Thema wieder an die Reihe.

Bill Mann hat eine Theorie rhetorischer Strukturierung entwickelt. Dabei gibt es in einem Dialog immer ein Schema mit genau einem Nukleus und mindestens einem Satelliten. Ein Satellit kann wieder ein Schema sein. Relationen verbinden Satelliten mit dem Nukleus. Ein Nukleus kann ein Sprechakt sein. Satelliten können Vorbedingungen zu einer im Nukleus angegebenen Handlung sein. Ein rhetorisches Schema ist z.B. das folgende:

Nukleus: Aufforderung anzurufen
Satellit1 -- Relation: Motivierung für den Anruf
Satellit2 -- Relation: Erfüllen einer Voraussetzung für den Anruf

Es kann realisiert werden durch:

Ruf mich an! Ich habe eine Überraschung für dich. Meine Durchwahl ist 881.

Das Problem bei diesem Ansatz ist, die Satelliten und die verbindenden Relationen so zu definieren, daß sie auf Äußerungen eindeutig anzuwenden sind. Dasselbe Problem hat auch ein weiterer Ansatz von Bill Mann, das Dialogspiel. Ein Dialogspiel soll für Dialoge sein, was Sprechakte für die Äußerungen eines Sprechers sind: eine funktionale Einheit, die in vielfältiger Weise sprachlich realisierbar ist. Ein Dialogspiel vergibt zwei Rollen: Initiator, I, und Erwiderer, R. Beide haben Ziele. Konventionen regeln das Dialogspiel:

I hat das Ziel IP.
I glaubt, daß er/sie das Ziel erreichen kann.
I hat das Recht, das Dialogspiel zu spielen.
I hat das Recht, IP zu erreichen.
IP ist noch nicht erreicht.
R will seine/ihre Ziele erreichen.

Zusätzlich zu diesen generellen Regeln, gibt es zu jedem Dialogspiel noch spezielle Vorbedingungen. Bei der Diskussion ist das Ziel von I, daß R hinterher glaubt, daß p. Das Ziel von R ist, daß R rechtfertigt, wieso R vielleicht nicht glaubt, daß p. Die Vorbedingung ist, daß I glaubt, daß p und R glaubt nicht, daß p. Bei der Frage gibt hat I das Ziel, p zu wissen. R hat ebenfalls das Ziel, daß I p weiß. Die Vorbedingung ist, daß R p weiß. Ein Dialogspiel wird durch eine Bitte eingeleitet und durch eine Bitte beendet.

| | |
|---|---|
| Ich bin hungrig. | Bitte um ein Dialogspiel "Erlaubnis" |
| Hast du deine Hausaufgaben schon gemacht? | Vorbedingung für "Erlaubnis" und Bitte um "Frage" |
| Ja. | akzeptierte "Frage" |
| Was gibt es zu essen? | Bitte um ein Dialogspiel "Erlaubnis" |
| Zeig sie mir. | Fortsetzung der Vorbedingung für "Erlaubnis" |
| Was ist die Hauptstadt von Brasilien? | Bitte um "Prüfen" |
| Brasilia. | akzeptierte "Prüfung" |
| Kann ich jetzt etwas essen? | Bitte um "Erlaubnis" |
| Willst du Bratkartoffeln? | akzeptierte "Erlaubnis", Bitte um "Frage" |
| Klar! | akzeptierte "Frage", Bitte um Beendigung |
| OK | akzeptierte Beendigung. |

Natürlich ist "Erlaubnis" hier ein Dialogspiel und nicht als die Erlaubnis selbst, etwas zu essen, mißzuverstehen. Der inhaltliche Zusammenhang zwischen Hausaufgaben, Essen und Bratkartoffeln liegt nicht in der Sprache, sondern in Handlungen und Planungen für Handlungen begründet. Die Form der Äußerung zeigt weder an, um welches Dialogspiel es sich handelt, noch läßt sich das Bitten, Akzeptieren und Ablehnen an der sprachlichen Form festmachen.

### 5.6 Anaphora

Eine andere Möglichkeit, über Sätze hinauszugehen, ist es, den Rückbezug auf vorhergehende Sätze -- egal, von wem sie geäußert wurden -- zu betrachten. Die Grundannahme bei Texten und Dialogen ist die Kohärenz, d.h. die Annahme, daß alle Äußerungen miteinander zusammenhängen. Dabei kommen eben jene Relationen ins Spiel, die wir oben zwischen Nukleus und Satellit angenommen haben.

Gestern fand eine Hochzeit statt. Die Braut trug schwarz.

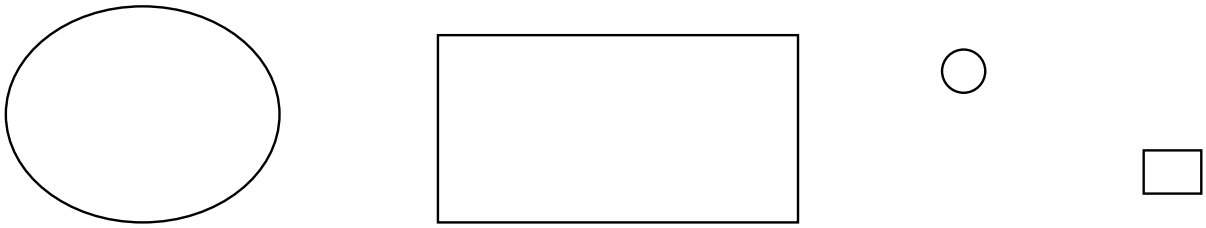
Peter hat sich einen Arm gebrochen. Dieses Unglück passierte ausgerechnet an seinem Geburtstag.

Gestern ist ein Unglück passiert. Peter hat sich seinen Arm gebrochen.

Bernd ist hungrig. Er sucht eine Imbißbude.

Bernd hat eine Imbißbude gefunden und kauft sich Pommes frites.

Hier sehen wir sprachliche Rückverweise. Solche Verweise heißen **Anaphora**. Verweise nach vorn heißen Kataphora. Die meisten operationalen Beschreibungen behandeln nur einen Ausschnitt von Anaphora. Es wird der referentielle Gebrauch von Kennzeichnungen behandelt. Der referentielle Gebrauch einer Bezeichnung dient dazu, einen bestimmten Gegenstand oder Sachverhalt so zu charakterisieren, daß der Hörer dieses Referenzobjekt ermitteln kann. Anaphora beziehen sich auf diesen einmal ermittelten Gegenstand. Sie werden in einem Satz gebraucht, in dem weitere Aussagen über das Refernzobjekt gemacht werden. Die Eignung einer Kennzeichnung dafür, daß der Hörer das Referenzobjekt ermitteln kann, wird bestimmt durch das Wissen von Sprecher und Hörer, insbesondere das wechselseitige Wissen, sprachliches Wissen und Eigenschaften der Objekte. Wenn z.B. Sprecher und Hörer beide Sicht auf die folgenden Gegenstände haben,



so ist die Kennzeichnung großer Kreis bestens geeignet, aus den Objekten das gemeinte herauszufinden.

Der große Kreis ist ganz links. Das kleine Viereck ist ganz rechts. Er ist eigentlich orange.

Das sprachliche Wissen regelt, daß er sich nur auf einen genannten Kreis, nicht aber auf ein -- vielleicht ebenfalls genanntes -- Viereck beziehen kann. Man könnte annehmen, daß die Anapher einfach eine Konstituente substituiert: Der große Kreis wird durch Er substituiert .

Wenn das Baby das Gemüse nicht essen mag, mußt du es mit etwas Milch verdünnen.

Das Wissen von Sprecher und Hörer kann eine Anapher auflösen, bei der der Rückbezug syntaktisch nicht eindeutig ist. Da man Babies nicht verdünnt, wohl aber Speisen, kann sich die Anapher es nur auf das Gemüse beziehen. Darüberhinaus kann das Weltwissen ein Referenzobjekt einführen, das gar nicht sprachlich realisiert wurde. So werden Teile eines Gegenstandes durch dessen Nennung indirekt eingeführt. Der folgende Text ist durchaus möglich:

Ich habe vorhin eine Hochzeit gesehen. Sie ganz in Lila und er in gelbem Anzug.

Durch das Wort Hochzeit ist bereits Braut und Bräutigam eingeführt, so daß auf beide mit einem Personalpronomen Bezug genommen werden kann.

Anaphora müssen nicht immer Koreferenz ausdrücken. So ist bei

Hans wäscht sein Auto. Peter auch.

normalerweise nicht gemeint, daß Peter auch das Auto von Hans wäscht, sondern daß Peter sein eigenes Auto wäscht. Gemeint ist also: `wäscht(X, Y) & gehört(X,Y), wobei X=hans` im ersten und `X=peter` im zweiten Satz. Eine Anapher ersetzt also nicht einfach eine Wortfolge. Wir können nicht erst sein Auto durch das Auto von Hans ersetzen und dann auch durch wäscht das Auto von Hans. Vielmehr ist gibt es einen Ausdruck, in dem die Variablen an verschiedene Referenzobjekte gebunden werden. Der Ausdruck ist hier gleich geblieben, nicht der Referent!

Interessanterweise können Anaphora sich nur auf innerhalb eine Texteinheit eingeführte Objekte beziehen. Was aber ist so eine Texteinheit? Wir haben bereits die Nachbarschaftpaare und Einschachtelung von Nachbarschaftspaaren kennengelernt. Wir haben auch schon anhand des KAMP Systems den Bezug zu einer kooperativen Aufgabe gesehen. Barbara Grosz hat eben die schon bei KAMP behandelte Pumpe am SRI als Beispiel für eine kooperative Handlung gewählt. Ein Handlungsplan wird nach und nach verbalisiert.

Jede Teilaufgabe bildet eine Texteinheit, die während der Verbalisierung im Fokus ist. Innerhalb des aktuellen Fokus können Anaphora verwendet werden. Die Teilaufgaben sind dem übergeordneten Ziel des Gesamtplans untergeordnet. Entsprechend können Anaphora sich auch auf Objekte des Gesamtplans beziehen.

## 6. Einige NLS

Die Beschreibung von NLS (Natürlichsprachliche Systeme) ist gar nicht so einfach. Wenn wir sagen Das System NN verarbeitet Präsuppositionen, was meinen wir damit?

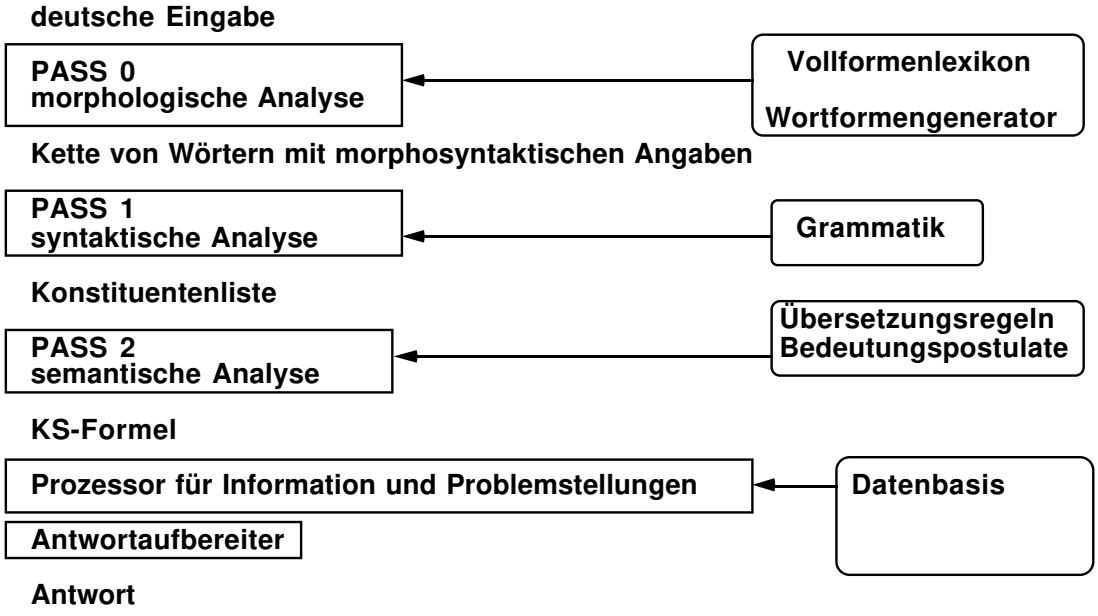
Das System NN akzeptiert eine Eingabe, die etwas enthält, was wir Präsuppositionen nennen.

Das System NN hat eine Präsuppositionskomponente, die einige Präsuppositionen als solche erkennt.

Wurde die Theorie, aus der die Bezeichnung des Phänomens stammt, auch tatsächlich als Leitlinie bei der Entwicklung des Systems gewählt? Werden alle Instanzen eines Phänomens behandelt? Wie können Grenzen eines implementierten Ansatzes gekennzeichnet werden? Dieser Schwierigkeiten eingedenk, sollen zwei Systeme kurz vorgestellt werden. Dabei wird keinesfalls eine Beschreibung der einzelnen Komponenten, sondern lediglich die Systemarchitektur und illustrierend jeweils eine Operationalisierung der Pragmatik dargestellt.

### 6.1 PLIDIS

PLIDIS wurde als Zugangssystem zu Daten genutzt, die über Wasserproben aus dem Neckar zur Überwachung der Abwässerverordnungen gesammelt wurden. Der Aufbau von PLIDIS (ProblemLösendes Informationssystem mit Deutsch als Interaktionssprache)[2] entspricht dem prototypischen Frage-Antwort-System. Das System wurde auf einer Siemens BS2000 in LISP implementiert.




PLIDIS verfügte über ein Lexikon mit 60 000 Einträgen. Ist ein Wort dem System unbekannt, so kann der Benutzer es in einem vom System geführten Dialog eintragen, wobei ein Wortformengenerator zu einer Stammform alle flektierten Formen generiert. Die Grammatik von PLIDIS deckt einen außerordentlich großen Teil der deutschen Syntax ab. Als Formalismus wurde das Augmented Transition Network (ATN) gewählt, ein Netzwerk, an dessen Kanten auch Prozeduren stehen können.

[2]Kolvenbach, Lötscher, Lutz "Künstliche Intelligenz und natürliche Sprache - Sprachverstehen und Problemlösen mit dem Computer" Forschungsberichte des Instituts für deutsche Sprache, Mannheim, 1979

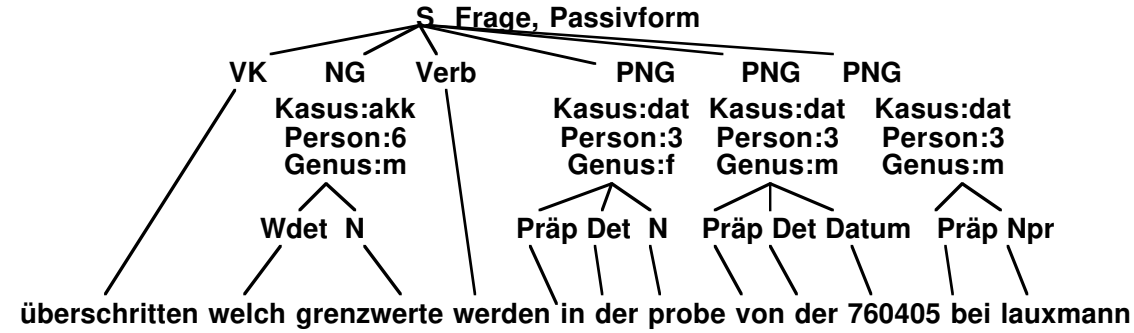
Die semantische Analyse besteht aus zwei Schritten. Im ersten Schritt wird die Konstituentenstruktur in eine oberflächennahe semantische Repräsentationssprache überführt. Hier haben zwei verschiedene Formulierungen für denselben Sachverhalt noch zwei verschiedene Formeln zur Folge. Im Gegensatz zu unseren Beispielen oben wird hier also strikt getrennt und nacheinander eine syntaktische Strukturbeschreibung erstellt, die dann durch Übersetzungsregeln in eine semantische Repräsentation überführt wird. Im zweiten Schritt verwendet ein Theorembeweiser Bedeutungspostulate, um zu vollständigen Formeln der semantischen Repräsentationssprache, genannt Konstruktsprache (KS), zu gelangen.[3] Diese SRS ist eine logikorientierte Sprache mit Lambda-Abstraktion. Sie läßt sich auf Prädikatenlogik reduzieren, wobei Axiome spezielle Bedeutungen festlegen. Jedes Prädikat hat bestimmte Sorten als Argumente. Die Sorten stellen eine Sicht auf den Anwendungsbereich und damit das begriffliche Wissen dar.

Die Auswertung einer KS-Formel bildet sie direkt auf eine Datenbasis-Anfrage ab. Das Datenbanksystem liefert Tabellen, die von der Antwortaufbereitung formatiert werden. Eine anspruchsvollere Aufgabe hat dies Komponente, wenn einfach `ja` oder `nein` geantwortet wird. Schließlich gibt es Fehlermeldungen, die vom System erzeugt werden.

Ein Beispiel vermittelt einen Eindruck von der Verarbeitung.

Welche Grenzwerte wurden in der Probe vom 5.4.76 bei Lauxmann überschritten?

Ergebnis Pass 1:




Ergebnis Pass 2:

```
(lX2
(grenzwu(EIN(lX9.dimzahl
 (anteil X2
       (EIN(lX6.stoffkoll
         (probe (EIN(lX5.betrieb (betrieb lauxmann (EIN(ort)) X5.betrieb)))
                (EIN(lX3.int (intemp 760405 X3.int)))
          X6.stoffkoll)))
 (EIN(methode))
 (EIN(abstrobj))
 X9.dimzahl)))
X1.sit)))
```

Die Variablen sind immer Variablen einer bestimmten, nach einem Punkt angegebenen Sorte. Jedes KS-Prädikat (`grenzwu, anteil, probe`) hat bestimmte geforderte Argumente. So hat z.B. die Grenzwertuntersuchung das Argument `anteil` und ist durch eine Variable der Sorte `sit` (Situation) zu kennzeichnen. Der Anteil eines chemischen Stoffes an einer Probe erfordert ein Argument von der Sorte `stoffkoll`, eine `methode` und ein `abstrobj` und ist durch eine Variable der Sorte `dimzahl` zu kennzeichnen. Eine Probe ist durch einen Betrieb und einen Zeitpunkt definiert und ist durch eine Variable der Sorte `stoffkoll` zu kennzeichnen. Ein Betrieb wird durch einen Namen und einen Ort definiert und ist mit einer Variable der Sorte `betrieb` zu kennzeichnen.

[3]Gisela Zifonun "Die Konstruktsprache als Semantiksprache für einen Ausschnitt des Deutschen", Mitteilungen des Instituts für deutsche Sprache 5, S. 35 - 46, 1979

```
Ergebnis Pass 3:

 CR1 (3+)
 NI  1
```

Das Ergebnis sind zwei chemische Stoffe, für die eine Zahl angibt, um wieviel Einheiten der zulässige Grenzwert überschritten wurde.

### 6.1.1 Präsuppositionsbehandlung

Für PLIDIS wurde eine Komponente zur Erkennung von Präsuppositionsverletzungen entwickelt.[4] Sie korrigiert die falsche präsupponierte Annahme, damit der Benutzer zu einer brauchbaren Neufassung seiner Frage kommen kann, damit der Benutzer Informationen erhält, die er offensichtlich nicht kennt, aber benötigt, und damit die Informationsausgabe seinen Intentionen entspricht. Präsuppositionsverletzungen sind in PLIDIS auf Nichtexistenz angenommener Objekte oder Sachverhalte beschränkt. Die Existenz bezieht sich stets auf eine Welt. Dabei gibt es die Abstufungen:

Prinzipielle Präsuppositionen bedeuten, daß es eine mögliche Welt gibt, in der das betreffende Objekt (der Sachverhalt) existiert. Eine prinzipielle Präsuppositionsverletzung ist z.B. der rote Schimmel oder andere Widersprüche.

Aktuelle Präsuppositionen beziehen sich auf eine bestimmte Welt. Eine aktuelle Präsuppositionsverletzung ist z.B. in dem PLIDIS-Weltausschnitt das Fehlen einer bestimmten Abwasserprobe -- es könnte sie geben, aber es gibt sie (gerade) nicht.

Der in PLIDIS modellierte Weltausschnitt ist korrekt, insofern er nichts enthält, was in der realen Welt nicht vorkommt, aber unvollständig, insofern er viele reale Sachverhalte nicht enthält. Bezogen auf den Weltausschnitt der Abwasserüberwachung sind Schulen so wenig möglich wie rote Schimmel in der Gesamtwelt. Eine prinzipelle Präsuppositionsverletzung kann also dadurch entstehen, daß der Benutzer über den Rahmen des modellierten Weltausschnitts hinausgeht. Dies wird bereits in Pass2 abgefangen, da dort durch die Sortenüberprüfung festgestellt wird, ob ein Objekt einer Sorte zugeordnet werden kann. Gelingt die Zuordnung nicht, so liegt eine prinzipielle Präsuppositionsverletzung vor und das System gibt eine Fehlermeldung aus. Das Gros der zu behandelnden Präsuppositionen sind mögliche aber nicht vorhandene Sachverhalte oder Objekte.

Die sprachlichen Formen, die etwas präsupponieren, sind bei PLIDIS: indefinite/definite Kennzeichnungen, Quantifikationen und die Verben `feststellen` und `enthalten`. Letztere präsupponieren, daß eine Probe auf jenen Stoff hin untersucht wurden, nach dem gefragt ist. KS repräsentiert bereits explizit die durch Wörter wie Grenzwert, Probe implizierten Entitäten.

Eine Präsuppositionsverletzung kann nur vorliegen, wenn sich eine leere Extension bei einer Datenbankanfrage ergibt. Es muß dann sichergestellt werden, daß alle Teilausdrücke der Anfrage eine nichtleere Extension hatten. Die geschachtelte KS-Formel wird zerlegt. Auf jeder Ebene wird festgestellt, ob ein leere Extension vorliegt. Wenn ja, werden die Teilausdrücke geprüft. Es wird also die Ebene darunter untersucht. Wenn nein, so wird auf derselben Ebene fortgefahren. Die leere Extension eines Ausdrucks `e` mit Teilausdrücken, die alle nichtleere Extensionen haben, bedeutet eine Negation von `e`.

Wenn z.B. für die KS-Formel des obigen Beispiels keine Grenzwertüberschreitung festgestellt wurde, wird geprüft, ob der `anteil`-Ausdruck eine nichtleere Extension liefert. Ist dies der Fall, so ist die negative Antwort (keine Grenzwertüberschreitung) richtig. Die Teilausdrücke von `anteil` müssen nicht untersucht werden. Ist die Extension des `anteil`-Ausdrucks aber leer, so werden dessen Teilausdrücke untersucht. Wird nun festgestellt, daß `X6` eine leere Extension hat, so werden die Teilausdrücke von `probe` untersucht. Gibt es zwar die Firma Lauxmann und natürlich auch das Datum, so haben wir als Grund für die leere Extension der gefragten Variable `X2` festgestellt:

```
Es gibt keine Probe bei Lauxmann am 5.4.76.
```

---

[4] Katharina Morik "Antwortverhalten eines Frage-Antwort-Systems bei Verletzung von Präsuppositionen - bezogen auf das in der Datenbasis des Systems PLIDIS abgebildete Weltmodell", Institut für deutsche Sprache, 1978.

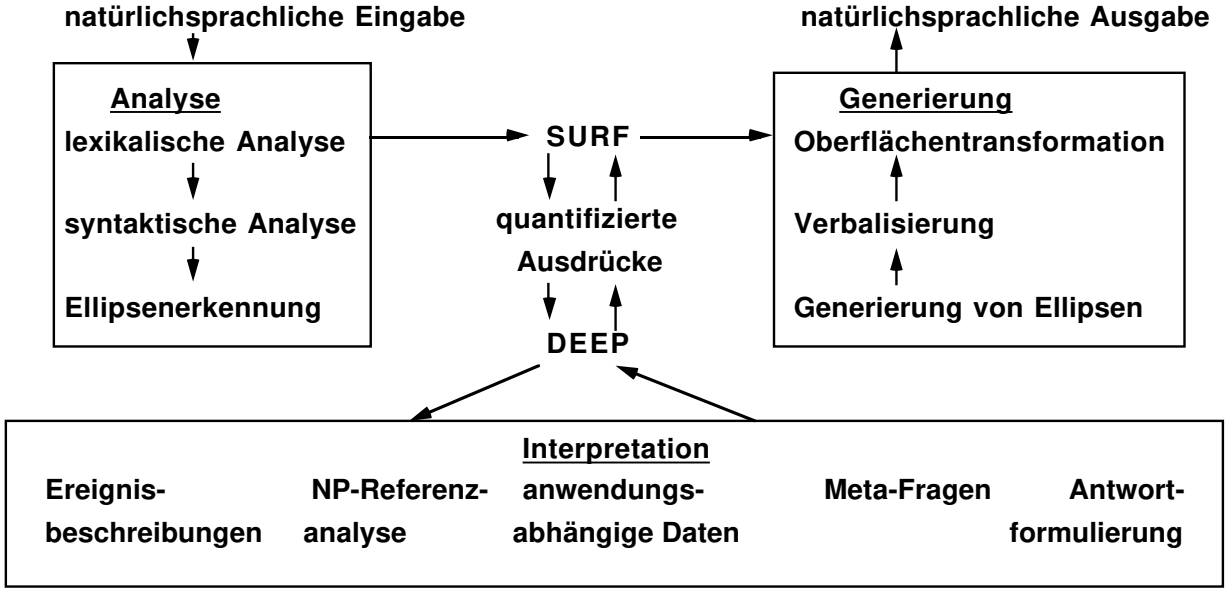
Diese Meldung wird ausgegeben. Sie vermeidet das Mißverständnis, das ohne Präsuppositionsprüfung aufgetreten wäre. Dann nämlich hätte das System geantwortet:

```
Nein.
```

und der Benutzer hätte die Firma Lauxmann für vorbildlich gehalten.

### 6.2 HAM-ANS

Das System HAM-ANS (HAMburger Anwendungsorientiertes Natürlichsprachliches System) ist ein Zugangssystem zu unterschiedlichen Performanzsystemen.[5] Dabei wurde darauf Wert gelegt, daß das System über kommunikative Fähigkeiten verfügt und daß es eine ebenso sorgfältige Generierungs- wie Analysekomponente enthält. Exemplarisch wurde der Zugang zu drei Systemtypen realisiert: zu einem Datenbanksystem, zu einem bildverarbeitenden System und zu einem Hotelbuchungssystem. Damit wurde gezeigt, daß für die Anwendungsunabhängigkeit keineswegs auf ein wissensbasiertes Vorgehen verzichtet werden muß. HAM-ANS wurde in der LISP-Erweiterung FUZZY auf der PDP10 unter TOPS10 implementiert.




Die Architektur zeigt deutlich, daß hier Generierung und Analyse gleich wichtig genommen wurden. Es wurde aber nicht derselbe Prozeß für die verschiedenen Verarbeitungsrichtungen eingesetzt.

Die Grammatik für das Parsing ist im ATN-Formalismus implementiert. Das Ergebnis des Parsings ist ein Ausdruck der Sprache SURF (semantische Repräsentationssprache )), die noch recht nah an der sprachlichen Oberfläche ist. Die SURF-Ausdrücke vorangegangener Sätze sind gespeichert, um so eine Ellipsenrekonstruktion für Ersetzungsellipsen zu ermöglichen. Die Quantoren werden in einem Normalisierungsprozeß so verarbeitet, daß ihr Skopus eindeutig ist. Das Ergebnis der Behandlung ist ein Ausdruck der semantischen Repräsentationssprache DEEP. Dieser Ausdruck wird über dem referentiellen Wissen des Sachbereichs ausgewertet. Das Ergebnis wird in einen SURF-Ausdruck überführt, der durch die Ellipsengenerierung verkürzt werden kann. Die Verbalisierung überführt einen SURF-Ausdruck in eine syntaktische Strukturbeschreibung. Die Verbalisierungskomponente findet für Referenzobjekte eindeutige

[5] Wolfgang Hoeppner, Thomas Christaller, Heinz Marburger, Katharina Morik, Mike O'Leary, Wolfgang Wahlster "Beyond Domain Independence: Experience With the Development of a German Language Access System to Highly Diverse Background Systems" in Procs. IJCAI-83, 1983, S. 588 - 594
Stephan Busemann, Wolfgang Hoeppner, Heinz Marburger, Katharina Morik "Representing and Processing Copula und Full-Verb Sentences in HAM-ANS", in Stoyan (ed) 9th GWAI, Berlin, 1985
Wolfgang Hoeppner, Katharina Morik, Heinz Marburger "Talking it Over: The Natural Language Dialog System HAM-ANS" in Bolc, Jarke (eds) Cooperative Interfaces to Information Systems, 1986, S. 189 - 258

Kennzeichnungen. Die hierarchische Struktur wird in eine lineare Kette von Wortformen und Merkmalen überführt. Die Oberflächentransformation macht aus dieser Kette einen natürlichsprachlichen Satz.

Für bestimmte metakommunikativen Äußerungen sind eigene Auswertungs- und Beantwortungsstrategien implementiert:

Tatsächlich? Begründungsanforderung löst Verbalisierung von Inferenzen aus.

Schön. löst in der Hotelszene die Übernahme der Initiative durch das System aus.

Was habe ich eben gefragt? die Frage wird anhand des Dialoggedächtnisses beantwortet.

Was verstehen Sie unter Meßart? die Frage wird anhand des begrifflichen Wissens beantwortet.

Hier wird also kurzerhand eine bestimmte Äußerung als Indikator für einen Sprechakt verarbeitet.

HAM-ANS verfügt über ein Partnermodell bei der Hotelanwendung.[6] Stereotype Zimmerwünsche für bestimmte Hotelkundentypen sind vorgegeben und werden dafür ausgenutzt, einem Anrufer ein bestimmtes Zimmer zuzuordnen. Ist kein Zimmer der vermutlich gewünschten Kategorie mehr frei, wird verbalisiert, welche Merkmale das noch verfügbare Zimmer hat. So wird z.B. die Eigenschaft, daß das Zimmer einen Tisch zum Schreiben hat, bei Akademikern für so wichtig gehalten, daß das System verbalisiert, wenn keines der freien Zimmer einen solchen Tisch hat. Oder bei Anrufern, die für jemand anderen ein Zimmer buchen und einen Firmennamen angeben, wird Fernseher und Zimmerbar sowie Sitzecke für wichtig gehalten.

Ein Beispieldialog aus der Verkehrsanwendung wird hier nicht mit allen internen Zwischenergebnissen widergegeben.

Wieviele Fahrzeuge waren zu sehen?

21

Wieviel Personenwagen sind gefahren?

Drei von den sechzehn auf der Schlüterstraße, und zwei von den sechzehn auf der Hartungstraße.

Bog ein Personenwagen von der Schlüterstraße ab?

Ja, einer in die Hartungstraße.

Fuhr danach ein Kombi oder ein LKW?

Ja, zwei Kombis auf der Schlüterstraße, und ein Kombi auf der Hartungstraße, und ein Lastkraftwagen fuhr danach auf der Hartungstraße.

Befinden sich auch Autos auf einem Bürgersteig?

Ja, sechs Autos sind direkt auf einem Bürgersteig.

Auf welchem Bürgersteig?

Zwei Autos sind direkt auf dem linken und vorderen Bürgersteig, und vier Autos sind direkt auf dem rechten und vorderen Bürgersteig.

OK

Ende.

[6]Katharina Morik " User Models and Conversational Settings: Modeling the User's Wants", in Kobsa, Wahlster (eds) User Models in Dialog Systems, Berlin, 1989, S.364 - 385

Schon dieser kurze Beispieldialog aus einer Anwendung zeigt, wie wichtig die Elliptifizierunge bzw. Ellipsenerkennung ist. Man stelle sich vor, man müßte alle Nachfragen als Benutzer ausformulieren! Die Analyse und Generierung von Ellipsen sind einander einbettende Prozesse. Eine *anticipation feedback loop* ruft die Analyse einer Ellipse auf für jede generierte Ellipse. Kann HAM-ANS selbst die Ellipse nicht rekonstruieren, wird sie auch nicht ausgegeben. Insofern wirkt sich jede Veränderung der Ellipsenrekonstruktion direkt auf die Elliptifizierungskomponente aus. [7]

Auch zeigt schon dies Beispiel die Wichtigkeit der Perspektivenübernahme und die sprachliche Realisierung räumlicher Verhältnisse. Voraussetzung für diese Fähigkeiten sind die Repräsentation begrifflichen und viesuellen Wissens.

### 6.2.1 Wissensquellen

Das System hat für jeden Anwendungsbereich das passende begriffliche, referentielle und visuelle Wissen. Das begriffliche Wissen wurde noch nicht in einem Termsubsumtionsformalismus dargestellt, sondern in einem semantischen Netz mit sechs Kantentypen:

U-Kante: ordnet Objektknoten in einer Unterbegriffshierarchie an und ist Basis für die Vererbung von Information auf untergeordnete Knoten; `((U SITZGELEGENHEIT STUHL).1.0)`

E-Kante: stellt die Bezeichnung zwischen einem Objektknoten und einem Eigenschaftsknoten her; `((E DING FARBE).1.0)`

V-Kante: ordnet einem Eigenschaftsknoten einen Wert-Knoten zu; `((V FARBE BLAU).1.0)`

D-Kante: ordnet Objektknoten inhärente Eigenschaften zu und entspricht damit der REF-Kante im referentiellen Wissen;

T-Kante: verbindet Objektknoten, die in einer Teil-von-Relation stehen, wobei der Fuzzy-Wert den notwendigkeitsgrad der Assertion repräsentiert; `((T STUHL BEIN)1.0)`

Anzahl-Kante: gibt die Anzahl von Teilen eines Objektknotens mit einem Notwendigkeitsgrad an; `((ANZAHL (T STUHL BEIN) 4).0.6)`

Es gibt vier Knotentypen:

Objektknoten: bezeichnen Klassen von Objekten oder Begriffe;

Dimensionsknoten: entsprechen linguistischen Variablen der Fuzzy-Logik und repräsentieren Eigenschaften ;

Dimensionswert-Knoten: müssen mit einem Dimensionsknoten verbunden sein und geben die Eigenschaftsausprägungen von dessen Eigenschaft an;

Wert-Knoten: Zahlen oder linguistische Hecken (z.B. viele, relativ wenige).

Mit diesem Formalismus wurde der Weltausschnitt der Hotelzimmer, der Weltausschnitt von Schiffen und ihren Fahrten sowie der Weltausschnitt des Straßenverkehrs modelliert. Das referentielle Wissen stellt dann mit Bezug auf das begriffliche Wissen die Information dar über ein bestimmtes Hotel und seine Zimmer, eine bestimmte Menge von Schiffahrten (gespeichert in einer relationalen Datenbank) sowie eine Szene bei einer bestimmten Straßenkreuzung mit bestimmten Bewegungen von Fahrzeugen (Ergebnis der Bildverarbeitung einer Szene). Einem bestimmten, eindeutig bezeichnetem Objekt werden über REF-Kanten seine Eigenschaften zugeordent.

Visuelle Daten in Form von Objekten in einem 2D-Koordinatensystem gab es nur über die Straßenkreuzung und die Hotelzimmer. Wichtig ist dabei die Repräsentation von Blickpunkten, für die

[7]Wolfgang Wahlster, Heinz Marburger, Anthony Jameson, Stephan Busemann "Over-Answering Yes-No Questions: Extended Responses in a NL Interface to a Vision System" in Procs. IJCAI-83, 1983, S.643 - 646

spezifiziert ist, welche Objekte mit welchem Auffälligkeitswert von ihnen aus gesehen werden können. Für einen Blickpunkt ist ein bestimmtes Objekt mehr oder weniger auffällig: `((AUFFAELLIG STUHL1).0.8)` Bei der Verkehrswelt sind außerdem Zeitintervalle für bestimmte Objekte angegeben: innerhalb dieses Intervalls sind sie sichtbar: `(EXIST AUTO20 (1. 20)).1.0)`

### 6.2.2 Nominalphrasenreferenz

Die Behandlung von Referenzobjekten wird im Rahmen der Verarbeitung von NPs durchgeführt. Bei der Analyseseite muß festgestellt werden, auf welches Referenzobjekt sich eine NP aus der Eingabe beziehen soll. Wird ein neues Referenzobjekt eingeführt oder auf ein bereits eingeführtes Bezug genommen? Bei der Generierungsseite muß eine Kennzeichnung ausgewählt werden, die es dem Benutzer ermöglicht, das Referenzobjekt zu ermitteln. Auch hierfür ist wichtig zu wissen, was im Dialog über ein Referenzobjekt bereits wechselseitig bekannt ist. Wenn beispielsweise nur das System (per Kamera) sieht, daß das Auto, das in die Hartungstraße abbog, gelb ist, dann kann es zur Kennzeichnung genau dieses Autos nicht die Farbe wählen.

HAM-ANS verfügt über ein Referenzgedächtnis, in dem die eindeutige, systeminterne Bezeichung eines Objekts verknüpft wird mit

- dem DEEP-Ausdruck für die kennzeichnende NP, die zur Kennzeichnung des Objekts verwendet wurde,
- dem Genus des verwendeten Substantivs,
- der Häufigkeit, mit der das Objekt bereits genannt wurde (in welcher sprachlichen Form auch immer),
- dem Zeitpunkt der Erwähnung (Satzzähler).

Pronomen werden so aufgelöst, daß dem zuletzt erwähnten Objekt, das mit einem Substantiv verbalisiert wurde, dessen Genus zum Pronomen passt, als Referenzobjekt betrachtet wird.

| Dialog | Objekt | DEEP-Ausdruck | Genus | Häufigkeit | Satzzähler |
|---|---|---|---|---|---|
| **Bog der gelbe PKW ab?** | **PKW3** | **(t-q: (for: (q-d: D- (1 1)) X4) (f-o: AND ...))** | **m** | **7.5** | **5** |
| **Ja, in die Hartungstraße** | **Hartung-straße** | **(t-q: ... ) )** | **f** | **5.5** | **5.5** |
| **Hielt er danach an?** | **PKW3** | **(t-q: (for: (q-d: D- (1 1)) X7) (af-a: proform X7 ER))** | **m** | **9.5** | **6** |
| **Nein.** | | | | | |

Wenn die Auswertung einer Anfrage als Ergebnis lediglich eine eindeutige Objektbezeichnis zurücklieferte, so wird zunächst aus begrifflichem und referentiellem Wissen alles herausgesucht, was sich auf dieses Objekt bezieht. Es wird dann die Eindeutigkeit dieser Kennzeichnung anhand des Referenzgedächtnisses geprüft .

Bei der Verbalisierung eines Nomens wird geprüft, ob es elidierbar ist. Wenn ein Nomen bereits in der letzten Frage vorkam wird es nicht verbalisiert, es sei denn der ermittelte Quantor für das Ergebnis der Auswertung sei `D- (1 1)` und es gibt keine Attribute zu seiner Kennzeichnung. Im letzteren Fall wird ein Pronomen generiert. Wenn ein Objekt das erste Mal erwähnt wird, dann mit dem unbestimmten Artikel, beim nächsten Mal mit dem bestimmten, dann mit einem Pronomen (wenn es nicht elliptifiziert werden kann).

Das System HAM-ANS integriert eine Fülle von pragmatischen Phänomenen: Anaphora, Personalpronomina, Präsuppositionen, Metakommunikation, wechselseitiges Wissen und seine Ausnutzung zur Kennzeichnung. Dabei wird jedes Phänomen nicht erschöpfend operationalisiert. Vielmehr wurde darauf Wert gelegt, das Zusammenwirken verschiedener Komponenten in einem NLS zu modellieren.

## 7. Und nun?

In dieser Vorlesung wurde das Spektrum an Phänomenen aufgezeigt, das NLS behandeln können. Es wurde auf viele Probleme hingewiesen, ohne daß deren Behandlung aufgezeigt wurde. Hier muß auf die Literatur verwiesen werden, die sich durch die Schlagwörter und die hier angegebene Literatur erschließen läßt. Daß die Künstliche Intelligenz einen Beitrag zur Erforschung natürlicher Sprache liefert, wird daraus deutlich, daß in linguistischen Fachtexten auf die Forschung zu NLS verwiesen wird. Gerade die operationale Beschreibung ermöglicht es, festzustellen, was man ansonsten leicht vergessen hätte. So produzierte z.B. das Vorläufersystem zu HAM-ANS, HAM-RPM, auf die Frage

```
Wo steht der Tisch?
```

gemäß der üblichen linguistischen Regeln die zutreffende Antwort

```
Unter dem Whiskeyglas.
```

Hier wird deutlich, daß die Auswahl des Referenzobjektes als Bezugspunkt für eine Ortsangabe och zusätzlichen Beschränkungen unterliegt, die vorher übersehen wurden, weil man die Regeln nicht austesten konnte. Ein anderer Beitrag zur Linguistik ist das Zusammenwirken verschiedener Vorschläge, das nur in einem lauffähigen System möglich ist. Lexikon, Syntax, Semantik und Pragmatik können im Zusammenhang studiert werden. Dabei werden dann die verschiedenen Anforderungen z.B. an das begriffliche Wissen aus den verschiedenen Komponenten deutlich.

Der Bezug von begrifflichem Wissen und Weltwissen muß für ein NLS modelliert werden. Das Problem ist nur, daß nicht alles, was wir glauben und wissen, in einem System repräsentiert werden kann. Hier muß ein anderer Ausweg gesucht werden, als die bisherigen Alternativen:

- Für den Sachbereich, für dessen Einsatz das NLS gedacht ist, wird alles begriffliche und referentielle Wissen modelliert, das zur Behandlung natürlichsprachlicher Dialoge/Texte notwendig ist.
- NLS lassen semantische und pragmatische Aspekte außer Acht.

Vielmehr muß zum einen der Aufbau von Wissensbasen durch das Verstehen von Äußerungen untersucht werden. Zum anderen ist ein direkter Zugriff auf die Welt selbst und nicht nur auf eine Repäsentation der Welt notwendig. Die beste Repräsentation der Welt ist die Welt selbst! Natürlich verfügt ein Rechner stets nur über Repräsentationen. Mit "Zugriff auf die Welt selbst" meine ich also Prozesse, die bei Bedarf Repräsentationen von relevanten Teilen der Welt herstellen. Zum Beispiel kann ein bildverarbeitenden System eine Repräsentation einer Szene aufbauen. Der direkte Zugriff betrifft aber nicht nur die Wahrnehmungsseite. Auch Handlungen sind wichtig, um das Sprachverstehen noch besser testen zu können. Die Kopplung von NLS und Robotern bietet die Möglichkeit, einen Zugriff auf die Welt für ein NLS herzustellen. Aber auch die Robotik verlangt meistens, daß alles notwendige Wissen vorgegeben ist. Für den flexibleren Aufbau von Wissen anhand der realen Welt und natürlichsprachlicher Äußerungen ist die Einbeziehung maschinellen Lernens unverzichtbar.

*Lernen ist ein Prozeß, bei dem ein System eine abrufbare Repräsentation von vergangenen Interaktionen mit seiner Umwelt aufbaut.* Paul D. Scott "Leanrning - The Construction of A Posteriori Knowledge Structures", Proc. AAAI 1983, Washington.

## 8. Literatur


Noam **Chomsky** (1965): Aspects of the Theory of Syntax, dtsch. 1972 "Aspekte der Syntax-Theorie", Suhrkamp, Frankfurt a.M.

Noam **Chomsky** (1975): Reflections on Language, dtsch. 1977 "Reflexionen über die Sprache", Suhrkamp, Frankfurt a.M.

Michael A. **Covington (1994):** Natural Language Processing for Prolog Programmers, Englewood Cliffs, Prentice Hall.

Günther **Görz (Hg.) (1995 2):** Einführung in die Künstliche Intelligenz, Addison-Wesley, Bonn.

Helen **Leuninger** (1989): Neurolinguistik - Probleme, Paradigmen, Perspektiven, Westdeutscher Verlag, Opladen.

Stephen C. **Levinson (1983):** Pragmatics, dtsch. 1990 "Pragmatik", Niemeyer, Tübingen.

Michael **Paetau** (1990): Mensch-Maschine-Kommunikation - Software, Gestaltungspotentiale, Sozialverträglichkeit, Campus, Frankfurt a.M.

Dieter **Wunderlich** (1974): Grundlagen der Linguistik, Rowohlt, Reinbek.

**Index**

Ambiguitäten 6

Anaphora 50; 51; 58

Bedeutung 27; 30

begriffliche Wissen 52; 56; 57

beobachtungsadäquat 8

beschreibungsadäquat 8

Buchstabenbaum 26

DCG 15; 17; 23

definite clause grammar 15

Deixis 39

Ellipsen 55; 56

erklärungsadäquat 8

Ersetzungsprobe 12

extensionale Bedeutung 27

Gelingensbedingungen 45

illokutiver Akt 45; 47

immediate dominance 13

intensionale Ausdrücke 28

intensionale Bedeutung 27; 30

Kohärenz 50

Kommunikation 6; 41

Kommunikationsmodell 7

Kompetenz 5

Kompositionalität 28

Konversationsimplikationen 41

Konversationsimplikatur 41

Kooperationsprinzip 41

Lambda 29; 52

language 5

langue 5

Lexem 21; 22

Lexikon 20

Lexikonkomponente 22

linear precedence 13

lokutiver Akt 45

ltree 26

Merkmalsstruktur 18; 19; 25

Modalitätsmaxime 42

Morphem 21

Morphologische Analyse 21; 23

Natürlichsprachliche Systeme 3; 51

parole 5

Parser 14

Performanz 5

perlokutiver Akt 45

Personalpronomina 39; 58

Pragmatik 5

Präsupposition 53; 54

Präsuppositionen 40

Pronomen 58

Qualitätsmaxime 42

Quantitätsmaxime 42

Quantor 34; 35

Referent 27

Referenzobjekt 57

Relationsmaxime 42

Resolution 10

Restriktion 35

Semantik 5; 30

semantische Repräsentationssprache 28; 30; 52; 55

Sinn 27; 30

Skopus 35

Sprechakt 55

Strukturbeschreibung 14; 16

Substitution 28

Syntax 5

top-down Strategie 14; 23; 25

Unifikation 10

unifikationsbasierte Grammatik 18

Unifikator 10

Verbrahmen 24; 25; 31; 34

Verschiebeprobe 12

Vollformenlexikon 21

Zeichensysteme 4