6 Natürlichsprachliche Systeme

Die menschliche Sprachfähigkeit ist eine der verblüffendsten Intelligenzleistungen des Menschen. Verblüffend deshalb, weil sie nachgewiesenermaßen mit äußerst komplexen Strukturen zu tun hat (kontextsensitiven Sprachen) und doch von jedem Kleinkind, sogar einem debilen Kind, erlernt wird. Verblüffend auch, weil Menschen einander verstehen können, obwohl die Sprache mehrdeutig ist. Menschen bilden mehr oder weniger wohlgeformte Texte und erkennen, welche Texte wohlgeformt sind und welche nicht (Syntax). Menschen machen mit sprachlichen Mitteln wahre oder falsche Aussagen. Ein Text kann wie eine Formelmenge widersprüchlich, unzutreffend oder wahr sein (Semantik). Menschen handeln mithilfe von Texten (Pragmatik), und zwar in zweierlei Hinsicht. Erstens veranlassen sie eine Handlung oder koordinieren eine gemeinschaftliche Handlung, indem sie miteinander reden. Wenn ich im Restaurant dem Kellner sage, welches Gericht ich gern essen möchte, werde ich es höchstwahrscheinlich bekommen. Der Kellner wird nicht darüber sinnieren, ob mit "Gericht" eine juristische Institution gemeint ist. Er wird meine Frage, ob ich ein vegetarisches Essen bekommen kann, nicht einfach mit "ja" beantworten, sondern es auch bringen. Sollte ich aus Versehen einen unvollständigen Satz geäußert haben, wird auch das ihn nicht stören. Das Reden wirkt sich aus, es veranlaßt andere zum Handeln. Zweitens ist schon das Reden selbst eine Handlung: ein Wunsch, eine Bestellung, eine Aufforderung, ein Befehl, eine Drohung, eine Warnung, ein Vorwurf, eine Anschuldigung, eine Verurteilung, ein Dank, ein Kompliment, ein Trost, ... Einige Handlungen sind nur sprachlich auszuführen wie etwa der Eid und die Taufe.

Das Verblüffen über die menschliche Sprachfähigkeit ist vielleicht in der Kluft zwischen Fertigkeit und Wissen begründet: wir alle haben die Fertigkeit, mit einer (oder gar mehreren) natürlichen Sprache umzugehen und kein Mensch bisher hat ausreichendes Wissen über diese Fertigkeit, um sie vollständig zu modellieren.

6.1 Einführung

Auch innerhalb der sprachverarbeitenden KI gibt es eine kognitive, eine theoretische und eine anwendungsorientierte Ausrichtung. Ihre konkrete Ausprägung wird kurz vorgestellt, bevor ein idealisiertes sprachverarbeitendes System skizziert wird, das die einzelnen Komponenten und ihre Wissensquellen vorstellt, die dann in den weiteren Abschnitten besprochen werden.

6.1.1 Kognitionsorientierte Arbeiten

Die kognitive Ausrichtung sprachverarbeitender KI versucht, sprachliche Prozesse beim Menschen zu modellieren. Das System ist eine Simulation kognitiver Vorgänge beim Menschen. Wie bilden wir einen Satz? Wie verstehen wir einen Text? Das beginnt mit der Wahrnehmung von Lauten und ihrer Zuordnung zu Zeichen und setzt sich bis zum Verstehen von Texten und ihrer Zuordnung zu Handlungen fort. Bei der kognitiven Ausrichtung ist die sprachverarbeitende KI auf psychologische oder psycholinguistische Arbeiten angewiesen. Die KI operationalisiert ein psycholinguistisches Modell, so daß mit der Simulation Experimente durchgeführt werden können. Ein Modell zur Satzgenerierung wird im folgenden vorgestellt. Natürlich gibt es noch viele andere Phänomene, die psycholinguistisch untersucht wurden, zum Beispiel das Geschichten-Verstehen und -Wiedergeben.

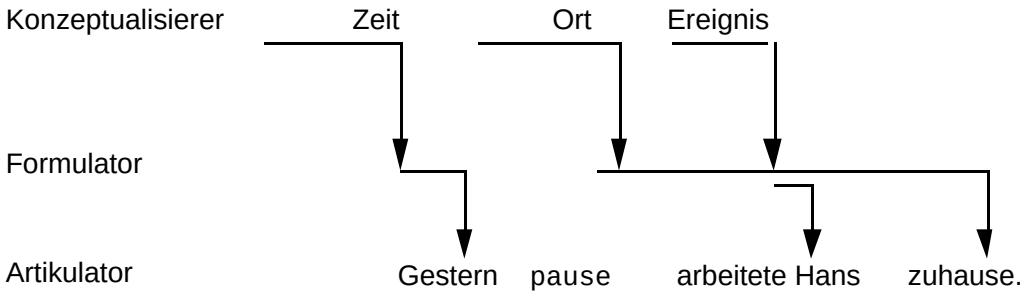
Levelt (1989) gibt ein Prozeßmodell eines Sprechers an, das aus drei Komponenten besteht:

- Der **Konzeptualisierer** wählt aus, was gesagt werden soll. Dabei wählt die Makroplanung gemäß einer Intention des Sprechers einen Plan für die Gesamtäußerung aus und ordnet die einzelnen Planschritte linear an. Die Mikroplanung bestimmt die Perspektive und die Betonungen für die einzelnen Teiläußerungen. Das Ergebnis des Konzeptualisierers enthält alle Informationen, die man zur Erzeugung natürlichsprachlicher Äußerungen braucht. Obendrein beinhaltet der Konzeptualisierer eine Selbstbeobachtungsinstanz, die bereits geäußerte (Teil-)Sätze oder Satzpläne auf Wohlgeformtheit und Verständlichkeit hin überprüft.

- Der **Formulator** übersetzt das Ergebnis des Konzeptualisierers in grammatische Strukturbeschreibungen und phonologische Strukturen. Dabei wird ein Lexikon verwendet, das semantische, syntaktische und phonologische Information enthält. Die semantische Information im Lexikon kann direkt mit der semantischen Information aus den Satzplänen abgeglichen werden. Die mit demselben Wort verbundene syntaktische Information schränkt dann weitere Wahlmöglichkeiten ein. Das Ergebnis ist ein Artikulationsplan.

- Der **Artikulator** schließlich realisiert den Artikulationsplan mithilfe der Atmung und der Sprechwerkzeuge (Zunge, Gaumen/Zähne/Rachen, Lippen).

Gesprochene Sprache wird mit einer Rate von zwei bis drei Wörtern pro Sekunde, etwa 15 Phoneme pro Sekunde geäußert (Busemann, Novak 1992 in Görz (ed) 1992). Wenn der Artikulator warten würde bis Konzeptualisierer und Formulator alles vorausgeplant haben, würde es im Redefluß lange Pausen geben. Dies ist nicht der Fall. Deshalb nimmt man ein **Kaskadenmodell** an: jede Komponente schickt alle Teilergebnisse fortlaufend auf eine Ausgabekaskade, die überläuft, so daß eine Eingabe in die nächste Kaskade entsteht. Es ist unklar, wie schnell eine Ausgabekaskade überläuft, wie groß die Teilergebnisse sind, die schon weitergegeben werden. Immerhin kann das Kaskadenmodell erklären, wie sowohl parallel als auch seriell verarbeitet werden kann. Das folgende Bild ist Busemann, Novak (1992) entnommen und illustriert, wie der Formulator aufgrund syntaktischer Beschränkungen die Reihenfolge, die vom Konzeptualisierer vorgegeben wurde, umstellt. Während "Gestern auf dem Bahnsteig traf ich Ruth." ein wohlgeformter deutscher Satz mit der Abfolge Zeit, Ort, Ereignis ist, ist "Gestern zuhause arbeitete Hans." nicht syntaktisch wohlgeformt.



Die Pause im Redefluß erklärt sich durch die Umstellung der vom Konzeptualisierer vorgegeben Abfolge. DeSmedt (1990) hat eine Simulation der Satzgenerierung implementiert.

6.1.2 Der Beschreibungsansatz

Die theoretische Ausrichtung der sprachverarbeitenden KI will die Sprachfähigkeit operational beschreiben. Das System, das die operationale Beschreibung darstellt, muß nicht genauso funktionieren wie ein Mensch. Unter kognitiven Gesichtspunkten kann es also ruhig falsch sein. Es muß aber Syntax, Semantik, Pragmatik sprachlicher Phänomene beschreiben oder sogar erklären, also beschreibungsadäquat oder sogar erklärungsadäquat sein. Dabei ist die sprachverarbeitende KI auf linguistische und sprachphilosophische Arbeiten angewiesen. Dies Verhältnis der KI zur Linguistik unterscheidet sich von ihrem Verhältnis zur Psychologie. Während die psychologischen Untersuchungen, die von Teilbereichen der KI herangezogen werden, die Rolle der Materialsammlung spielen, aufgrund derer eine Hypothese über menschliches Verhalten zurückgewiesen oder bestärkt werden kann, sind die linguistischen und sprachphilosophischen Arbeiten, die bei der Entwicklung natürlichsprachlicher Systeme Verwendung finden, bereits konstruktive Theorien, die sich desselben formalen Apparates[33] bedienen und ihn ebenso weiterentwickeln wie es die Informatik tut. Nur ein Schlaglicht: Jeder Informatik-Student kennt die Chomsky-Hierarchie für Sprachen und Noam Chomsky ist ein Linguist. Die sprachverarbeitende KI muß also eine systematische, formale Theoriebildung einer anderen Disziplin berücksichtigen. Die Darstellung linguistischer Theorien erfordert eine Reihe von Spezialvorlesungen und Seminaren und kann nicht nebenher in einer Einführung in die KI geschehen. Hier kann nur darauf hingewiesen werden, daß äußerste Vorsicht bei Urteilen über die natürliche Sprache geboten ist. Insbesondere warne ich vor dem Fehlurteil, natürliche Sprachen seien mit simplen Techniken, wie sie für künstliche Sprachen ausreichen mögen, in den Griff zu bekommen. Wenn auch ein Parser für natürliche Sprachen an einen Compiler für Programmiersprachen zu Recht erinnert, so darf die Entwicklung einer Syntaxtheorie für natürliche Sprachen keinesfalls auf die Entwicklung von Programmiersprachen reduziert werden: nicht nur ist allein schon ihre Syntax komplexer als die einer Programmiersprache - sie ist obendrein eingebettet in die komplexe Semantik und Pragmatik natürlicher Sprachen.

Das Programm der Grammatiktheorie (**Syntax**) ist gemäß Chomsky so angegeben:

1) eine rekursive Aufzählung aller potentiellen Grammatiken

2) eine Aufzählung aller potentiellen Sätze einer Grammtik

3) die konstruktive Definition einer Strukturbeschreibung und die Angabe einer Funktion, die einem beliebigen Satz gemäß einer Grammatik eine Strukturbeschreibung zuordnet.

Die Semantiktheorie muß den Strukturbeschreibungen oder deren Umformungen Sinn und Bedeutung zuweisen. Die **Semantik** soll also sowohl sprachlichen Ausdrücken eine Extension (also eine Menge von Objekten) zuordnen als auch die Bedingungen explizieren, unter denen ein Ausdruck sich auf diese Menge bezieht (Intension). Es ist also alles so ähnlich wie bei der Logik, nur komplizierter, weil nicht von vornherein eine eingeschränkte Kunstsprache vorliegt. Der Sinn oder die Intension eines Ausdruckes wird als eine Funktion mit möglichen Welten als Argumenten und möglichen Individuen (Objekten) als Wertebereich betrachtet. Der Sinn oder die Intension eines Satzes ist eine Funktion mit möglichen Welten als Argumente und Wahrheitswerten als Wertebereich. Die Intension bestimmt also, wann ein Satz wahr oder falsch ist. Daß er wahr ist, macht - nach wie vor - seine Bedeutung aus. Bei natürlichen Sprachen ist außerdem ein

[33] Logik, Produktionensysteme, Funktionstheorie, Termersetzungssysteme, Automatentheorie

Gebrauchskontext (z.B. im Restaurant, nach der Äußerung eines bestimmten Textes, von wem geäußert,...) zu berücksichtigen. Also sind nicht nur mögliche Welten, sondern auch Gebrauchskontexte Argument der Sinnfunktion. Ein Paar, bestehend aus einer möglichen Welt und einem Gebrauchskontext, heißt Referenzpunkt. Montague konstruierte die semantische Funktion für Sätze so, daß sie Referenzpunkte als Argumente und Wahrheitswerte als Werte hat.[34] Damit sind bestimmte pragmatische Bedingungen bereits in die Semantik einbezogen: wer von wo aus spricht, Zeigegesten, welcher Kommunikationskanal gewählt wurde (Telefon ist zeitgleich, raumverschieden, ein hinterlassener Zettel ist raumgleich, zeitverschieden). Andere Autoren rechnen solche Bedingungen nicht zur Semantik.

Die **Pragmatik** beschäftigt sich mit sprachlichem Handeln und Kommunikationsverhalten. Die Dialoggestaltung mit ihren Regeln der Übernahme der Initiative (turn taking), dem regelhaften Übergang von einem Themenkomplex zum anderen, den Kommunikationsmaximen und dem Aushandeln von Rollen bei den Kommunikationspartnern kann als zentrales Untersuchungsgebiet betrachtet werden. Waren bei der Syntax die Strukturbeschreibung und bei der Semantik Sinn und Bedeutung die zentralen Begriffe, so muß bei der Pragmatik das (kommunikative oder sprachliche) Handeln konstruktiv definiert werden. Ob die Einflüsse von Vorwissen, Zeit und Raum der Redepartner der Semantik oder der Pragmatik zugeordnet werden, ist bei verschiedenen Autoren unterschiedlich. Natürlich werden in der Pragmatik weiterhin syntaktische und lexikalische Einheiten behandelt. An ihnen ist jetzt nur ein anderer Aspekt interessant. Ein Zettel an der Tür "bin in einer Stunde zurück" oder der Eintrag in das Logbuch eines Schiffes "der Kapitän war heute nüchtern" sind eben auch Sätze, Ketten von Phonemen. Die Bezeichnungen "Syntax", "Semantik" und "Pragmatik" beziehen sich nicht auf unterschiedliche Dinge, sondern auf unterschiedliche Fragestellungen bei der Untersuchung dieser Dinge.

6.1.3 Anwendungen sprachverarbeitender KI

Die Anwendungen sprachverarbeitender KI sind direkt oder indirekt. Direkte Anwendungen sind zum Beispiel:

- maschinelle Übersetzung wie beim kanadischen Wetterdienst, wo automatisch zwischen englisch und französisch übersetzt wird, oder Übersetzungsunterstützung;
- textverstehende Systeme, bei denen ein System aus Texten Information entnimmt *(message understanding, information extraction)*;
- Zugangssysteme (Frage-Antwort-Systeme oder Dialogsysteme), bei denen Informationen (z.B. aus einer Datenbank) in natürlicher Sprache erfragt und, bei Dialogsystemen, auch natürlichsprachlich bereitgestellt werden können;
- Schreibunterstützungssysteme, die Texte auf Rechtschreibung, Grammatik und Stil hin untersuchen und Verbesserungen vorschlagen.

---

[34] Die Zusammenfassung der Zielsetzungen von Syntax und Semantik lehnt sich stark an die von Wolfgang Stegmüller an: "Hauptströmungen der Gegenwartsphilosophie, Band II", Stuttgart, 1987. Eine ausführlichere, linguistische Einführung gibt Dieter Wunderlich "Grundlagen der Linguistik", Reinbek, 1974. Zu speziellen Verfahren und Theorien sind beide Bücher überholt, nicht aber zum grundsätzlichen Ansatz. Die Arbeit von Montague kann am besten bei Dowty nachgelesen werden, nicht bei ihm selbst, da er zu früh starb, als daß er es noch verständlich darstellen konnte.

Durch die ungeheure Fülle von natürlichsprachlichen Dokumenten im Internet oder auch einfach in Freitext-Einträgen von Datenbanken ist der Bedarf an textverstehenden Systemen stark angestiegen. Dabei wird nicht erwartet, daß die Texte gründlich analysiert werden. Vielmehr sollen die sprachlichen Strukturen soweit analysiert werden können, daß eine ungefähre Übersicht über den Inhalt des Textes gegeben wird. Hier soll die rein statistische Herangehensweise, die die sprachlichen Strukturen außer acht läßt, ersetzt werden. Die linguistische Analyse kann aber auch einer statistischen Verarbeitung vorausgehen (z.B. die Reduktion der Wörter im Text auf ihre Stammformen (Morphologische Analyse 6.3.16.3.1) vor der statistischen Analyse).

Als ebenso wichtig wie die direkten Anwendungen erscheinen mir aber die indirekten Anwendungen linguistischer oder kommunikationswissenschaftlicher Forschung in der Informatik:

- Selbst wenn kein natürlichsprachlicher Satz oder Text von einem System verwendet wird, sollte das System in seiner Interaktion mit Menschen nach deren Regeln der Kommunikation gestaltet werden. Selbst wenn mithilfe graphischer Objekte kommuniziert wird, so gelten doch die Regeln linguistischer Pragmatik, nach denen Menschen unbewußt handeln.

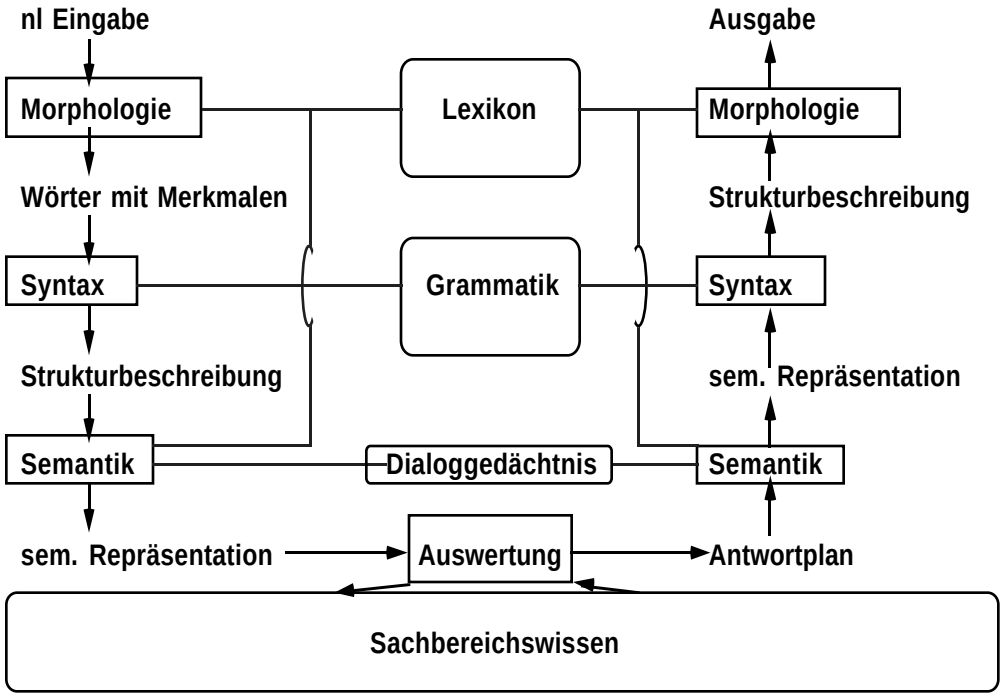
- Rechnereinsatz verändert Kommunikationsabläufe in Firmen oder Behörden. Wenn dieser Aspekt nicht berücksichtigt wird, kann statt einer Steigerung der Produktivität und Effektivität deren Minderung hervorgerufen werden.

- Die neuen Medien (e-mail, world wide web) bieten neue Möglichkeiten sprachlichen Handelns. Diese Möglichkeiten können besser genutzt werden, wenn die Systeme zu ihrer Unterstützung auf der Grundlage linguistischer Erkenntnisse, zumindest aber mit sprachlichem Fingerspitzengefühl entworfen werden.

Da die KI stets den Menschen mit seinen Fertigkeiten und Disziplinen wie Psychologie, Linguistik, Philosophie mit ihrem Wissen beachtet hat, kann die KI-Ausbildung vielleicht etwas von dem nötigen Feingefühl bei der Gestaltung und Einführung von Systemen vermitteln.

6.1.4 Überblick über ein sprachverarbeitendes System

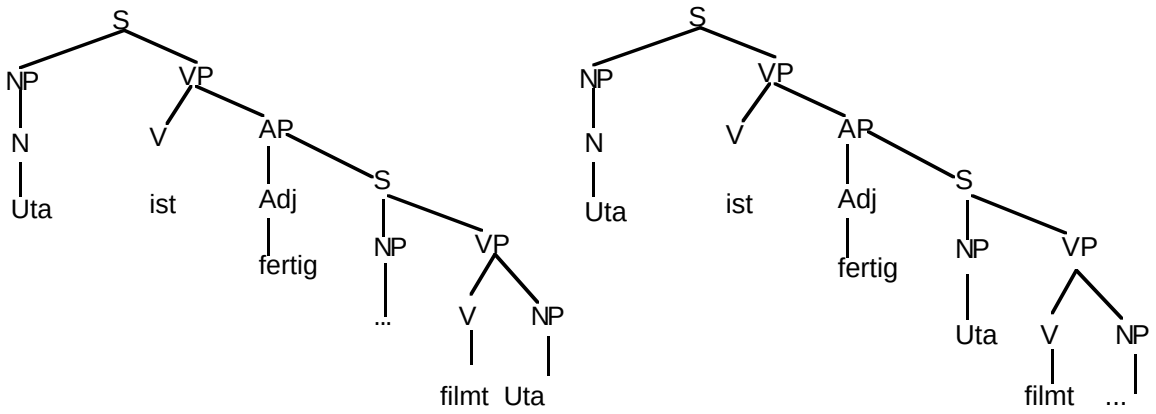
Als Orientierung für die folgende Behandlung der Komponenten und ihrer Wissensquellen wird hier ein idealisiertes natürlichsprachliches System (NLS) dargestellt. Dabei wird nicht festgelegt, ob dieses System ein Zugangssystem, ein Schreibunterstützungssystem oder Teil eines Übersetzungssystems ist. Die 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.

Das Wissen über Wörter ist im Lexikon abgelegt. Die **Morphologie** analysiert Wortformen oder generiert eine flektierte Wortform. So wird z.B. das zusammengesetzte und flektierte Wort "Kaufhäusern" in die Bestandteile Kauf, Haus und Akkusativ-Plural-Endung zerlegt. Schon so eine Zerlegung kann schwierig sein. Wenn z.B. gerade von einer Pferdenärrin namens Blumento die Rede war und danach das Wort Blumentopferde kommt. Flexionen werden ebenfalls in der Morphologie behandelt. So wird aus lauf und 3. Person Singular, Präsens, Indikativ wird läuft generiert. lauf gehört zur Klasse der im Präsens regelmäßigen Verben mit Umlautung. Im Lexikon steht auch, daß laufen ein intransitives Verb (vi) ist. Für so ein einfaches Wort wie Hans wird im Lexikon wohl nur Nomen, vielleicht auch das Merkmal mensch stehen, da es viele Verben gibt, die nur bei Menschen verwendet werden (essen statt fressen). Natürlich gibt es mehrdeutige Wörter wie z.B. Schloß, denen zwei verschiedene Bedeutungen zugeordnet werden, schloß1 und schloß2.

Die **Syntax** verwendet bei der Analyse das Ergebnis der Morphologie oder gibt sein Ergebnis bei der Generierung an die Morphologie weiter. Wenn das Subjekt des Satzes im Singular ist, so muß das flektierte Verb in der 3.Person Singular sein. Kategorien wie Subjekt oder intransitives Verb werden in der Grammatik verwendet, um Regeln des Satzbaus auszudrücken. Eine einfache Regel teilt jeden Satz in einen Nominal- und einen Verbteil. Bei transitiven Verben enthält der Verbteil dann wiederum einen Nominalteil, bei intransitiven Verben nicht. Manche Sätze sind syntaktisch mehrdeutig. So kann bei dem Satz Uta ist fertig zum Filmen entweder Uta filmen oder jemand filmt Uta. Das ergibt unterschiedliche Satzstrukturen:



Bei der Analyse natürlichsprachlicher Äußerungen wird das Ergebnis der syntaktischen Analyse an die **Semantik**-Komponente weitergegeben. Semantisches Wissen ist einerseits auf Zustände in einer Welt gerichtet, in der ein Satz wahr ist. Wir können uns dies als A-Box eines Termsubsumptionssystems, als Wissensbasis oder als Datenbank vorstellen. Andererseits ist semantisches Wissen auf die Repräsentation von Begriffen und ihre Relationen gerichtet. Wir können uns dies als T-Box eines Termsubsumtionssystems vorstellen. Es wird eine semantische Wohlgeformtheit definiert, die etwa Chomskys berühmten Satz Grüne Ideen schlafen wütend ablehnt. Die Semantik kann sinnlose Interpretationen eines Satzes ausschließen und somit manche syntaktisch mehrdeutigen Sätze eindeutig machen. Müllers sahen den Film auf dem Flug nach New York drückt natürlich aus, daß Müllers geflogen sind und sich dabei einen Film ansahen - Filme können nicht fliegen. Hier sieht man, daß Wissen über sprachliche Begriffe und Wissen über die Welt eng verknüpft sind. Deshalb zählen einige Linguisten solche Auflösung von Mehrdeutigkeiten zur Pragmatik. Die Semantik hat außerdem mit semantischen Mehrdeutigkeiten zu kämpfen. Udo möchte eine Künstlerin heiraten hat die folgenden zwei Bedeutungen:

$\exists$ x | künstlerin (x) & möchte(hans, heiraten(hans, x)) oder

möchte (hans, $\exists$ x| künstlerin(x) & heiraten(hans, x)).

Im ersten Fall gibt es eine bestimmte Frau, die Hans heiraten möchte und diese ist Künstlerin. Im zweiten Fall möchte Hans eine Künstlerin heiraten -- welche auch immer.

Die **Pragmatik** verbindet Äußerungen mit dem sprachlichen und außersprachlichen Kontext. Das Text- oder Dialogwissen beschreibt Regelhaftigkeiten beim Dialog bzw. beim Textaufbau. Der Bezug zu dem vorher geäußerten Text ermöglicht die Auflösung von Anaphora. Anaphora sind rückbezügliche Äußerungen wie z.B. Pronomen. In Hans läuft. Er will den Bus erreichen. bezieht sich er auf Hans. Auch solche Anaphora können mehrdeutig sein. Hans ließ das Glas auf das Parkett fallen und zerbrach es. Hier könnte das Glas oder das Parkett zerbrechen. Wissen über die Zerbrechlichkeit von Gegenständen erlaubt die Auflösung der Mehrdeutigkeit.

Das Dialoggedächtnis speichert, wer was worüber bereits gesagt hat und stellt den Bezug zwischen den Äußerungen verschiedener Dialogpartner her. Wenn sich zwei gegenübersitzen, kann das rechte Regal und das linke Regal sich auf dasselbe Regal beziehen. Auch das im Partnermodell gespeicherte Vorwissen sowie die angenommenen Ziele der Dialogpartner werden berücksichtigt, soweit sie aus dem vorangegangenen Text hervorgehen. Wenn z.B. Mirjam sagt Der Bus zum Bahnhof Langendreer fährt um 10 nach 12 und hinterher von dem 10 nach Bus spricht, so meint sie den Bus zum Bahnhof Langendreer, auch wenn der tatsächlich erst 15 Minuten nach 12 fährt.

Die Pragmatik greift nicht nur auf linguistisches Wissen zu, sondern berücksichtigt auch einen Handlungszusammenhang, der sprachlich sein kann, aber nicht sein muß. Wenn z.B. Uta, Hans und Mirjam gemeinsam den Bus erreichen wollen, Hans losläuft, Mirjam ebenfalls schneller wird, Uta aber ruhig weitergeht, so steht Mirjams Äußerung Hans läuft in einem ganz bestimmten Zusammenhang. Die Äußerung kann dann ein Vorwurf an Uta sein und eine Aufforderung, ebenfalls zu laufen.

6.2 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 exponentiell. 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 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.

6.2.1 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
... jagt den kleinen Hasen

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

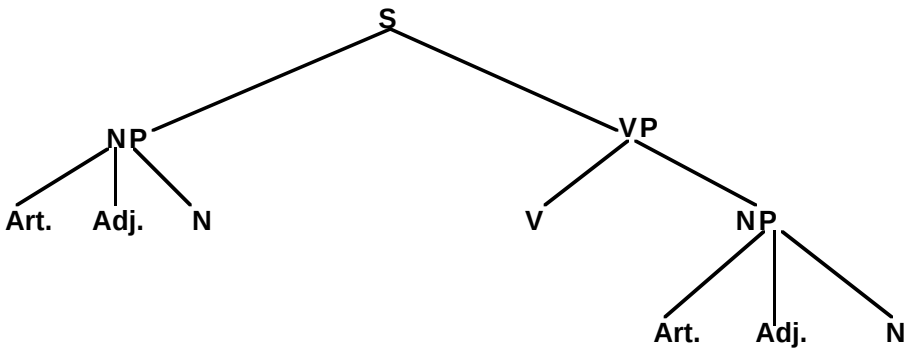
Wir können uns auch fragen, in welchen Kontexten jagt vorkommen kann. Auf diese Weise erhalten wir die Distribution von Teilsätzen oder Wörtern. Alle Wörter oder Teilsätze, die im selben Kontext vorkommen können, sind distributionsäquivalent. Im Beispiel sind {schwarze, graue, verfressene}, {Die, Omas}, {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 (von oben nach unten) in einer 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: Hans, 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 (mit Akkusativobjekt): jagt, ditransitiv (mit 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,

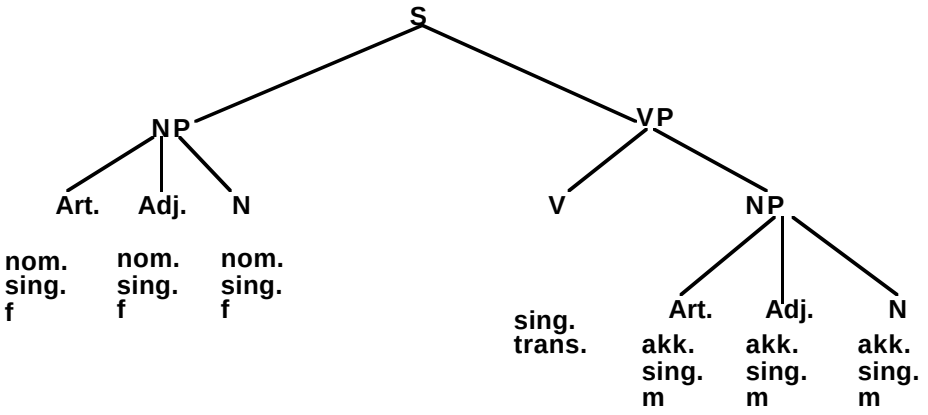
Genus: feminin, masculin, neutrum,

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.

6.2.2 Parsing in Prolog

> 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 (Rückziehverfahren). Wir brauchen lediglich die Grammatikregeln als Prolog-Klauseln zu schreiben!

Grammatik:	**Prolog-Klauseln**
S --> NP VP	`s(L1,L):- np(L1,L2),vp(L2,L).`
NP --> Det N	`np(L1,L):- det(L1,L2),n(L2,L).`
Det --> die	`det([die|L],L).`
N --> Katze	`n([katze|L],L).`
N --> Ratte	`n([ratte|L],L).`
VP --> V NP	`vp(L1,L):- v(L1,L2),np(L2,L).`
V --> jagt	`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.

6.2.2.1 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!

6.2.3 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]`.

6.2.4 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].
```

6.2.5 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 darstellen. 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.

6.2.6 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.

> 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.

6.3 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.

> 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:

> 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.

> **Morpheme** sind die kleinsten bedeutungstragenden Einheiten eines Wortes.

> **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.

> 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. Die große Produktivät unserer Sprache drückt sich auch in der Wortbildung aus. So kommen beispielsweise nur 70% der Wörter eines Zeitungsausschnittes in unseren Wörterbüchern vor, da sehr viele ad hoc Bildungen existieren. Die Behandlung Komposita ist nicht einfach. Probleme sind zum einen auf orthographischer Ebene zu finden, wie beispielsweise Fugenzeichen. D.h. manchmal werden Buchstaben weggelassen oder auch hinzugenommen wie beispielsweise das s innerhalb von Bahnhofshalle oder das f in Schiffahrt. Noch problematischer ist die Bestimmung der Eigenschaften von zusammengesetzten Wörtern aus ihren Einzelteilen. Beispielsweise können die semantischen Eigenschaften des Wortes Schweineschnitzel nicht nach denselben Kriterien aus den Einzelteilen gewonnen werden werden, wie dies bei dem Wort Jägerschnitzel erfolgen kann.

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. 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.

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

6.3.1 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, person:P,numerus:Num,temp:T)-->
      stamm(Stamm,Kat),
      suffix(Suffix, person:P,numerus:Num,temp:T).

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

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

% Zum Testen nötige Grammatik
s(s(NP,VP)) -->
      np(NP, person:P, numerus:Num, kasus:nom),
      vp(VP, person:P,numerus:Num,temp:T).

      np(np(N), person:P, numerus:Num, kasus:nom) -->
      pron(N, person:P, numerus:Num, kasus:nom).

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

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

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! Selbst dieses kleine Beispiel zeigt aber 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.

6.3.2 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. Eine feinere Einteilung der Verben in Subkategorien kann durch folgende Regeln angegeben werden:

```
%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. Auch können Merkmale eingesetzt werden, um die Präpositionen anzugeben, die zu einem Verb gehören.

```
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].
```

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 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.

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

Hier begegnet uns also wieder die Repräsentation von EPAM, wobei ein Wort aber auch in der Reihenfolge seiner Buchstaben angegeben werden kann (nicht: erster und letzter Buchstabe an erster und zweiter Stelle). 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.

6.4 Semantik

Bisher haben wir lediglich die Struktur eines Satzes formalisiert. In diesem Abschnitt 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. Analog zu der oben angeführten Unterscheidung in Sinn und Bedeutung der Logik, gibt es die Begriffe:

> Die **extensionale Bedeutung** (Referenz) eines Ausdrucks ist das Objekt, auf den er sich bezieht, oder (bei Sätzen / Formeln) der Wahrheitswert (siehe Kap. 3.2) Das Objekt ist der **Referent** oder das Referenzobjekt.
>
> Die **intensionale Bedeutung** eines Ausdrucks ist eine Beschreibung oder ein qualitativer Sachverhalt.

Die Intension entspricht dem Sinn, die Extension entspricht der Bedeutung. Ein 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.

> Das Prinzip der **Kompositionalität** besagt, daß die Bedeutung eines Ausdrucks sich aus der Bedeutung seiner Teile ergibt.
>
> 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 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)`	`∃X arbeitet(X)`	`∀X|an_uni(X)->arbeitet(X).`

6.4.1 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.

> 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, weil sich auf diese Weise am leichtesten (d.h. ohne weitere Vorarbeiten) vorführen läßt, wie

man von syntaktischen Strukturen zu Ausdrücken in SRS kommt und wie man sie dann auswertet.

Wir führen ein Prädikat sem ein, das für ein Lexem dessen Bedeutung (semantische Eigenschaften) angibt. Die Eigenschaften, die sich auf dasselbe Objekt beziehen, werden durch die Funktion l zusammengehalten. Diese Funktion lehnt sich an den Lambda-Operator an -- ein Operator, der Variablen als Platzhalter für ein Objekt einführt. An erster Stelle der Funktion l steht eine Variable, an zweiter Stelle ein Ausdruck, der wiederum eine l-Funktion sein kann. Im Lexikon wird für jedes Wort (jeden Wortstamm) im Prädikat sem die l-Funktion mit konkreten Werten angegeben.

```
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].
```

Wir sehen hier, wie das transitive Verb zwei Platzhalter einführt: einen für den Jäger, einen für den Gejagten. Die Nomina erhalten als semantisches Merkmal lediglich ihren Bezeichner. Dies kann ersetzt werden durch eine Menge definierender Merkmale, ewa in Form einer Begriffsdefinition in einer T-Box.

6.4.2 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. Hier wählen wir die zweite Möglichkeit.

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 der Lesbarkeit halber nur angedeutet.

```
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. 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
```

6.4.2.1 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	∀X mensch (X) --> sterblich (X)
eine Katze jagte eine Ratte	∃X katze(X) & (∃Y ratte(Y) & jagt(X,Y))

Wie 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 `(∃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.

```
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

```
Nº1

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])

Nº1

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))))
```

6.4.3 Auswertung der semantischen Beschreibung

Wir haben eine semantische Beschreibung erstellt und wollen nun die Satzsemantik über dem Weltwissen auswerten. 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, Udo, eine Ratte, Ulf, jagt. Für die Auswertung nehmen wir die aufgebaute quantifizierte Formel.

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

Wir wollen 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)), !.                       %Existenzprüfung

%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 Existenzprüfung 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, 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)
Nº1
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)
Nº1
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)
Nº1
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.

Je nachdem, was das Ziel des NLS ist (z.B. natürlichsprachliches Zugangssystem zu einer Wissensbasis oder zu einer Datenbank) wird die SRS und ihre Auswertung unterschiedlich gestaltet. So kann z.B. ein Ausdruck in der SRS in einen SQL-Ausdruck überführt werden. Ein Datenbanksystem übernimmt dann die Auswertung des SQL-Ausdrucks. Das Ergebnis der Auswertung ist dann eine Tupelmenge.

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. Für diese semantische Beschreibung 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.

6.5 Literatur

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

De Smedt, K.J.M.J. (1990): Incremental Sentence Generation, Ph.D. Thesis, Katholieke Universiteit Nijmegen, Report 90-01.

Dowty, David R., Wall, Robert E., Peters, Stanley (1989): Introduction to Montage Semantics, Boston: Reidel, 1989

Gazdar, G. & C. Mellish. (1989): *Natural Language Processing in PROLOG*. Workingham, England: Addison-Wesley.

Günther Görz (ed) Einführung in die künstliche Intelligenz, Bonn: Addison-Wesley, 372 - 424

Levelt, W.J.M. (1989): Speaking: From Intention to Articulation, Cambridge: MIT Press.

Stegmüller, W. (1987): *Haupströmungen der Gegenwartsphilosophie*. 2 Band. Stuttgart: Alfred Kröner Verlag

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

**Einen guten Überblick bieten die Artikel in: *Readings in Natural Language Processing*. B. Grosz, K. Sparck Jones & B.L. Webber. Los Altos, CA: Morgan Kaufmann.

7 Kritik an der KI

Die KI ist von Anfang an heftig kritisiert worden. Zu großen Teilen bezieht sich die Kritik auf Gefahren, die mit dem Einsatz von Rechenanlagen immer verbunden sind. Es sind dies also Befürchtungen, die auf alle Bereiche der Informatik gleichermaßen zutreffen. Daß sie gerade gegenüber der KI vorgebracht werden, mag an der Vorreiterrolle der KI liegen: oft erschließt die KI der Operationalisierung einen Bereich, der vorher noch unformalisiert war. Viele haben vergessen, daß auch das Rechnen einmal eine rein menschliche Fähigkeit war.

Ein weiterer Grund für die Häufigkeit von Kritik ist das Mißverständnis, das von den KI-Wissenschaftlern selbst verschuldet ist. Tatsächlich bezieht sich Kritik oft auf Schriften, die innerhalb der KI keinen besonderen Stellenwert haben, außerhalb der KI aber stark rezipiert werden. Es sind dies insbesondere Schriften, die das Paradigma der KI beschreiben wollen. Oft sind dies populärwissenschaftliche Schriften. Die meisten KI-Wissenschaftler explizieren nicht ihre Annahmen und den Rahmen, in dem sie arbeiten, sondern schreiben über ihre Ergebnisse innerhalb dieses Rahmens. Damit bleiben ihre Arbeitsgrundlagen implizit und der Widerspruch zwischen diesen und den in einigen Übersichtsartikeln dargestellten Annahmen wird nicht deutlich. Da insbesondere die im Beschreibungsansatz der KI arbeitenden Wissenschaftler nur selten metatheoretische Aussagen gemacht haben,[35] habe ich hier diesem Ansatz mehr Raum gegeben als den üblicherweise beschriebenen.

7.1 Physical Symbol System Hypothesis und Wissensebene

Das häufigste Mißverständnis soll hier einmal kurz beschrieben werden. Es geht auf die Arbeiten von Newell und Simon (1958) zurück. Die "Physical Symbol System Hypothesis", mit der Newell 1980 versuchte, seine grundlegenden Annahmen zu verdeutlichen, ebenso der grundsätzliche Aufsatz von Newell (1982) über die Wissensebene wird häufig als das Paradigma der KI betrachtet. Interessanterweise erwähnen die wichtigsten KI-Einführungsbücher die erste Schrift überhaupt nicht. Zwei Kernpunkte werden von Newell herausgestellt:

- Prozesse können unabhängig von ihrer physikalischen Basis betrachtet werden.
- Tätigkeiten von Menschen können nur beschrieben werden, wenn man die Ziele, die sie verfolgen, mit einbezieht.

Mit dem ersten Punkt sollen Beschreibungsebenen unterschieden werden. Es soll verdeutlicht werden, daß uns die genaue Beschreibung physischer Abläufe nicht dabei hilft, mentale Vorgänge zu erfassen. Die Tätigkeit des Essens wird in vielen Zusammenhängen nicht dadurch angemessen beschrieben, daß wir die Kiefer, Zähne, Muskeln und den Magen medizinisch darstellen. Wenn wir einen Restaurant-Besuch beschreiben wollen, müssen wir vielmehr darlegen, daß Essen ein Grundbedürfnis des Menschen ist, es für die meisten Menschen erfreulich ist zu essen und gemeinsames Essen als geselliges Ereignis betrachtet wird. Simon wählt als Beispiel Newtons Gravitationsgesetz, das man als "Newton-Maschine" bezeichnen könnte, die Planetenbewegungen vorhersagt. Wir könnten sagen, "da

[35] Eine ausgezeichnete Studie über operationale Modelle ist Clancey (1989). Es ist eines der seltenen Papiere, die von einem in der KI aktiven Wissenschaftler geschrieben sind und grundlegende Annahmen diskutieren. Er greift Newells Wissensebene auf und klärt die verschiedenen Interpretationsebenen.

die 'Newton-Maschine' aus einer Reihe von Differentialgleichungen besteht, mit Tinte auf Papier geschrieben, und da wir wissen, daß die Planeten nicht aus Tinte bestehen und der Himmel kein Stück Papier ist, kann die ganze Theorie gar nicht stimmen; jeder würde wohl zugeben, daß eine derartige Argumentation barer Unsinn wäre." (Simon 1977:17)

Der erste Punkt bedeutet auch, daß alle Beschreibungen kognitiver Tätigkeiten sich nicht auf einen physischen Ort beziehen. Das heißt, es wird nicht behauptet, daß Sprachverstehen, Lernen oder Szenenanalyse sich in einem Körper lokalisieren lassen, bzw. sich einem körperlichen Ort zuordnen lassen (z.B. einem Ort im Gehirn). Von einer physischen Realisierung wird abstrahiert. Verschiedene physische Realisierungen sind für die Beschreibung gleichwertig. Damit kann die Beschreibung nicht mehr zwischen auf einem Digitalrechner, auf einem Analogrechner, im Gehirn oder im Rückenmark realisierten Prozessen unterscheiden. Die Kritik besagt, daß von physikalischen Prozessen gerade nicht abstrahiert werden darf (s. 1.2.2.1). Das Mißverständnis besteht darin, daß eine Gleichsetzung von Rechner-Systemen mit menschlicher Gehirntätigkeit unterstellt wird. Diese Gleichsetzung ist durch den Unterschied zwischen Beschreibung und Beschriebenem jedoch ausgeschlossen.

Der zweite Punkt ist keinesfalls eine Erfindung von Newell. Schon Dray hat 1957 darauf hingewiesen, daß rationale Erklärungen die Ziele und Überzeugungen eines Menschen unterstellen müssen (Dray 1957). Newell sagt nun, daß in die Beschreibung von Tätigkeiten nur solche Annahmen einbezogen werden, die einem rationalen Akteur unterstellt werden <u>müssen</u>, um sein Verhalten beschreiben zu können. Seine Konstruktion geht so: Wenn ein rationaler Akteur ein Ziel Z hat und weiß, daß er mithilfe der Handlung H dieses Ziel erreichen kann, und er kann H ohne Nachteile ausführen, dann führt er H aus. Eine Tätigkeitsbeschreibung besteht also aus Zielen, Handlungen und Wissen über die Situation. Sie gibt an, welches Ziel und welches Wissen einem Akteur berechtigterweise unterstellt (zugeschrieben) werden kann - egal, ob er dieses Ziel und Wissen nun wirklich hat, oder nicht. Es ist der Versuch, Beschreibungen nicht ins Uferlose oder Spekulative abgleiten zu lassen, auch wenn wir keine direkten Beobachtungen zur Eingrenzung heranziehen können.

Innerhalb der KI wurde hervorgehoben, daß Menschen die inferentielle Hülle ihres Wissens nicht vollständig zur Verfügung steht (Konolige 1984). Es gibt also einen Unterschied zwischen dem, was man wissen könnte, und dem, was als Wissen für Handlungen herangezogen wird. Dadurch kann dann die Tätigkeitsbeschreibung nicht mehr zur Vorhersage von Tätigkeiten bei einem bestimmten Wissensstand genutzt werden. Eine Reparatur wäre es, wenn nur aktuell verfügbares Wissen in das Handeln einginge. Aber: welches ist das?

7.2 Der chinesische Raum

John Searle hat mit einem Gedankenexperiment 1980 eine lange Diskussion ausgelöst, die immer wieder beschrieben wurde. Das Gedankenexperiment soll festlegen, wann wir kognitive Tätigkeiten zuschreiben dürfen, und wann nicht. Insofern ist es sehr ähnlich dem Turing-Test. Während der Turing-Test auf die Verhaltensähnlichkeit von Mensch und Maschine (ähnlich der Verhaltensähnlichkeit von Frau und Mann) abzielt, will Searle mit seinem Experiment den Unterschied zwischen äußerem und innerem Verhalten verdeutlichen.

Ein Mensch, der über keinerlei Kenntnisse der chinesischen Sprache verfügt, sitzt in einem Raum. Er hat eine Menge von Karteikarten, auf denen Regeln stehen. Die Regeln geben an, was der Mensch mit bestimmten chinesischen Zeichen tun soll. In den Raum werden nun Karten mit Fragen in chinesischer Sprache hineingereicht. Der Mensch befolgt die Regeln, die auf seinen Karteikarten

stehen, und kann so selbst Zettel anfertigen, die er herausreicht. Wenn die Regeln auf den Karteikarten alle richtig und vollständig sind, kann er Fragen in chinesisch auf chinesisch richtig beantworten. Dabei versteht er nach wie vor überhaupt kein Chinesisch und könnte nicht sagen, worum es bei Fragen und Antworten überhaupt ging. Der Mensch würde den Turing-Test bestehen, weil er sich verhält wie jemand, der das Chinesische beherrscht. Dennoch wissen wir, daß der Mensch das Chinesische nicht beherrscht.

Sprachwissenschaftler würden sich unendlich freuen, diese Karteikarten zum Chinesischen zu sehen, die eine vollständige und korrekte Syntax und Semantik dieser Sprache zu sein scheinen. Die Karteikarten zusammen mit dem Menschen stellen ein operationales Modell der Sprachverarbeitung dar. Das allein würde ja schon genügen. Wir müssen dann nicht unterstellen, daß Mensch und Karten tatsächlich chinesisch können.

Man kann aber behaupten, daß der Raum, also Mensch und Karteikarten, chinesisch können. Nicht die einzelne Person, wohl aber der Raum als Ganzes, beherrscht die chinesische Sprache. Dieses ist das wesentliche Argument der KI. Interessant ist Searles Antwort: "Die Vorstellung ist die, daß eine Person zwar kein Chinesisch versteht, aber die Konjunktion dieser Person mit ein paar Zetteln irgendwie Chinesisch verstehen soll. Es ist für mich schwer begreiflich, wie jemand, der nicht in den Fängen einer Ideologie zappelt, diese Vorstellung auch nur einen Moment lang plausibel finden kann." (Searle 1982, zitiert in Turkle (1984:329 f)) Searle hält es für selbstverständlich, daß es einen Ort des Denkens geben müsse. Dieser Ort soll ein Subjekt sein. Alles andere ist für ihn "Ideologie". Umgekehrt kann man natürlich Searle vorwerfen, daß er nicht ein einziges Gegenargument präsentiert hat, sondern lediglich eine Ideologie. Vielmehr können wir Gründe anführen, warum Wissen repräsentiert, aber niemals "in der Hand gehalten" werden kann (Clancey 1989:289).

"Searle hält beharrlich an dem Vorrang von Dingen gegenüber Prozessen fest. Er sucht nach einem Menschen in dem Raum, nach einem Neuron im Gehirn, nach einem Ich im Denken. Seine AI-Opponenten halten mit der gleichen Beharrlichkeit am Vorrang von Prozessen gegenüber Dingen fest. Woraus ein System besteht - ob aus Silikon, Metallteilen oder Flüssigkeiten -, ist dabei vollkommen irrelevant.

Die Dramatik, die in der Konfrontation der beiden Kulturen liegt, kommt auch in einer fundamental unterschiedlichen Wahrnehmung zum Ausdruck. Für Searle ist Behauptung "Ein Raum denkt" per definitionem absurd. In der AI-Kultur gilt die Überzeugung, daß so etwas unmöglich sei, als archaischer Glaube. Für sie ist die Auffassung, daß in dem Raum ein Wesen "das Denken erledigen" müsse, nichts weiter als die moderne Fassung der Vorstellung, daß sich in der Zirbeldrüse eine "Seele" befinde." (Turkle 1984:331)

7.3 Literatur

Als Leseempfehlung für diejenigen, die sich intensiver mit dem in diesem Kapitel angesprochenen Stoff beschäftigen möchten, gelten nur die mit * markierten Literaturhinweise. Die anderen sind der Überprüfbarkeit wegen und als Verweis auf Spezialwissen zu einzelnen Punkten aufgeführt. Wer genau einen weiteren Artikel lesen möchte, soll sich einen der beiden ** markierten aussuchen.

Chomsky, Noam (1964): Aspects of the Theory of Syntax, deutsche Übersetzung 1969 Frankfurt a.M.: Suhrkamp

**Clancey, William (1989): The Knowledge Level Reinterpreted - Modeling How Systems Interact, in: *Machine Learning* 4, No.3/4

*Dietz, Peter (1996): Menschengleiche Maschinen, Vorlesungsskript WS 96/97, Universität Dortmund

Dray, W. (1957): Laws and Explanation in History, angeführt in Wunderlich (1974:101)

Konolige, Kurt (1984): Belief and Incompleteness, CSLI Report 84-4, Stanford University

Newell, Alan/Simon, Herbert (1958): Heuristic Problem Solving: The Next Advance in Operations Research, in: *Operations Research* 6

Newell, Alan (1980): Physical Symbol Systems, in: *Cognitive Science* 4

**Newell, Alan (1982): The Knowledge Level, in: *Artificial Intelligence* 18

Searle, John (1980): Minds, Brains, and Programs, in: *Behavioral Sciences* 3

Simon, Herbert (1977): The Sciences of the Artificial, Cambridge

Simon, Herbert (1978): Acht Vorlesungen über kognitive Psychologie, gehalten an der Universität Hamburg, Fachbereich Psychologie, 1978

*Simon, Herbert (1989): Foundations of Cognitive Science, in: Posner (ed) *Foundations of Cognitive Science*, Cambridge (Mass): MIT Press

*Turkle, Sherry (1984): Die Wunschmaschine - Vom Entstehen der Computerkultur, Reinbek: Rowohlt

8 INDEX

—F—

—G—

—H—

—I—

—K—

—L—

—M—

—N—

—O—

—P—

—Q—

—R—

—S—

—T—

—U—

—V—

—W—

—Z—