

**Ein
Fallschablonenzerteiler
für
Deutsch**

Diplomarbeit von cand. inf. Lutz Prechelt

Teil I

Betreut von Dipl. Inf. Rolf Adams

Fakultät für Informatik
Institut für Programmstrukturen und Datenorganisation
Universität Friedericiana, Karlsruhe

31. Oktober 1989

Ich erkläre hiermit eidesstattlich, daß ich diese Diplomarbeit selbständig und ohne unzulässige Hilfe angefertigt habe.

Die verwendeten Quellen sind im Literaturverzeichnis vollständig angegeben.

Karlsruhe, den 14. Oktober 1989

Lutz Prechelt

Zusammenfassung

Diese Arbeit beschreibt ein System zur Verarbeitung natürlicher Sprache (Deutsch), das auf Fallschablonen als Grammatikformalismus und einem Chart-Parser als Zerteilungsverfahren basiert.

Das System erhält eine eigene Wissensbasis und verwendet keine anderen Programme oder Datenbestände. Es liefert im Dialog in weniger als einer Sekunde aus den Eingaben des Benutzers anhand der Wissensbasis instanziierte Fallschablonen, eine Art Normalform der Eingabe, die relativ einfach weiterverarbeitet werden kann.

Die syntaktische Leistungsfähigkeit des Zerteilers umfaßt die Unterscheidung von Aussagesätzen, Imperativsätzen, Ja/Nein-Fragen und W-Fragen (alles ohne Bedarf von Satzzeichen). Ferner die Handhabung von Modalverben, von Nominalphrasen mit Präpositionen, Artikeln/Pronomen und Adjektivketten, von Adverbien, sowie von Relativsätzen und von Nebensätzen, die mit einer Konjunktion eingeleitet werden.

Das System ist portabel und wird zur Zeit eingesetzt, um natürlichsprachliche Hilfssysteme zu gestalten (bislang z.B. für L^AT_EX).

Abstract

This work describes a system for processing german natural language input, that is based on the grammar formalism of case frames and uses an active chart parser. A dedicated knowledge base is given to the system; it does not use any other program or data. It interactively accepts natural language user input and returns its results within less than one second. The results are instantiated case frames, a kind of normalized form of the input, that is easy to process further.

The syntactic power of the parser includes the discrimination of declarative and imperative sentences, yes/no-questions and wh-questions, all without a need of any punctuation. Further features are the handling of modal verbs, nominal phrases (with prepositions, articles/pronouns and adjectives), adverbs, relative clauses and subclauses that start with a conjunction.

The system is portable and is currently being used for the implementation of natural language help systems (e.g. for L^AT_EX).

Inhalt

Vorbemerkungen	13
Zum Aufbau dieser Arbeit	13
Zu den Definitionen	13
Zur Rolle der Linguistik	14
Zur Verwendung von Fremdwörtern und Fachausdrücken	14
Zum Stil	14
Zum Layout	15
1 Verarbeitung natürlicher Sprache	16
1.1 Aufgabe	16
1.2 Probleme	18
1.2.1 Exkurs: Syntax, Semantik und Pragmatik	18
1.2.1.1 Syntax	19
1.2.1.2 Semantik	19
1.2.1.3 Pragmatik	20
1.2.2 Mehrdeutigkeit	20
1.2.2.1 Syntaktische Mehrdeutigkeit	21
1.2.2.2 Semantische Mehrdeutigkeit	22
1.2.2.3 Pragmatische Mehrdeutigkeit	22
1.2.3 Mächtigkeit von Sprache	23
1.2.4 Subjektivität und Kontext	23
1.3 Lösungsansätze	24
1.3.1 Füllschablonen	24
1.3.2 ad-hoc-Ansatz	25
1.3.3 Kontextfreie Ersetzungsregeln	25
1.3.4 Transformationsgrammatik	25
1.3.5 Erweiterte Übergangnetze	26
1.3.6 Unifikationssysteme	27
1.3.7 Fallschablonen	27
2 Deutsch	29
2.1 Wortarten	30
2.1.1 Verb	30
2.1.1.1 Arten von Verben	30
2.1.1.2 Präfixe	31
2.1.2 Substantiv	32
2.1.2.1 Substantivarten	32
2.1.2.2 Substantivbildung	33
2.1.3 Adjektiv	33

2.1.4	Adverb	34
2.1.5	Präposition	34
2.1.6	Konjunktion	35
2.1.7	Sonstige	36
2.2	Beugung	37
2.2.1	Verb	37
2.2.1.1	regelmäßige Verben	38
2.2.1.2	unregelmäßige Verben	39
2.2.2	Substantiv, Artikel und Pronomen	40
2.2.2.1	Singular	41
2.2.2.2	Plural	42
2.2.3	Adjektiv	44
2.2.3.1	Komparativ	45
2.2.3.2	Superlativ	45
2.2.4	Sonstige	46
2.3	Kongruenz	46
2.3.1	Subjekt—Prädikat	46
2.3.1.1	Nichtkongruenz der Person	46
2.3.1.2	Nichtkongruenz der Zahl	47
2.3.2	Nominalphrasen	48
2.3.3	Gleichsetzungssatz	48
2.3.4	Substantivisches Attribut (Apposition)	49
2.4	Prädikat	49
2.4.1	Zeitenbildung	49
2.4.2	Modale Färbung	50
2.4.3	modifizierende Verben	51
2.5	Satzteile	51
2.5.1	Subjekt	51
2.5.2	Objekte	52
2.5.2.1	Akkusativobjekt	52
2.5.2.2	Dativobjekt	52
2.5.2.3	Präpositionalobjekt	53
2.5.2.4	Genitivobjekt	53
2.5.2.5	Gleichsetzungsnominativ	53
2.5.3	Adverbial gebrauchte Satzglieder	53
2.5.3.1	Adverbiales Präpositionalgefüge	54
2.5.3.2	Adverbialgenitiv	55
2.5.3.3	Adverbialakkusativ	55
2.5.4	Zugeordnete Glieder	55
2.5.5	Satzadjektiv	56
2.5.6	Satzpartikel	56
2.6	Satz	57
2.6.1	Grundsätzliches	57
2.6.2	Nebensätze	58
2.6.3	Satzzeit	61
2.6.4	Satzmodus	61
2.6.5	Satzgenus	62
2.6.6	Aussagesatz	62
2.6.7	W-Frage	63

2.6.8	Ja/Nein-Frage	63
2.6.9	Befehlssatz	63
2.7	Besondere Themen	63
2.7.1	Ellipsen	63
2.7.2	Negation	64
2.7.3	Zahlen und Zahlwörter	65
3	Analyse mit Fallschablonen	67
3.1	Fallschablonen und Zerteiler	67
3.1.1	Was ist eine Fallschablone ?	67
3.1.2	Was ist der Zweck von Fallschablonen ?	68
3.1.3	Was ist ein Zerteiler ?	69
3.1.4	Was ist ein Fallschablonenzerteiler ?	69
3.2	Tolerante Zerteilung	69
3.3	Rollen	71
3.4	Chart-Parsing	73
4	Architektur	75
4.1	Generelles	75
4.2	Ein-/Ausgabekategorien	76
4.3	Die Wissensbasis	77
4.4	Das Wörterbuch	78
4.4.1	Grundgedanke	78
4.4.2	Vorüberlegungen und Entwurfsentscheidungen	79
4.4.3	Aufbau	80
4.5	Die Konzepthierarchie	85
4.5.1	Grundgedanke	85
4.5.2	Vorüberlegungen und Entwurfsentscheidungen	85
4.5.3	Aufbau	86
4.6	Das Rollenverzeichnis	87
4.6.1	Grundgedanke	87
4.6.2	Vorüberlegungen und Entwurfsentscheidungen	88
4.6.3	Aufbau	88
4.7	Die Fallschablonen	90
4.7.1	Vorüberlegungen und Entwurfsentscheidungen	90
4.7.2	Aufbau	91
4.8	Der Zerteiler	93
4.8.1	Leistungsumfang	93
4.8.2	Zerteilungsphasen	94
4.8.3	Zerteilungsalgorithmus	95
4.8.3.1	Generelles	95
4.8.3.2	Information in den Kanten	97
4.8.3.3	Initialisierung und Kantenauswahl	98
4.8.3.4	Instanzieren von Fallschablonen	100
4.8.3.5	Das Startpunktproblem	103
4.8.3.6	Erweitern einer Instanziierung	105
4.8.3.7	Abschluß und Auswertung einer Instanziierung	107
4.8.4	Zeitkomplexität	108
4.9	Die Ausgabe	109

5	Implementation	113
5.1	Allgemeines	113
5.2	Modularisierung (Top-Down)	113
5.2.1	Wissensbasis (gesamt)	114
5.2.2	Wörterbuch	115
5.2.3	Konzepthierarchie	116
5.2.4	Rollenverzeichnis	116
5.2.5	Fallschablonen	117
5.2.6	Chart	117
5.2.7	Zerteiler	118
5.3	Sonstige Module (Bottom-Up)	118
6	Test	120
6.1	Schwächen beim Test	120
6.2	Allgemeines Vorgehen	121
6.3	Big Bang Test \iff inkrementeller Test	121
6.4	Blackbox-Test \iff Whitebox-Test	121
6.5	Bottom-up Test \iff Top-down Test	122
6.6	Codeinspektionen und Walkthroughs	122
6.7	Fehlerlokalisierung	123
6.8	Automatisches Rückfalltesten	124
6.9	Konkretes Vorgehen	124
7	Bedienung	126
7.1	Aufruf und Kommandoeingabe	126
7.2	Beschreibung der Kommandos	127
7.3	Ausgabe des Chart	129
7.4	Ausgaben im EDGE-Format	129
7.4.1	Konzepthierarchie	129
7.4.2	Chart	129
8	Auswertung und Ausblick	131
8.1	Beobachtungen	131
8.1.1	Stärken	131
8.1.2	Schwächen	132
8.1.3	Messungen	132
8.2	Ausbau und Ausbaufähigkeit	133
A	Standard-Wissensbasis sara.std	135
B	Beispiel-Wissensbasis	144
	Literaturverzeichnis	145

Tabellen

2.1	Die einteiligen Formen regelmäßiger Verben	39
2.2	Die Beugungsformen von sein und haben	40
2.3	Die Beugungsformen von werden	40
2.4	Beugung des bestimmten und des unbestimmten Artikels	41
2.5	Beugung der Substantive in der Einzahl	42
2.6	Beugung der Substantive in der Mehrzahl (Pe,PUe,P,PU)	43
2.7	Beugung der Substantive in der Mehrzahl (Pn,Pen,Per,PUer)	43
2.8	Beugung der Substantive in der Mehrzahl (Ps,Pss,Pi,Pue)	44
2.9	Beugung der Adjektive (Typ I/II/III)	44

Abbildungen

4.1	Architektur von SARA	76
5.1	Modulstruktur von SARA	114
5.2	Verweisstruktur des Wörterbuches	115
5.3	Verweisstruktur der Konzepthierarchie	116
5.4	Verweisstruktur des Rollenverzeichnisses	116
5.5	Verweisstruktur der Fallschablonenmenge	117
5.6	Die Operationen des Moduls list	118

Vorbemerkungen

*Oh, laß es die Weisen doch verständlich sagen,
mir das Hirn nicht mit Erkenntnis plagen.*

Crabbe

Zum Aufbau dieser Arbeit

Diese Diplomarbeit besteht aus zwei Teilen, deren erster dies ist. Der zweite Teil enthält die Quellcodes des im Rahmen dieser Arbeit erstellten Programms, sowie einige organisatorische Erklärungen dazu. Der erste Teil ist wie folgt aufgebaut:

Das Kapitel 1 enthält eine kurze Einführung in das Gebiet der Verarbeitung natürlicher Sprache. Dazu gehört eine Formulierung der Problemstellung, eine Beschreibung der Schwierigkeiten und eine kurze Beschreibung der populären Arbeitsansätze.

Im Kapitel 2 wird der Objektbereich beschrieben, also die Sprache Deutsch. Dies ist ein Überblick über die Grammatik, der vom Rest der Arbeit weitgehend losgelöst ist.

Das Kapitel 3 führt in den hier verwendeten Arbeitsansatz ein: Fallschablonen und Chart-Parsing.

Die Kapitel 4 und 5 beschreiben den konkreten Aufbau des Systems. Dabei enthält das Kapitel 4 die externe Sicht (Architektur), einschließlich der Beschreibung der externen Datenformate für die Wissensbasis, während Kapitel 5 grob die interne Sicht (Modularisierung) darstellt. Weitere Beschreibung kann dem Teil II entnommen werden.

In Kapitel 6 wird die Teststrategie dargestellt, Kapitel 7 enthält das Benutzerhandbuch und Kapitel 8 eine Auswertung der Arbeit.

Zu den Definitionen

Die in dieser Arbeit enthaltenen Definitionen sind zumeist informal oder halbformal gehalten. In “minder schweren Fällen” der Begriffserklärung wurde keine vom Text abgesetzte Definition geschrieben, um die Arbeit nicht völlig damit zu überladen; in diesen Fällen ist ein definierendes, erklärendes oder beschreibendes Auftreten eines Begriffes im Text *in dieser Form* gesetzt.

Insbesondere wurden formale Definitionen von Begriffen unterlassen, wenn diese vermutlich allgemein bekannt sind, oder wo ein geringfügig anderes Verständnis eines Begriffes kein Fehlverständnis im Ganzen nach sich zieht.

Zur Rolle der Linguistik

“Die Linguistik ist eine empirische Wissenschaft. Linguistische Sprachforschung strebt kontrollierte und empirisch verifizierbare Beobachtungen an, die sich an einer allgemeinen Sprachtheorie orientieren.” [Bünting71]

Sie strebt, jawohl, und orientiert sich!¹

Zur allgemeinen Aufmunterung kursiert jedoch unter Linguisten immer noch der Spruch “that the only secure generalization on language that linguists are prepared to make is that ‘some members of some human communities have been observed to interact by means of vocal noises.’” [Fillmore68] und nur manche sind der Meinung, daß diese Ansicht überholt sei.

Kurz und klein: die meisten Ergebnisse der Linguistik sind zu unpräzise oder zweifelhaft um Anlaß zu der Hoffnung zu geben, darauf ein universelles System zur Verarbeitung natürlicher Sprache aufbauen zu können. Außerdem sind viele Resultate der Linguistik entweder informal oder zwar formal, jedoch nicht zur Berechnung mit Computern geeignet dargestellt. Zudem bewegt sich das heutige Geschehen im Bereich der Verarbeitung natürlicher Sprache auf einem eher pragmatischen Niveau.

Aus diesen Gründen sehe ich in dieser Arbeit davon ab, tiefer in sprachwissenschaftliche Darstellungen einzusteigen. Ergebnisse aus dem Bereich der Linguistik werden nur dort angeführt, wo sie unmittelbar nutzbar sind oder die Entwicklung maßgeblich beeinflusst haben.

Zur Verwendung von Fremdwörtern und Fachausdrücken

Im Titel dieser Arbeit kommt das Wort “Deutsch” vor und Deutsch ist ein wesentliches Thema darin. Ich nehme das zum Anlaß auch hauptsächlich die deutsche Sprache zu *verwenden*. Natürlich ist es im Bereich der Informatik fast unvermeidlich einige englische Fachausdrücke zu gebrauchen und auch ein gewisser Satz von lateinischen oder griechischen Fremdwörtern ist so fest im deutschen Sprachgebrauch (zumal dem akademischen) verankert, daß die Ersetzung durch deutsche Begriffe mehr stören als reinigen würde. Andererseits möchte ich mich doch so deutlich wie möglich vom Zunftjargon der Informatik und Linguistik abgrenzen und die Gelegenheit nutzen vielleicht einige nützliche deutsche Fachbegriffe zu prägen.

Zum Stil

Meiner Ansicht nach geht es in unserer Gesellschaft und dort besonders im wissenschaftlichen Leben etwas zu ernst zu.

Ich versuche deshalb bei jeder sich bietenden Gelegenheit einen humorvollen Stil zu schreiben und glaube, daß darunter die inhaltliche Präzision nicht notwendigerweise leidet. Informatik–Gurus wie Donald Knuth führen es vor.

¹Nichts gegen die Linguistik oder die sie betreiben. Gerade in dieser Hinsicht hat sie schließlich viel mit der Informatik, zumal der KI, gemein.

In diesem Sinne ist auch die Namensgebung des hier vorgestellten Systems zu verstehen. Es heißt **SARA**. Das bedeutet nichts Bestimmtes, insbesondere ist es keines der in der Informatik auch so beliebten Akronyme. Der Name hat jedoch zwei Vorteile, die für mich ausschlaggebend waren, ihn zu wählen: erstens gefällt er mir und zweitens läßt er sich vernünftig aussprechen.

Zum Layout

Dieses Dokument wurde mit L^AT_EX formatiert [Kopka88] und auf einem Apple Laserwriter ausgedruckt.

Ich habe das Schwergewicht bei der Formatierung nicht darauf gelegt, möglichst viele oder möglichst wenige Seiten zu füllen, sondern versucht bestmögliche Übersichtlichkeit zu erreichen; jeder, dem diese Diplomarbeit zu dünn oder zu dick ist, möge mir das nachsehen.

Zwei Zeichensätze werden, außer demjenigen für den laufenden Text, systematisch verwendet. So werden Beispiele aus der natürlichen Sprache als *hier ist ein Beispiel* geschrieben oder manchmal

so abgesetzt. Dabei sind Hervorhebungen *auf diese Weise* angedeutet.

Formulierungen, die aus einer Programmiersprache oder ähnlichem stammen, sind in Schreibmaschinenschrift gehalten.

Kapitel 1

Verarbeitung natürlicher Sprache

*Im Anfang war das Wort,
und das Wort war bei Gott,
und letzterer identifizierte sich mit ersterem.*

In diesem Kapitel soll allgemein auf das Gebiet der Verarbeitung natürlicher Sprache eingegangen werden.

Es werden zunächst unterschiedliche Auffassungen der Aufgabe von Verarbeitung natürlicher Sprache skizziert und die Definition dieser Aufgabe angegeben, die der vorliegenden Arbeit zugrundeliegt. Es folgt eine kurze Beschreibung der fundamentalen Probleme und der bisher benutzten Ansätze, diese zu lösen.

1.1 Aufgabe

Die Ansichten über die Aufgabe der Verarbeitung natürlicher Sprache sind in den damit befassten Gebieten der Wissenschaft, nämlich der Informatik und der Computerlinguistik, alles andere als einheitlich.

Eine mögliche Sichtweise liefert der *Turing-Test*: Ziel ist, den Computer in seinen Dialogfähigkeiten dahin zu bringen, daß ein Mensch ihn im Prinzip nicht mehr von einem menschlichen Kommunikationspartner unterscheiden kann. Dies ist also ein dialog- und resultatsorientierter Ansatz.

Eine ähnliche Richtung verfolgt die Sichtweise der *maschinellen Übersetzung*: Ziel ist, einen beliebigen Text aus einer (natürlichen) Sprache in eine andere sinngemäß korrekt zu übersetzen. Dies ist ein ebenfalls resultats- jedoch nicht dialogorientierter Ansatz.

Beide haben gemein, daß sie aus heutiger Sicht in ihrer Allgemeinheit nicht realisierbar sind und daher bislang nur von beschränktem praktischen Nutzen als Leitschnur wissenschaftlicher Arbeit sein können.

Andere Betrachtungen der Verarbeitung natürlicher Sprache orientieren sich mehr an technischen Gesichtspunkten. So definieren Hayes und Carbonell *natural language processing* als die Formulierung und Untersuchung effektiv berechenbarer Mechanismen zur Kommunikation durch natürliche Sprache [HayCar87].

Auch dieser Ansatz muß sich jedoch vorwerfen lassen, daß er bei weitem zu allgemein ist,

um als eine Handlungsrichtlinie für die Erarbeitung eines praktisch einsetzbaren Systems brauchbar zu sein.

Das Fernziel dieser Arbeit ist jedoch gerade ein praktisch einsetzbares System, daher benötigen wir hier eine Definition, deren Inhalt realisierbar und konkret ist.

Um beschreiben zu können, was unter Verarbeitung natürlicher Sprache im Sinne dieser Arbeit zu verstehen ist, benötigen wir zunächst einmal die folgenden zwei Begriffe:

Definition “Fachidiot”

Ein Fachidiot ist ein System, das auf einem engumgrenzten Gebiet ein weitgehend vollständiges und exaktes Wissen hat und in der Lage ist, dieses anzuwenden.

Das Wissen eines Fachidioten über die Welt außerhalb seines Fachgebietes kann beliebig gering sein.

Dies unterscheidet sich etwas von der umgangssprachlichen Verwendung des Wortes Fachidiot:

- bei dem Wort “System” ist natürlich hauptsächlich an Menschen oder digitale Computersysteme gedacht; im Prinzip kann aber auch ein andersgeartetes technisches System oder ein Tier ein Fachidiot sein.
- das erwähnte Allgemeinwissen kann sehr gering sein, das ist jedoch nicht notwendigerweise so. In jedem Fall muß das Spezialwissen wesentlich exakter und vollständiger sein als das Allgemeinwissen, um von einem Fachidioten sprechen zu können.

Als Zweites nun der davon unabhängige Begriff des *Sprechpartners*:

Definition “Sprechpartner”

Ein System S ist Sprechpartner für System A, wenn es Sinn und Zweck von Aussagen, Fragen und Anweisungen von A in natürlicher Sprache so erfasst, daß es im Rahmen seiner Aktionsmöglichkeiten darauf so reagieren kann, wie es von A gewünscht ist.

Beachte dabei

- von S wird keinerlei Initiative verlangt, die Relation Sprechpartner zu sein ist also nicht symmetrisch.
Auf menschliche Verhältnisse übertragen bedeutet das etwa die folgende Situation: zwei Personen mit derselben Muttersprache sind im Normalfall wechselseitig Sprechpartner. Trotz gleicher Sprache kann jedoch die Sprechpartnereigenschaft in einer oder beiden Richtungen ausfallen, wenn die beiden Personen stark unterschiedliche Weltansichten haben, zum Beispiel deshalb, weil sie aus unterschiedlichen Kulturkreisen kommen. Schließlich kann der Fall vorkommen, daß für eine der Personen X und Y, sagen wir X, die gemeinsam verwendete Sprache eine Fremdsprache ist, die nicht gut beherrscht wird, so daß zwar X ein Sprechpartner für Y ist, denn er kann verstehen und reagieren, jedoch nicht umgekehrt, weil X nicht in der Lage ist, die Initiative zu ergreifen.

- es ist keine unbedingte Handlungsfähigkeit gefordert, die Verstehensfähigkeit darf die Handlungsfähigkeit übersteigen. Das bedeutet, daß die Sprechpartnereigenschaft nicht verletzt ist, wenn ein S eine Äußerung von A zwar versteht, jedoch nicht in der gewünschten Weise reagieren kann. Dies bedeutet eine Trennung der geistigen von der physikalischen Welt.
- es ist auch keine unbedingte Handlungsbereitschaft gefordert. Eine Handlung darf auch unterbleiben, ohne daß die Sprechpartnereigenschaft dadurch verletzt wird, wenn die zugehörige Äußerung verstanden wurde und die Handlung durchführbar wäre. Dies ermöglicht einen freien Willen.

Nun läßt sich die Aufgabe von Verarbeitung natürlicher Sprache wie folgt definieren

Definition “System zur Verarbeitung natürlicher Sprache”

Ein System zur Verarbeitung natürlicher Sprache ist ein technisches System, das als Sprechpartner mit den Eigenschaften eines Fachidioten für einen menschlichen Benutzer bereitsteht.

Ziel dieser Arbeit ist es, ein System zur Verarbeitung der natürlichen Sprache Deutsch zu beschreiben und zu implementieren¹.

1.2 Probleme

Die Aufgabe der Verarbeitung natürlicher Sprache ist bisher noch nicht wirklich zufriedenstellend gelöst worden. Das rührt allerdings nicht allein daher, daß die Informatik noch in ihren Kinderschuhen steckt, sondern vor allem von einigen Schwierigkeiten, die dem Problem Sprache direkt innewohnen : Mächtigkeit, Subjektivität und Mehrdeutigkeit.

Diese Faktoren sollen im folgenden einer kurzen Analyse unterzogen werden. Ich konzentriere mich dabei hauptsächlich auf Deutsch. Die meisten Aussagen sind aber auf sehr viele andere Sprachen übertragbar.

1.2.1 Exkurs: Syntax, Semantik und Pragmatik

Die Verwendung von Sprache wie überhaupt jede Kommunikation vollzieht sich im allgemeinen auf drei Ebenen: einer Formebene (Syntax), einer Sinnebene (Semantik) und einer Zweckebene (Pragmatik). In der Verarbeitung natürlicher Sprache müssen im Prinzip alle drei Ebenen berücksichtigt werden, wobei jede auf der vorherigen aufbaut und bisher jede technisch deutlich schlechter beherrscht wird als die darunter.

In der Informatik wird in den meisten Fällen jeder dieser drei Begriffe in mehreren Bedeutungen vermischt verwendet. Diese Bedeutungen sind sinnverwandt und werden stets aus dem Zusammenhang richtig erkannt; daher wird diese Praxis hier beibehalten. In den folgenden Absätzen werden jeweils alle diese Bedeutungsmöglichkeiten beschrieben. Es handelt sich dabei im Wesentlichen immer um die zwei Varianten der Betrachtung von Syntax, Semantik oder Pragmatik als eine Lehre oder als das Ergebnis einer Analyse einer einzelnen Äußerung.

¹Dieses System enthält allerdings nur rudimentäre Aktionsmöglichkeiten, so daß die Handlungsfähigkeit fast gleich Null ist. Der Schwerpunkt liegt auf den ersten Phrasen der Analyse.

1.2.1.1 Syntax

Die *Syntax (Form)* kann als System von Regeln, verstanden werden, die man in ihrer Gesamtheit Grammatik nennt. Die Syntax wäre danach eine Lehre. Diese Regeln legen fest wie aus den elementaren Symbolen der Sprache (Buchstaben oder Lauten) über mehrere Stufen hinweg Äußerungen zusammengesetzt werden. Diese Zwischenstufen sind zum Beispiel Wort, Wortgruppe, Satzteil, Teilsatz und Satz.

Eine nur subtil andere Betrachtung von Syntax ist die, daß die Syntax nicht eine Lehre ist, welche die Regeln vorgibt, sondern ein abstraktes Objekt, das durch eben diese Regeln beschrieben wird.

Als Syntax kann man jedoch auch das Ergebnis der formalen Analyse einer Äußerung aufgrund dieser Regeln bezeichnen. Syntax erhält dann einen Objektbezug ("Die Syntax dieser Aussage") und ist ihrerseits jeweils eine Menge von Aussagen, die eine Struktur beschreiben. Diese Beschreibung kann korrekterweise mehrdeutig sein, da natürliche Sprachen tatsächliche, syntaktisch unauflösbare Mehrdeutigkeiten enthalten.

Die Syntax wird in der Linguistik sehr gut beherrscht [Duden(4)].

Die Beherrschung des Phänomens Syntax im Rahmen der Verarbeitung natürlicher Sprache ist mittelmäßig: Für manche Sprachen, vor allem solche mit verhältnismäßig fester Wortstellung wie das Englische, lassen sich die meisten relevanten Konstruktionen zufriedenstellend syntaktisch analysieren. Allerdings liegt der bisher realisierte Deckungsgrad auch dort noch unter 100 Prozent. Für Sprachen, die aufgrund freierer Wortstellung eine größere Variationsbreite gängiger Ausdrucksformen aufweisen, wie zum Beispiel dem Deutschen, ist die bisher erreichte syntaktische Abdeckung wesentlich geringer. Dies ist allerdings in beiden Fällen nicht als ein grundsätzliches Problem aufzufassen, vielmehr muß man die rein syntaktische Analyse als ein im Prinzip gelöstes Problem ansehen. Allerdings ist der Aufwand für eine vollständige Beherrschung wegen der großen Komplexität und Unregelmäßigkeit natürlicher Sprachen gewaltig. Im Deutschen sind die Erfolge bisher vor allem deshalb recht gering, weil erstens (im Vergleich zu Englisch) keine allzu großen Anstrengungen unternommen wurden und zweitens der Aufwand wegen der Kompliziertheit der deutschen Grammatik besonders groß ist.

1.2.1.2 Semantik

Bei der *Semantik* ergibt sich eine ähnliche Unterscheidung wie bei der Syntax: Man kann sie als die Lehre von der Bedeutung von Äußerungen auffassen oder als die Bedeutung einer einzelnen Äußerung oder deren strukturelle Beschreibung².

Betrachtet man die Einzelfälle, so kann man noch sinnvoll unterscheiden zwischen der objektiven Bedeutung, der vom Sprechenden beabsichtigten und der vom Empfänger der Äußerung wahrgenommenen, wobei alle drei mehrdeutig sein können. In allen Fällen bildet die syntaktische Zerlegung der Äußerung zusammen mit einer Art Bedeutungswörterbuch die Grundlage jeder semantischen Analyse.

In der Linguistik ist die Semantik ein reiches Forschungsgebiet in dem zwar eine ganze Reihe von Ergebnissen vorliegen, die sich aber meist einer Formalisierung im Sinne der

²Was genau die Bedeutung von "Bedeutung" ist, soll hier nicht weiter erörtert werden, da uns das geradewegs in die Philosophie führen würde. Das intuitive Verständnis des Begriffes ist an dieser Stelle ausreichend.

Informatik, also der Darstellung in einer berechenbaren Form, entziehen. Auch ist der Abdeckungsgrad sprachlicher Ausdrucksmöglichkeiten bisher gering.

Auch in der Informatik stellt die semantische Analyse bei der Verarbeitung natürlicher Sprache ein ungelöstes Problem dar. Es gibt zwar eine Reihe von theoretischen Ansätzen, verwertbare Ergebnisse beschränken sich bisher aber meist darauf in winzigen, klar abgegrenzten Problembereichen eine semantische Analyse zu ermöglichen, indem man im Wesentlichen alle Bedeutungen zu erwartender Eingaben (klassenweise) vorhersieht, im Wissensvorrat des Systems verankert und somit nur nach vorbestimmten Bedeutungen sucht.

1.2.1.3 Pragmatik

Die *Pragmatik* ist, als Lehre betrachtet, die Lehre vom Zweck sprachlicher Äußerungen. Es geht also um die allgemeine Beantwortung der Fragen “Wodurch wird ein Sprecher zu einer Äußerung motiviert?” und “Was will ein Sprecher mit einer Äußerung erreichen?”. Die Beantwortung dieser Fragen für einen Einzelfall stellt offenbar die andere Betrachtungsweise dar, analog zur Unterscheidung bei Syntax und Semantik. Es bietet sich dann noch eine dritte Frage an, die man sinnvoll mit in die Untersuchung von Pragmatik einbeziehen kann, und zwar “Was erreicht der Sprecher mit der Äußerung tatsächlich?”.

In der Linguistik befinden sich die Studien der Pragmatik auf einem ähnlichen Niveau wie die der Semantik.

In der Informatik sieht die Lage dagegen schlecht aus: in den heute existierenden Systemen zur Verarbeitung natürlicher Sprache wird die Verarbeitung meist auf der semantischen Ebene beendet. Wenn überhaupt Aspekte der Pragmatik berücksichtigt werden, die sich nicht in der Umsetzung semantischer Erkenntnisse erschöpfen, dann handelt es sich in der Regel um statische Vorgaben der Art “Wenn Dich jemand fragt, ob Du weißt, wie spät es ist, dann antworte nicht mit JA, sondern antworte mit der Uhrzeit”.

Glücklicherweise ist die Behandlung der Pragmatik in Computeranwendungen jedoch bei weitem nicht von der gleichen Bedeutung wie im zwischenmenschlichen Umgang, so daß diese simplen Maßnahmen schon recht gute Erfolge zeigen. Der Grund dafür ist einerseits psychologischer Art, denn Menschen gehen an einen Computer nicht mit der selben Kommunikationshaltung heran, wie an einen anderen Menschen, und andererseits sachlicher Art, denn die Dialoge, die mit dem Computer geführt werden sollen, berühren seltener solche Gebiete, auf denen pragmatische Aspekte, die über die unmittelbare Anwendung des semantischen hinausgehen, von Bedeutung sind.

1.2.2 Mehrdeutigkeit

Betrachte folgendes Beispiel

Susi fährt Motorrad. Sie überholt gerade ihre Maschine. Sie hat nämlich zu lange in der Garage gestanden.

Drei einfache Sätze, deren Sinn völlig klar ist und sehr einfach erkennbar zu sein scheint. Das einzige Problem könnte sein, herauszufinden, wer oder was mit dem zweiten “Sie” gemeint ist: “Susi” oder “ihre Maschine”, syntaktisch ist nämlich beides möglich. Nun

für den Menschen ist das gar kein Problem, es ist völlig klar die Maschine gemeint. Aber woher weiß man das ?

Das ist jedoch tatsächlich gar nicht das einzige Problem. Wie kompliziert die Lage in Wirklichkeit ist, wird klarer, wenn man noch das folgende Beispiel dazunimmt.

Susi fährt Motorrad. Sie überholt gerade Martins Maschine. Sie hat nämlich zu lange in der Garage gestanden.

Jetzt wird eine zweite, völlig veränderte Interpretation möglich, obwohl die Konstruktion der Sätze gleich ist und nur ein einziges Wort ausgetauscht wurde.

Das Problem basiert hier offensichtlich auf dem Wort “überholen”. Es wird nämlich einmal im Sinne von “pflegen, reparieren” benutzt und einmal als “vorbeifahren”. Die Auswirkungen sind jedoch noch größer und reichen sogar rückwärts! Denn durch die Sinnänderung beim zweiten Satz ändert sich auch der Sinn des Wortes “fährt” von einer allgemeinen Aussage, die eine Fähigkeit oder Gewohnheit angibt, in eine augenblickliche Tätigkeit, das Wort “nämlich” ändert seine Bedeutung von einer Begründung der Notwendigkeit zu einer Begründung der Möglichkeit, im ersten Beispiel steht das Motorrad (eventuell) immer noch in der Garage, im zweiten jedoch nicht. . .

Dieses Beispiel gibt einen ungefähren Blick darauf, wieviel Wissen nötig ist, um selbst so einfache Sätze korrekt verstehen zu können und wie stark die Wechselwirkungen zwischen Aussagen sind.

Es sollen nun die wichtigsten Kategorien solcher Mehrdeutigkeiten zusammengestellt werden, um einen ungefähren Überblick darüber zu schaffen, mit welchen Einzelphänomenen man sich herumzuschlagen hat.

1.2.2.1 Syntaktische Mehrdeutigkeit

Eine *syntaktische Mehrdeutigkeit* wird meist von fehlenden Beugungsformen verursacht. Die fehlende Information macht es möglich, den Bezug eines Wortes oder einer Wortgruppe verschieden auszulegen. Sie tritt auf in den Ausprägungen von Rollenmehrdeutigkeit oder Verweismehrdeutigkeit.

Eine Falle witterte das Mädchen.

Wer wittert hier wen ?

Die Rolle von Subjekt und Objekt wird hier durch das Übereinanderfallen der Beugungsformen im Nominativ und im Akkusativ mehrdeutig. Im Deutschen gibt es zumeist eine Vorzugsinterpretation durch die Wortstellung, die aber (wie in diesem Fall) nicht immer gültig sein muß. Dann hilft nur noch Weltwissen und die Auswertung des Zusammenhangs.

Eine Falle witterte das Mädchen. Sie tappte jedoch nicht in sie hinein.

Wer ist “Sie” und wer ist “sie” ?

Hier kann jedes der beiden “sie” sowohl auf das Mädchen als auch auf die Falle verweisen, es ergeben sich 4 Bedeutungsmöglichkeiten für den zweiten Satz, unabhängig von der Bedeutung des ersten. Sprachkonventionen scheiden davon zwei aus: die beiden Pronomen müssen sich zumindest auf zwei verschiedene Objekte beziehen. Weltwissen verschließt die dritte, denn Fallen tappen nicht. Also bezieht sich das erste “sie” als Subjekt auf das Mädchen und das zweite auf die Falle.

1.2.2.2 Semantische Mehrdeutigkeit

Die schlimmste Quelle von Mehrdeutigkeit sind mehrere Bedeutungsmöglichkeiten für einzelne Wörter. Beispiele dafür gibt es in rauen Mengen, etwa “überholen” oben. Eine solche Wortmehrdeutigkeit kann in ihrer Wirkung auf das eine Wort beschränkt sein, oder eine Satzmehrdeutigkeit induzieren, also eine strukturell neue Möglichkeit für die Interpretation des gesamten Satzes eröffnen.

Die Auflösung solcher Mehrdeutigkeiten ist nur mit einer großen Menge von Zusatzwissen möglich, etwa dem Wissen um die Tatsache, daß es eben *nicht* zum üblichen Verhaltensrepertoire von Fallen gehört, zu tappen. In der ungeheuren Größe dieser Anforderung nach Umfang und Komplexität liegt der Hauptgrund dafür, daß man sich bisher bei Systemen zur Verarbeitung natürlicher Sprache stets auf sehr sehr kleine Objektbereiche beschränkt.

Es ist bisher überhaupt nicht klar, wie groß die Menge des Wissens (in bit) ist, das man zum Erreichen eines allgemeinen Verständnisses benötigt und wie man dieses Wissen in einem Digitalrechner darstellen kann. In den heutigen Systemen mogelt man sich im Allgemeinen um die Lösung des Problems dadurch herum, daß man versucht alle Möglichkeiten zu solcher Mehrdeutigkeit vorherzusehen und im vorhinein aufzulösen; beispielsweise indem man Bedeutungen von Worten oder Wortkombinationen unterschiedlich gewichtet.

1.2.2.3 Pragmatische Mehrdeutigkeit

Die letzte Quelle von Mehrdeutigkeit rührt von der Tatsache her, daß der *Zweck* einer Äußerung eigentlich nur im Falle von Anweisungen stets klar ist: man möchte, daß die Anweisung ausgeführt wird. Im Falle von Aussagen oder Fragen ist aber unterschiedliche Auslegung möglich.

Wissen Sie, wie spät es ist ?

Will dieser Mensch nun wirklich nur wissen, ob ich es weiß ? Oder möchte er oder sie nicht vielmehr mein Wissen auch gleich anzapfen und wünscht sich, daß ich die Zeit sage, statt zu proklamieren, daß sie mir bekannt sei ?

Ziemlich schlechte Luft hier drin.

Aha.

Und nun ? Ich soll bitte das Fenster öffnen. Das hat zwar niemand gesagt, aber es ist für einen Menschen leicht herauszuhören. Bei diesem Beispiel spielt auch der durch die gemeinsamen Sinneswahrnehmungen der beteiligten Personen gebildete Kontext eine Rolle: in Situationen, in denen man kein Fenster öffnen kann oder will (vielleicht in einem Operationssaal), entfällt der Aufforderungscharakter dieses Satzes wieder.

Diese Art der Sprachverwendung ist von recht großer Bedeutung im täglichen Umgang von Menschen untereinander. Glücklicherweise ist der Grund dafür hauptsächlich Höflichkeit: da Menschen gegenüber Computern sowieso nicht höflich sind, ist die Bedeutung der pragmatischen Mehrdeutigkeit für Systeme zur Verarbeitung natürlicher Sprache recht gering und damit die beinahe perfekte Unfähigkeit existierender Systeme, damit umzugehen, nicht allzu beklagenswert.

1.2.3 Mächtigkeit von Sprache

Ein fundamentales Problem der Verarbeitung natürlicher Sprache kann man in der übergroßen Mächtigkeit natürlicher Sprachen sehen. Bereits weit vor der Frage des Weltwissens taucht das Phänomen auf, daß man Äußerungen nicht analysieren kann, weil ihre syntaktische Vielfalt zu groß ist. Dies bezieht sich zum einen auf die enorme Zahl von Worten, die zum Standardwortschatz natürlicher Sprachen gehören, zum andern aber auch auf die große Zahl gleichwertiger Satzkonstruktionen für dieselbe Aussage. Hier findet sich ein weiterer Grund für die strenge Bereichsbeschränkung bei existierenden Systemen.

Ein anderer Aspekt der Mächtigkeit von Sprache ist die Tatsache, daß es möglich ist "Unerhörtes zu sagen". Es ist prinzipiell ausgeschlossen, auf der Ebene der Semantik oder gar der Pragmatik alles vorauszusehen, was auftreten kann. Selbst auf syntaktischer Ebene sind fast beliebige neue Konstruktionen für einen Menschen noch verständlich ! Das heißt das vollständige Verstehen sprachlicher Äußerungen kann nicht simuliert werden, sondern ist nur auf dem Weg über ein komplettes Weltwissen zu erreichen. Allerdings ist dies ein Punkt, der bei den meisten Anwendungen überhaupt nicht zum Tragen kommt.

1.2.4 Subjektivität und Kontext

Als letztes großes Hindernis für die erfolgreiche Verarbeitung natürlicher Sprache ist die Tatsache zu erwähnen, daß man Äußerungen normalerweise nicht isoliert analysieren kann: sie müssen einerseits eingebettet in eine Kommunikationssituation (Dialogkontext) und andererseits im Lichte zusätzlichen (sicheren oder vermuteten) Wissens über den Kommunikationspartner betrachtet werden.

So kann also dieselbe Äußerung von derselben Person in zwei verschiedenen Situationen oder Gesprächen zwei verschiedene Bedeutungen haben und ebenso kann dieselbe Äußerung in derselben Situation verschiedene Bedeutungen haben, je nachdem, von wem sie kommt.

Betrachten wir als Beispiel den einfachen Satz

Wo ist der blöde Meier ?

Dieser Satz liefert völlig unterschiedliche implizite Information, je nachdem, was wir über die Person wissen, die ihn ausspricht und über den Kontext, in dem er ausgesprochen wird: Anscheinend wird hier ein Herr Meier gesucht, es könnte sich aber auch um einen Kater handeln. Nehmen wir an, das sei nicht der Fall. Weiß man vom Sprecher, daß er oder sie das Wort "blöd" als ein Lieblingswort gebraucht, dann enthält der Satz keinerlei Beleidigung und keine zusätzliche Information. Wissen wir vom Sprecher, daß er wütend ist (per Beobachtung festzustellen) oder daß es sich um einen sehr förmlichen und höflichen Menschen handelt, so können wir schließen, daß Meier anscheinend großen Mist gemacht hat. Andernfalls liegt offenbar eine allgemeine Antipathie des Sprechers gegen ihn vor.

Die Subjektivität hat beim Umgang mit dem Computer keine Bedeutung, weil die Maschine keine Persönlichkeit hat und der Mensch daraufhin seine eigene in den Hintergrund stellt. Daß jemand einem Computer die Anfrage eingibt **Können Sie mir bitte sagen, wie**

spät es ist?, ist schwer vorstellbar; zumindest, solange die Eingabe per Tastatur erfolgt. Dagegen ist die Herstellung und Fortschreibung eines Dialogkontexts eines der großen Probleme bei der Verarbeitung natürlicher Sprache. So muß als Hauptschwierigkeit die Handhabung von Verweisen über mehrere Äußerungen hinweg und im Zusammenhang damit die Ergänzung von Ellipsen gelöst werden.

1.3 Lösungsansätze

Im Folgenden seien kurz die wichtigsten bisher realisierten Ansätze im Bereich der Verarbeitung natürlicher Sprache angeführt. Diese Auflistung beschränkt sich auf Lösungsansätze für die Probleme, die auch von SARA in Angriff genommen werden, also die Analyse von Syntax und einigen Teilen der Semantik.

Den Anfang machten sehr geradlinig erfundene Systeme, die auf einfacher Zeichenketten-suche basierten. Entscheidende Impulse gaben ab den 50er Jahren die Arbeiten Chomskys, der die Theorie formaler Sprachen einführte und salonfähig machte, aus der nicht nur fast sämtliches systematische Wissen über unsere heute gebräuchlichen (kontextfreien) Computersprachen und ihre Handhabung hervorging, sondern auch die wichtigsten formalen Ansätze in der Linguistik: die (verallgemeinerte) Phrasenstrukturgrammatik und die Transformationsgrammatik. Diese in der Linguistik verwendeten Beschreibungsmittel erwiesen sich auch als verwertbar zur Implementation von Systemen zur Verarbeitung natürlicher Sprache. Heute muß man bei der Beschreibung der Systemansätze trennen zwischen dem Grammatikformalismus, also der Art, wie das Wissen über das Sachgebiet und manchmal auch über die Syntax abgebildet wird und dem Zerteilungsverfahren, also dem Algorithmus, der das syntaktische Wissen enthält (eventuell auch nur anwendet) und das Sachgebietswissen benutzt.

Im Einzelnen:

1.3.1 Füllschablonen

Bei frühen Systemen wurde meist der Ansatz verwendet, eine Reihe von Satzschablonen vorzusehen, in denen nur eine Reihe von Leerstellen für einzelne Wörter freigelassen war. Es mußten also alle Formen denkbarer Eingabesätze exakt vorausgesehen werden.

Diese Systeme arbeiteten also ohne jegliche semantische Analyse, sondern auf rein syntaktischer Ebene; die Semantik wurde vollkommen statisch den einzelnen Schablonen zugeschrieben. Füllschablonen sind als Grammatikformalismus zu verstehen, mit einem weitgehend trivialen Zerteilungsverfahren.

Beispiele für solche Systeme sind das Abfragesystem BASEBALL von B.F. Green und natürlich die allseits bekannte ELIZA [Weizenb66].

Dieser Ansatz eignet sich prinzipiell nur für Sprachen mit relativ fester Wortstellung wie das Englische und außerdem nur für *sehr* eng eingegrenzte Themenbereiche oder Systeme, die (wie ELIZA) den größten Teil der Eingabe überhaupt nicht wirklich verarbeiten.

1.3.2 ad-hoc-Ansatz

Immer wieder wurden und werden Systeme gebaut, die sich auf keinen theoretischen oder formalen Unterbau stützen. Man versucht stattdessen, mit beliebigen, dem zu lösenden Problem angepasst erscheinenden Methoden, die Analyse zu betreiben. Meist wird dabei zum Beispiel einzelnen Worten prozedurales Wissen zugeordnet.

Dieser Ansatz ist naturgemäß sehr flexibel, scheitert jedoch gerade durch seine Beliebigkeit, denn ein solches System ist nur äußerst schwierig in irgendeiner Weise zu ändern oder zu erweitern und eine Übertragung auf einen anderen Objektbereich ist geradezu unmöglich. Das liegt auch daran, daß sich ohne eine geordnete Darstellung des syntaktischen und semantischen Wissens die Komplexität eines Systems zur Verarbeitung natürlicher Sprache nur auf einem niedrigen Niveau beherrschen läßt.

1.3.3 Kontextfreie Ersetzungsregeln

Die frühen Versuche zur Formalisierung natürlicher Sprachen in der Linguistik stützten sich zum großen Teil auf die sogenannte “simple phrase structure grammar”, kontextfreie Produktionensysteme, als Grammatikformalismus. Auch in der Informatik wurde dieser Ansatz aufgegriffen und während er im Bereich der künstlichen Sprache große Fortschritte und ein festes theoretisches Strukturgebäude hervorbrachte mit zahlreichen effizienten Zerteilungsverfahren für (künstliche) Sprachen mit gewissen Eigenschaften, waren die Erfolge bei der Verarbeitung natürlicher Sprache gering. Dafür gibt es verschiedene Gründe:

- Um der Vielfältigkeit natürlicher Sprachen Herr zu werden würde man gigantische Produktionensysteme benötigen.
- Solche Grammatiken lassen auch nicht die geringsten Abweichungen von der vorgeschriebenen Form der Eingaben zu.
- Kontextfreie Regeln sind für manche Konstruktionen natürlicher Sprachen vermutlich nicht mächtig genug³.
- Die Form der Beschreibung ist unnatürlich und daher schwer in einem größeren Umfang zu erarbeiten oder zu verändern.

Ein Beispiel für ein solches System ist SAD–SAM [Lindsay63], das Verwandtschaftsrelationen analysiert.

Die resultierenden Systeme waren also relativ schwach und empfindlich, weshalb die Arbeiten mit diesem Ansatz auch nicht sehr viel weiter verfolgt wurden.

1.3.4 Transformationsgrammatik

Der Ursprung der Transformationsgrammatiken liegt in Chomskys “generative transformational grammar”. Unter diesem Namen ist die Theorie von Chomsky selbst und von anderen vielfach überarbeitet worden. Die folgende Darstellung ist also nur als ein ungefährer

³Über diese Frage gibt es einen jahrzehntelangen Streit unter den Linguisten

Fingerzeig auf den Aufbau der Transformationsgrammatik zu verstehen. Die Motivation der Transformationsgrammatik ist die Beobachtung, daß Äußerungen mit gleichem Sinn recht unterschiedliche Formen (sogenannte *Oberflächenformen*) in der tatsächlichen sprachlichen Formulierung annehmen können.

Man teilt also die Erzeugung⁴ einer sprachlichen Äußerung in mehrere Phasen ein. So enthält die syntaktische Komponente⁵ einen Basis- und einen Transformationsteil. Im Basisteil wird die sogenannte Tiefenstruktur einer Äußerung erzeugt, die dann im Transformationsteil in die endgültige Form, die sogenannte Oberflächenstruktur, umgewandelt wird.

Diese Theorie wird aber, das klang schon an, vor allem in der Linguistik verwendet. Ihre Bedeutung im Bereich der Verarbeitung natürlicher Sprache ist eher gering. Dies rührt vor allem daher, daß zu diesem Grammatikformalismus keine effizienten Zerteilungsverfahren gefunden wurden.

1.3.5 Erweiterte Übergangnetze

Ein erweitertes Übergangnetz (augmented transition net, ATN) basiert auf zwei Erweiterungen von endlichen Automaten. Man stelle sich zunächst ein Übergangdiagramm eines endlichen Automaten vor, an dem die Kanten mit Wörtern oder Wortarten markiert sind und den Knoten Semantik zugeordnet wird. Ein solcher endlicher Automat kann reguläre Sprachen erkennen.

Erweiterung 1:

Die Kanten dürfen nun zusätzlich mit den Namen von untergeordneten Übergangdiagrammen (diese repräsentieren höhere syntaktische Kategorien) gekennzeichnet sein. Die Benutzung einer solchen Kante bewirkt den Übergang in den Eingangsknoten des untergeordneten Übergangdiagramms, wobei die Position im übergeordneten gespeichert wird und die Verarbeitung nach Abschluß des untergeordneten an dieser Stelle wieder aufsetzt. Dieser Prozeß kann verschachtelt wiederholt werden, weshalb die Diagramme (oder Maschinen) mit dieser Erweiterung "recursive transition nets" (RTNs) genannt werden. RTNs haben die Mächtigkeit von Kellerautomaten, können also kontextfreie Sprachen erkennen.

Erweiterung 2:

Um aus einem RTN ein ATN zu machen, werden folgende drei Elemente hinzugefügt:

- frei verfügbarer Speicher (die sogenannten *Register*),
- Tests an Kanten, deren Erfüllung Voraussetzung dafür ist, daß die Kante benutzt wird und
- Aktionen an Kanten, die bei Benutzung der Kante durchgeführt werden und den Speicherinhalt verändern.

⁴Als eine linguistische Theorie beschreibt die generative Transformationsgrammatik nicht die Analyse sprachlicher Äußerungen, sondern deren Erzeugung. Das Rahmenwerk ist aber auf die Analyse übertragbar.

⁵Das System der generativen Transformationsgrammatik, wie es in der Linguistik verwendet wird, ist sehr umfangreich. Es enthält eine phonologische, eine morphologische, eine syntaktische und eine semantische Komponente.

ATNs haben damit die Mächtigkeit von Turing-Maschinen. Sie vereinigen Grammatik und Zerteilungsverfahren.

Ein Beispiel für ein ATN-basiertes System ist LUNAR [Woods73], das chemische Analysedaten über das von der Apollo 11 Mission mitgebrachte Mondgestein verarbeiten hilft, indem es die Eingaben in eine formale Abfragesprache übersetzt.

Einige Zeit galt der ATN-Ansatz als eine sehr vielversprechende Technik auf dem Gebiet. Es hat sich jedoch gezeigt, daß gerade die große Mächtigkeit von ATNs zu einer unsystematischen Verarbeitung (ver)führt und deshalb diesem Ansatz ähnliche Nachteile wie bei einer ad-hoc Methode verschafft. Ein zusätzlicher Nachteil, der auf der massiven Syntaxorientierung beruht, ist die geringe Robustheit gegenüber grammatikalisch falschen (aber sinnvollen) Eingaben.

1.3.6 Unifikationssysteme

Ein in der letzten Zeit in der Linguistik immer stärker diskutierter Ansatz ist der der Unifikationsgrammatiken. Ein Vertreter dieser Richtung ist beispielsweise die lexikalisch funktionale Grammatik.

Unifikation heißt die Rückführung aller vorkommenden Terme auf gewisse Grundformen (bei gleichzeitiger Ansammlung von Information) und Weiterverarbeitung nur auf Basis dieser Grundformen. Die Unifikation hat Eigenschaften, wie Idempotenz und Monotonie, die sie einer algebraischen Beschreibung zugänglich machen.

Unifikations-basierte Systeme können, wenn der Prozeß nicht zu eng an der Oberflächenform orientiert ist, robust sein und weisen, zumindest prinzipiell, die Fähigkeit zur erweiterbaren (inkrementellen) Analyse auf.

Ein solches System ist GuLP [Görz88]. Dabei handelt es sich um ein Meta-System, das auf einer Chart-Analyse als Rahmen des Zerteilungsverfahrens basiert und durch Parameteränderungen die unterschiedlichsten konkreten Zerteilungsverfahren verwenden kann. Es ist zur Erkennung gesprochener Sprache konzipiert.

1.3.7 Fallschablonen

Der Ausdruck Fallschablonen (case frames) stammt von [Fillmore68].

Der Grundgedanke dabei ist, den aus der Grammatik bekannten syntaktischen Begriff des Falles (Kasus) auf "semantisch relevante syntaktische Beziehungen" auszudehnen. Fillmore spricht dabei von Oberflächen- und Tiefenkasus. Dabei ist der Oberflächenkasus die übliche, meist Kasus genannte, grammatikalische Kategorie, während der Tiefenkasus eine semantische Kategorie ist, nämlich die Beschreibung einer semantischen Relation. Er schlägt vor das Verb als die zentrale Komponente des Satzes aufzufassen, um die herum mehrere Substantivphrasen angeordnet sind, von denen jede zu dem Verb eben in einer bestimmten semantischen Beziehung steht (Tiefenkasus), diese Beziehung nennt er eine Rolle. Er bietet in seinem ursprünglichen Aufsatz über dieses Thema dabei die folgenden Rollen an: Agent, Counter-Agent, Object, Result, Instrument, Source, Goal, Experiencer. Man bildet dann zu jedem Verb eine Fallschablone, die angibt welche Rollen mit welcher Priorität bei diesem Verb vorkommen. Dies definiert eine Menge von semantischen Einschränkungen für die Interpretation von Wörtern. Fallschablonen sind also ein Grammatikformalismus.

Es sind später von Fillmore selbst und von anderen Autoren unterschiedliche Sammlungen von Rollen angegeben worden, als ein Beispiel siehe [Winston81]. Manchmal wird die Theorie auch dahingehend verändert, daß man den Standpunkt verläßt, eine kleine und feste Menge von Rollen haben zu müssen und statt dessen Rollen auffasst, als eine irgendwie geartete und im Analysezusammenhang wichtige Relation bei einem Ereignis. Eine “konservative” Anwendung ist auch möglich: man betrachte die Fälle nicht als semantische Kategorien, sondern als syntaktische, die aber in ihrer Komplexität oberhalb der schieren grammatikalischen Fälle liegen. Schließlich steht die gesamte Bandbreite von Mischformen zur Verfügung. Für die konkretere Beschreibung des bei **SARA** verwendeten Fallschablonenbegriffs, siehe die Kapitel 3 und 4.

Durch diese Auslegungsbreite des Rollen- oder Fallbegriffs kann man von einer noch weitgehend syntaktisch orientierten Vorgehensweise, über eine semantisch orientierte aber noch konkret wortabhängige, bis hin zu einer auf abstrakten Konzepten aufgebauten, also rein semantisch basierten Analyse gelangen. Dabei kann und muß, je mehr man sich von der syntaktischen Repräsentation entfernt und sich der abstrakten nähert, immer mehr syntaktisches Wissen in den Zerteiler integriert werden, der diese Fallschablonen auswertet.

Man erhält dadurch eine klare Unterteilung eines Fallschablonen-basierten Systems:

1. Die Menge der Fallschablonen, die semantisches Wissen über den zu behandelnden Bereich enthält.
2. Der Zerteilalgorithmus selbst, der das allgemeingültige syntaktische Wissen enthält und die Auswertung der Fallschablonen steuert.

Bindeglied zwischen beiden Teilen ist ein Wörterbuch.

Beispiele für Systeme, die mit Fallschablonen arbeiten, sind NLH/E [TiAdHo88], ein adaptierbares Hilfesystem (zum Beispiel für CommonLisp) und CFPE [Bodsb89], das eine Erweiterung von NLH/E darstellt.

Die Vorzüge von Fallschablonensystemen sind:

- Die Erweiterung des semantischen Wissens wird durch dessen saubere Abgrenzung und seine Realisierung als Datenstruktur einfach.
- Solche Systeme sind somit auch sehr leicht übertragbar auf ein anderes Sachgebiet der Anwendung.
- Sie können robust gestaltet werden, das heißt unempfindlich gegenüber grammatikalisch falschen Eingaben, indem man eine tolerante Anwendung der Regeln im Zerteilungsalgorithmus vorsieht.
- Die syntaktischen und semantischen Fähigkeiten können unabhängig voneinander verbessert werden.

Kapitel 2

Deutsch

*Es gibt Dinge, die man fünfzig Jahre weiß,
und im einundfünfzigsten ist man erstaunt
über die Schwere und Furchtbarkeit ihres Inhalts.*

Adalbert Stifter

In diesem Kapitel sollen kurz die Kernbegriffe der deutschen Grammatik dargelegt werden, mit einigen Hinweisen auf die Problempunkte bei der Gestaltung eines Fallschablonensystems für Deutsch.

Es gibt eine kaum überschaubare Anzahl von strukturell unterschiedlichen Darstellungen der deutschen Grammatik. Ich werde mich nicht einer dieser Darstellungen vollständig anschließen, sondern mich lediglich an einer davon in der Hauptsache orientieren [Duden(4)], und immer dann abweichen und eine eigene Systematik einführen, wenn das im Hinblick auf die Aufgabenstellung der Analyse mit Fallschablonen eine Erleichterung verspricht.

Die Sichtweise bei der Beschreibung ist nicht so sehr eine streng formale, grammatikorientierte, sondern wird beeinflusst von der Wichtigkeit der einzelnen Teile für die angestrebte Funktionalität und der Struktur der Implementation. Das bedeutet erstens, viele Teile der Grammatik einfach auszulassen, um die Grammatik insgesamt überhaupt beherrschbar zu machen und zweitens manchmal althergebrachte Begriffe der Grammatik anders aufzufassen, weil sich daraus eine glattere Struktur ergibt, die eine Erleichterung der Beschreibung oder der Implementation verspricht.

Die Beschreibung beginnt nicht von oben, von der Satzebene her, sondern von unten bei den Wortarten. Leider lassen sich zahlreiche Vorwärtsverweise nicht vermeiden; es ist aber im Interesse der Lesbarkeit dieses Kapitels ohnehin sinnvoll, ein gewisses (allerdings geringes) Grundwissen über die deutsche Grammatik vorauszusetzen — und das tue ich dann auch.

Wenn in diesem Kapitel von “Fall” oder “Kasus” die Rede ist, ist stets der grammatikalische Fall gemeint, nicht der Rollenbegriff der Fallschablonentheorie; für das Wort “Kasus” gilt das ohnehin immer.

Das hier präsentierte Wissen über die Deutsche Grammatik wurde zum größten Teil dem *Duden, Band 4: Die Grammatik* [Duden(4)] entnommen, auf diesen verweisen auch die Nummernangaben. So steht z.B. (D112-132) als ein Verweis auf die Randnummern 112 bis 132 in der angegebenen Ausgabe.

2.1 Wortarten

Die Wortart eines Wortes aus dem Eingabesatz läßt sich nur anhand eines Wörterbuches ermitteln. Dieses enthält ferner auch noch eine Reihe weiterer Angaben zum Wort, die für die weitere Verarbeitung notwendig sind, insbesondere Beugungsinformation. Aus diesem Grund nimmt das Wörterbuch eine sehr wichtige Position ein. Die Wortarten und deren Unterwortarten können auch zur Bildung und Beschreibung semantischer Kategorien herangezogen werden; sie haben deshalb für ein Fallschablonensystem eine Bedeutung, die über bloße grammatikalische Aspekte hinausgeht.

2.1.1 Verb

Verben bilden das Prädikat eines Satzes, das nach den meisten Grammatiktheorien, so auch in der Fallschablonentheorie, den Kern des Satzes darstellt, also der wesentlich bedeutungstragende Teil ist.

2.1.1.1 Arten von Verben

Man unterscheidet zwischen

- den *Vollverben* (D125), die alleine das Prädikat bilden können, und dabei wiederum (beispielsweise)
 - den *Tätigkeitsverben* (D116), die eine Aktivität des Subjekts angeben,
 - den *Vorgangsverben* (D117), die eine Veränderung oder Aktivität *am* Subjekt angeben und
 - den *Zustandsverben* (D118), die keine Aktivität angeben (“statische Verben”)
- den *Hilfsverben* **sein**, **haben** und **werden** (D126), die zur Bildung der Zeiten und des Passivs benötigt werden und
- den *Modalverben* **dürfen**, **können**, **mögen**, **müssen**, **sollen**, **wollen** und zunehmend auch **brauchen** (D127), die die Bedeutung des gesamten Prädikats und damit der ganzen Aussage modifizieren. Der genaue Gebrauch und Sinn der Modalverben (D130-167), einschließlich der konkurrierenden Konstruktion des *modalen Infinitivs*, ist zu komplex, um hier beschrieben zu werden (siehe aber Abschnitt 2.4).

Sowohl die Hilf- als auch die Modalverben können an der Stelle eines Vollverbs als solches gebraucht werden. Diese Dreiteilung beschreibt eine unterschiedliche Handhabung der Worte im Analyseprozeß: Vollverben treten als zentrales Element einer Fallschablone auch nach außen hin in Erscheinung, Hilfsverben werden vollständig im Innern der Analyse abgehandelt und wirken sich im Ergebnis nur auf die Modalangaben (Zeit, Genus verbi, Modus) aus und Modalverben schließlich werden im Analyseresultat getrennt als der Fallschablone zugehörig angegeben und bewirken somit eine “Färbung” der Aussage.

Für die semantische Gestaltung ist zum einen die Sinnunterteilung der Vollverben wie oben angegeben und zum zweiten der Begriff der *Verbklassen* hilfreich: jedes Vollverb gehört einer oder mehreren der folgenden Verbklassen an, wodurch sein semantischer Gehalt und damit seine Verwendungsmöglichkeiten eingengt werden.

- *absolute Verben* sind solche, die ganz ohne Objekte stehen können; manchmal im Grunde sogar ohne Subjekt, das dann nur aus grammatikalischen Gründen in einer Pseudoform ergänzt wird (*Es regnet.*).
- Alle Verben, die dem Sinne nach Ergänzungen brauchen, heißen *relative Verben*.
- Ein Verb heißt *transitiv (zielend)* (D167), wenn es im aktiven Satz ein Akkusativobjekt hat, das bei Umwandlung des Satzes ins Passiv zum Subjekt wird, z.B. **sehen**, **umwandeln**. Nur Tätigkeitsverben können transitiv sein.
- Alle nicht transitiven Verben heißen *intransitiv (nichtzielend)*, z.B. **gehen**, **stehen**. Alle Vorgangs- und alle Zustandsverben sind intransitiv.
- Verben, die sich mit einem Reflexivpronomen verbinden, nennt man *reflexiv (rückbezüglich)* (D168-171).
Bei den *echt reflexiven Verben* (D172) ist das Reflexivpronomen nicht wegläÙbar, ersetzbar, erfragbar oder negierbar, z.B. **sich schämen**, **sich beeilen**.
Dabei gibt es den Fall von *nur reflexiven Verben* (D173), z.B. **sich auskennen**, und den von *teilreflexiven Verben* (D174), z.B. **sich oder jemanden ängstigen**, **sich oder jemanden ärgern**, die in einer anderen Bedeutung auch irreflexiv gebraucht werden.
Bei den *unecht reflexiven Verben* (D175) ist der Gebrauch ohne Bedeutungsänderung sowohl reflexiv als auch irreflexiv möglich, z.B. **sich oder jemanden waschen**. Hier wird das Reflexivpronomen zum Satzglied und damit wegläÙbar, ersetzbar, erfragbar und negierbar; es kann mit **selbst** erweitert werden.
Eine weitere Variante sind die *reziproken (wechselbezüglichen) Verben* (D176), z.B. **sich streiten**, die als *im Plural nur reziproke Verben* (D177) vorkommen, z.B. **sich anfreunden**, als *teilreziproke Verben* (D178), die in anderer Bedeutung auch nichtreziprok vorkommen, z.B. **sich oder etwas vertragen**, und als *reziprok gebrauchte Verben* (D179), die in gleicher Bedeutung auch nichtreziprok vorkommen, z.B. **ähhneln**, **belügen**; hier kann das Reflexivpronomen durch **einander** ersetzt werden.
- Alle nicht reflexiven Verben heißen *irreflexiv*.

2.1.1.2 Präfixe

Abtrennbare (separierbare) Präfixe bei Verben sind ein Phänomen, das für sehr viele Schwierigkeiten bei der syntaktischen Analyse von Deutsch verantwortlich ist.

Ein *Präfix* ist ein Wortbestandteil, der vor ein anderes Wort gestellt werden kann und damit aus diesem ein neues Wort mit einer anderen Bedeutung macht. Die allermeisten Verben lassen keinen Präfix zu, jedoch erlauben andererseits gerade die einfachen und deshalb viel verwendeten Verben oft eine große Zahl verschiedener Präfixe.

Man unterscheidet *separierbare (abtrennbare) Präfixe*, im Duden Halbpräfixe genannt (D751-764), z.B. **ab-**, **an-**, **auf-**, **aus-**, **ein-**, **mit-**, **nach-**, **vor-**, **weiter-**, **zu-**, **zurück-**, und *nicht separierbare Präfixe*, im Duden einfach Präfixe genannt (D744-750), z.B. **ver-**, **be-**, **ent-**, **er-**. Letztere stellen jedoch kein Problem dar, da sie nur zum Verständnis der Wortbildung betrachtet werden müssen; im Hinblick auf die Satzanalyse kann man Verben mit nichtabtrennbarem Präfix problemlos als eigenständige Wörter auffassen.

Das Problem bei der Behandlung von abtrennbaren Präfixen ist folgendes: Das Verb (z.B. **geben**) und das zugehörige Verb mit Präfix (**nachgeben**) können nicht als zwei eigenständige

Wörter angesehen werden; diese Methode führte nämlich leider zu dem Problem, daß die Kategorie Verb uneinheitliches syntaktisches Verhalten zeigte:

- In Hauptsätzen wird nämlich ein Präfix im Präsens und Imperfekt abgetrennt: Ich **gebe/gab dem Druck nach**.
- Bei der Bildung des Partizip Perfekt wird die Bildungssilbe **ge-** nicht einfach vorangestellt, sondern zwischen Präfix und Grundverb eingeschoben, also **nachgegeben** statt **genachgeben**.

Es gibt auch noch Zwitterfälle, bei denen ein Präfix (sogar bei ein und demselben Wort) sowohl abtrennbar als auch nichtabtrennbar vorkommt, je nach Verwendung (Bedeutungswechsel) des Wortes, z.B. **übersetzen**: Er **setzt das Boot über**. Er **übersetzt den Text**. Diese Fälle sind akustisch an der Betonung zu unterscheiden, schriftlich jedoch nur bei den abweichenden Formen.

2.1.2 Substantiv

Gegenständliche oder abstrakte Dinge und Begriffe werden mit *Substantiven* bezeichnet. Substantive bilden den Kern eines Falles im Sinne der Fallschablonentheorie, um den sich die restlichen im betreffenden Satz zum Fall gehörenden Wörter gruppieren.

2.1.2.1 Substantivarten

Substantive lassen sich in verschiedene Bedeutungsgruppen einordnen (D325-330). Die Zugehörigkeit eines Wortes zu einer solchen Gruppe beeinflusst oft sein syntaktisches Verhalten vor allem in Hinblick auf Einschränkung der Zahlbildung und Durchbrechen der Kongruenz.

- Die erste Hauptgruppe der Substantive sind die *Konkreta*. Diese bezeichnen Gegenstände, im einzelnen
 - Eigennamen (**Karlsruhe, Lutz Prechelt, Universität Fridericana**)
 - Gattungsnamen. Eine Gattung ist eine Menge von Objekten, die wesentliche Eigenschaften gemein haben. Ein Gattungsname bezeichnet entweder die Gattung als solche oder ein einzelnes Exemplar der Gattung, z.B. die Gattung **Person** oder eine **Person**. Die Substantive dieser Klasse eignen sich zur Bildung einer Konzepthierarchie, da eine Klasse meist in einer anderen Klasse vollständig enthalten ist und ihrerseits mehrere Klassen vollständig enthält. Ein wichtiger Sonderfall ist die Klasse der *Sammelbezeichnungen*. Dies sind Wörter die bereits im Singular mehrere Objekte bezeichnen, z.B. **Herde, Familie, Volk, Wald**.
- Die zweite Hauptgruppe bilden die *Abstrakta*, die etwas nichtgegenständliches bezeichnen. Zu diesen gehören
 - Handlungen und Vorgänge: **Eingabe, Abstieg, Fahrt**. Insbesondere kann der Infinitiv eines jeden Verbs als Substantiv in dieser Kategorie verwendet werden: **Bleiben, Gehen, Wollen, Sein**.

- Menschliche Vorstellungen und Institutionen: **Seele, Physik, Kunst**
- Physikalische Eigenschaften: **Länge, Masse**
- Verhältnisse: **Nähe, Gleichheit**
- Maße: **Meter, Grad, Jahr**

2.1.2.2 Substantivbildung

Deutsch ist eine sehr substantivreiche und substantivbetonte Sprache. Aus fast allen Verben und Adjektiven lassen sich Substantive bilden, Substantive lassen sich durch die Suffixe **-chen** und **-lein** in eine Verkleinerungsform überführen und aus fast allen Substantiven lassen sich zusammengesetzte Substantive bilden. Insbesondere diese letzte Möglichkeit führt dazu, daß in den meisten längeren Texten Wörter vorkommen, die in keinem Wörterbuch zu finden sind.

2.1.3 Adjektiv

Adjektive werden auf fünf verschiedene Arten gebraucht.

- *attributive Adjektive* beschreiben Eigenschaften, z.B. **rot, schwer, himmlisch, blöd** (D439). Solche Eigenschaftswörter stehen entweder bei einem Substantiv (**rotes Auto**), bei einem anderen Adjektiv (**schrecklich kalter Kaffee**), bei einem Adverb (**tief unten liegen**) oder als ein eigenes Satzglied in der Form des Satzadjektivs (**Das Auto ist rot.**) oder des Gleichsetzungsgliedes (**Das Auto ist ein rotes**). Oftmals ist mit dem Begriff Adjektiv nur dieser attributive Typ gemeint.
- *Possessive Adjektive* kennzeichnen eine Zugehörigkeit (**ihr Programm, dein Schlip**). In dieser Verwendung kann man Sie in einem Fallschablonensystem als Füller einer Besitzerrolle auffassen. Sie können nämlich auch durch Substantive im Genitiv ersetzt sein oder durch Konstruktionen mit **von** und Dativ umschrieben.
- *Interrogative Adjektive* fragen nach einem Substantiv (**welches Programm**). Allerdings bilden die Fragewörter in einem Fallschablonensystem ein selbständiges Problem.
- *Demonstrative Adjektive* verweisen auf ein Substantiv (**dieses Programm**).
- Schließlich sind auch *Zahlwörter* wie Kardinal- und Ordinalzahlen sowie unbestimmte Zählwörter (**viel, wenig, einige, andere** etc.) als Adjektive anzusehen.

Während die Verwendung von possessiven, interrogativen und demonstrativen Adjektiven sehr geradlinig und einheitlich ist (sie können als Artikel verstanden werden), ergeben sich für die anderen beiden Arten äußerst vielfältige Möglichkeiten. Die Zahlwörter müssen wegen ihrer besonderen Stellung für Computeranwendungen in einem eigenen Abschnitt behandelt werden, es sei also hier nur kurz auf die Vielfalt semantischer Möglichkeiten (vor allem der Gradabschattung, siehe unten) bei der Verwendung attributiver Adjektive eingegangen (D507-526).

- Die einfache *Steigerung (Komparation)* mit ihren Formen Positiv, Komparativ und Superlativ dient zur Formulierung von Vergleichen.

- Dabei präsentiert ein Adjektiv in *Positiv*form eine einfache Ausprägung der jeweiligen Eigenschaft, sozusagen den Ausgangspunkt des Vergleichs.
 - Der *Komparativ* beschreibt die stärkere gegenüber *einer* anderen Ausprägung (bei zweifacher Gegenüberstellung).
 - Der *Superlativ* beschreibt die stärkere gegenüber *allen* anderen Ausprägungen (bei mehrfacher Gegenüberstellung).
- Die *Gleichsetzung* erfolgt als eine Konstruktion mit den Worten **so** und **wie**, zwischen die ein Positiv gesetzt wird. **Ich bin so schlau wie Du**. Diese Gleichsetzung kann durch Verwendung von **ebenso** oder **genauso** noch betont und in verschiedener Weise in der Form modifiziert werden (D508).
 - Der *Elativ (absoluter Superlativ)* stimmt in seiner Form mit dem Superlativ überein, dient jedoch nicht einem Vergleich (ist also nicht relativ), sondern bezeichnet einen sehr hohen Grad. Ein solcher kann auch ausgedrückt werden durch Betonungswörter wie **sehr**, **höchst**, **äußerst**, **überaus**, **ungemein** etc. in Verbindung mit einem Positiv.
 - Ein zu hoher Grad (im Sinnzusammenhang) wird ausgedrückt durch das Wort **(all)zu** mit einem Positiv oder mit dem Präfix **über** (oder auch **hyper**): **Es war zu warm**.
 - Einen gesteigerten Grad, jedoch ohne Vergleich, zeigt **mehr als** mit dem Positiv an: **Es war mehr als warm**.
 - Einen höchsten Grad (im Sinnzusammenhang) ohne Vergleich zeigt das Wort **möglich** an: **so gut wie möglich**, **möglichst gut**, **bestmöglich**.
 - Ein zunehmender Grad, also eine zeitliche Entwicklung, kann durch **immer** mit dem Komparativ (**immer wärmer**), Positiv mit Komparativ (**warm und wärmer**), doppelten Komparativ (**wärmer und wärmer**) oder die Umschreibung **mehr und mehr** dargestellt werden.
 - Einen eingeschränkten Grad deuten Adverbien wie **mäßig** an: **Es war mäßig warm**.

2.1.4 Adverb

Die Adverben zählen zur Wortartgruppe der Partikeln.

Ein *Adverb* gibt einen Umstand an (D582-613), z.B. Menge (**viel**), Zeit (**lange**), Ort (**oben**) oder Intensität (**sehr**, **kaum**) bzw. Art und Weise (**ruhig**, **eilends**). Es modifiziert ein Verb (*adverbiale Bestimmung*), ein Adjektiv, ein Substantiv oder ein anderes Adverb (*Adverbialattribut*).

Adverbien werden nicht gebeugt, insbesondere können alle Adjektive zumindest im Prinzip in ungebeugter Form als Adverbien verwendet werden.

2.1.5 Präposition

Die Präpositionen zählen zur Wortartgruppe der Partikeln.

Mit dem Begriff *Präpositionen*, wird nicht eigentlich eine eigenständige Wortart bezeichnet, sondern eine Verwendungsart von Mitgliedern anderer Wortarten. Dieser Effekt ergibt sich auch bei anderen Wortarten, ist jedoch bei Präpositionen besonders ausgeprägt.

Präpositionen stammen nämlich ab von Lokaladverbien wie **an**, **auf**, **aus**, **bei**, **durch**, **hinter**, **mit**, **nach**, **über**, **um**, **unter**, **von**, **vor**, **wieder**, **zu**, Adjektiven (**anlässlich meines Geburtstages**) oder Partizipien (**ungeachtet der Kälte**); sie werden nicht gebeugt und bedingen einen bestimmten Fall des Substantivs mit dem sie stehen.

Mit Präpositionen werden sogenannte *Präpositionalgefüge* gebildet. Diese dienen

- als *adverbiale Bestimmung (Umstandsbestimmung)* zur Erläuterung eines Verbs (**Ich falle in den Fluß.**).
- als *Präpositionalattribut* zur Erläuterung eines Substantivs (**Das Wasser in meiner Kleidung ist naß.**).
- als *Präpositionalobjekt* zur kompletten Ausfüllung einer Rolle (**Ich fluche über mein Pech.**). Die Präposition wird von einem solchen Objekt niemals getrennt; andernfalls handelt es sich um ein abtrennbares Präfix. Welche Präpositionen möglich sind hängt vom Verb des Satzes ab.

2.1.6 Konjunktion

Die Konjunktionen zählen zur Wortartgruppe der Partikeln. *Konjunktionen (Bindewörter)* verbinden Worte, Phrasen oder Sätze und werden nicht gebeugt (D653).

Man unterscheidet die Konjunktionen nach ihrer Semantik, dabei tritt eine große Vielfalt bezeichneter Beziehungen auf. Es gibt nebenordnende und unterordnende Konjunktionen:

- *bei- oder nebenordnende (koordinierende) Konjunktionen* (D654-659) stellen Teile gleichen semantischen Gewichts gegenüber. Dabei gibt es die Varianten
 - *anreihend (kopulativ)*, z.B. **und**, **wie**, **sowie**, **sowohl als auch**
 - *ausschließend (disjunktiv)*, z.B. **oder**, **entweder oder**
 - *einschränkend, entgegensetzend (restriktiv, adversativ)*, z.B. **aber**, **allein**, **jedoch**, **nur**, **sondern**, **doch**
 - *begründend (kausal)*, z.B. **denn**
- *unterordnende (subordinierende), sog. Teilsatzkonjunktionen* (D660-674) leiten Teile niederen semantischen Gewichts oder logisch untergeordnete Teile ein. Auch hier gibt es mehrere Varianten
 - *zeitlich (temporal)*, z.B. **während**, **als**, **nachdem**, **seitdem**, **bis**, **ehe**
 - *beschreibend (modal)*, z.B. **als**, **wie**, **als ob**, **insofern**, **insoweit**
 - *begründend (kausal)*, z.B. **weil**, **da**, **so daß**, **wenn**, **obwohl**
 - ohne eigene Bedeutung, z.B. **daß**, **ob**, **wie**

Die semantische Vielfalt der Konjunktionen an sich und vor allem die flexible Verwendung macht die Analyse sehr problematisch¹. Insbesondere bei den beiden grundlegenden Konjunktionen der booleschen Logik **und** und **oder** ist es recht aufwendig, syntaktisch zu entscheiden, ob eine Verbindung von Sätzen, Wortgruppen oder einzelnen Worten vorliegt. Als Verbindungsebenen kommen in Betracht

¹Siehe auch Abschnitt 2.6.2 ab Seite 58.

1. Worte verschiedener Wortarten.
Adjektive: Er wurde grün und blau geschlagen.
Adverbien: Sie wippte auf und nieder. Es passiert morgen oder übermorgen.
Verben: Sie lachte und grinste.
Substantive: Wähle Kopf oder Zahl.
Präpositionen: Sie stehen auf und neben dem Fußballfeld.
2. verschiedene Arten von Wortgruppen (Teilsatzteile). Nominalgruppen: Setzt Euch auf die roten Stühle oder das Bett. Attributgruppen: Willst du sehr vertrocknete oder fast matschige Äpfel? Ich will die, die im Schrank sind oder die auf dem Tisch stehen. Das sind Äpfel meines Bruders oder meiner Schwestern. Präpositionalgruppen: Ich trete immer feste in oder knapp neben jeden Fettnapf.
3. Satzteile: Mein Fahrrad und meine Turnschuhe stehen im Trockenen. Im Wald und auf der Heiden, da mag ich alle leiden.
4. Teilsätze: Er stolperte über den Beckenrand und fiel ins Wasser.
5. Sätze: Ich entschieße mich zu bleiben oder ich gehe fort.

2.1.7 Sonstige

Die *bestimmten Artikel* *der, die, das* stehen mit einem Substantiv und dienen zur konkreten Benennung eines bestimmten Exemplars des von dem Substantiv bezeichneten Begriffes (D530-531). Demgegenüber dienen die *unbestimmten Artikel* *einer, eine, ein*, die ebenfalls mit einem Substantiv stehen, zur Benennung eines beliebigen, unbestimmten Exemplars des von dem Substantiv bezeichneten Begriffes und kommen nur in der Einzahl vor.

*Pronomen (Fürwörter)*² stehen an Stelle eines Substantivs und bezeichnen dieses. Es gibt verschiedene Arten von Pronomen, die jeweils in der Ein- oder Mehrzahl in der ersten, zweiten und dritten Person (dritte Person Einzahl in drei Geschlechtern) auftreten (D537-580). Man unterscheidet

- die *Personalpronomen* *ich, du, er, sie, es, wir, ihr, sie* (und im Genitiv *meiner, deiner, seiner, ihrer, seiner, unserer, eurer, ihrer*)
- die *Reflexivpronomen* *mich, dich, sich, sich, sich, uns, euch, sich*
- die *Possessivpronomen (Besitzfürwörter)* *mein, dein, sein, ihr, sein, unser, euer, ihr*
- die *Demonstrativpronomen (Zeigefürwörter)*, die nur in der dritten Person auftreten, *dieser, diese, dieses, jener, jene, jenes, der, die, das, derselbe, dieselbe, dasselbe, derjenige, diejenige, dasjenige* etc.
- die *Relativpronomen*, die ebenfalls nur in der dritten Person auftreten, *der, die, das, welcher, welche, welches*
- die *Interrogativpronomen*, ebenfalls nur in der dritten Person, *welcher, welche, welches*

²Die ursprüngliche Mehrzahlform des Wortes Pronomen lautet Pronomina. Laut Fremdwörterduden [Duden(5)] ist jedoch die schwache Pluralbildung ebenfalls korrekt. Da es eine generelle Tendenz zur zunehmenden Bildung schwacher Formen in der deutschen Sprache gibt (die ich für sinnvoll erachte) verwende ich hier, wie auch in anderen ähnlichen Fällen, bevorzugt die schwache Form.

2.2 Beugung

Das Deutsche ist eine Sprache mit einer weitgehend freien Wortgruppenstellung, d.h. die Wortstellung ist nur innerhalb kleiner Wortgruppen fest, z.B. eine Folge Artikel–Adjektiv–Substantiv, die Wortgruppen können jedoch in vielfacher Weise untereinander vertauscht oder gar in einzelne Teile zerrissen und verstreut werden, obwohl es auch hier noch Einschränkungen gibt. Die Freiheit ist aber jedenfalls so groß, daß die Identifikation der Satzbausteine nicht allein anhand der Wortstellung erfolgen kann.

Den Schlüssel zu Lösung dieses Problems liefert, die *Beugung (Konjugation)*. Man unterscheidet genauer zwischen Konjugation beim Verb auf der einen und Deklination bei Substantiv und Adjektiv auf der anderen Seite. Dabei werden die Wörter ausgehend von ihrer Grundform so verändert, daß sie zusätzlich zu ihrer Bedeutung auch noch grammatikalische Strukturinformation tragen. Mit diesem Instrument ist es möglich, nicht nur eine freie Wortstellung zu realisieren, sondern auch noch zusätzliche Information wie Modus, Genus, Zeit, Zahl und Person ohne zusätzliche Wörter im betroffenen Wort unterzubringen. Diese Möglichkeit wird zum Beispiel im Beugungssystem der lateinischen Sprache voll genutzt. Im Deutschen ist dagegen eine Mischform realisiert, in der z.B. bei der Zeitbildung des Verbs für Präsens und Imperfekt eine Darstellung nur durch die Beugung realisiert ist, während alle anderen Zeiten mit zusammengesetzten Formen (also mit Hilfswörtern) gebildet werden.

Das deutsche Beugungssystem ist sehr kompliziert, mit folgenden unangenehmen Eigenschaften. Es ist nämlich

- unsystematisch, d.h. die Regeln sind sehr umfangreich. So gibt es beispielsweise allein für die Pluralbildung bei Substantiven 11 verschiedene Regeln.
- unregelmäßig, d.h. zu fast allen Regeln gibt es auch eine große Zahl von Ausnahmen. So ist bei der obenerwähnten Pluralbildung trotz der vielen regelmäßigen Formen auch noch ein unregelmäßiger Plural möglich.
- unvollständig, d.h. nicht überall, wo Beugung nützliche Information liefern könnte, wird auch tatsächlich gebeugt, so kann zum Beispiel nicht durch Beugungsformen zwischen Wort- und Satzkonjunktionen unterschieden werden.
- uneindeutig, d.h. viele Beugungsformen für unterschiedliche grammatikalische Formen sind identisch. So fällt zum Beispiel bei weiblichen und sächlichen Substantiven der Nominativ immer mit dem Akkusativ zusammen.

2.2.1 Verb

Die relevanten Merkmale bei der Beugung des Verbs (*Konjugation*) mit ihren möglichen Werten und Beispielen in der zweiten Person sind (D183)

- *Zeit (Tempus)*: Präsens, Präteritum (Imperfekt), Perfekt, Plusquamperfekt, Futur I, Futur II,
liebst, liebtest, hast geliebt, hattest geliebt, wirst lieben, wirst geliebt haben,
- *Genus verbi*: Aktiv, (Vorgangs- oder Zustands-)Passiv
liebst, wirst geliebt, bist verzaubert,

- *Aussageweise (Modus)*: Indikativ, Konjunktiv I, Konjunktiv II (Irrealis), Imperativ, **liebst, liebest, liebtest, liebe**,
- *Person*: erste, zweite, dritte, **(ich) liebe, (du) liebst, (er/sie/es) liebt**,
- *Zahl (Numerus)*: Einzahl (Singular), Mehrzahl (Plural) **(du) liebst, (ihr) liebt**.

Verbformen, die diese Merkmale tragen, nennt man *finite Formen*. Daneben gibt es auch noch die *infiniten Formen*, die nur das Merkmal Zeit tragen (das dabei nicht in allen Werten vorkommen kann):

- die *Infinitive*³ im Präsens, Futur I und II und Perfekt **lieben, lieben werden, geliebt haben werden (oder: verzaubert sein werden), geliebt haben (oder: verzaubert sein)**,
- die *Partizipien* im Präsens und Perfekt **liebend, geliebt**.

Die Regeln für die Bildung der Beugungsformen des Verbs sind bei weitem zu umfangreich, um sie hier vollständig darzustellen; siehe dazu (D184-220). Es sollen jedoch die wesentlichen Mechanismen der Formenbildung kurz beschrieben werden, um die Möglichkeiten zu einer automatischen Analyse verstehen zu können und wenigstens die häufigsten Fälle ganz zu erfassen.

Man unterscheidet die Verben grob in regelmäßig konjugierte und unregelmäßig konjugierte Verben⁴.

2.2.1.1 regelmäßige Verben

Bei den *regelmäßigen Verben (schwachen Verben)* werden alle Formen ausgehend von einem einzigen festen Wortstamm gebildet, und zwar durch Anhängen einer Endung und beim Partizip Perfekt meist Vorschalten des Präfix **ge-** (D184-202). Der komplizierteste Fall tritt hierbei auf, wenn das Verb ein abtrennbares Präfix hat (**nachgeben**): in diesem Fall wird das **ge-** nämlich nicht vorangestellt, sondern zwischen Präfix und Stamm eingeschoben (**nachgegeben** statt **genachgeben**). Es gibt auch Verben, die das Präfix **ge-** überhaupt nicht verwenden. Dies sind (D317):

- alle Verben, die nicht auf der ersten Silbe betont sind, also vor allem alle mit einem nichtabtrennbaren Präfix und viele Fremdwörter, z.B. **entgeistert, funktioniert**,
- alle Zusammensetzungen solcher Verben, z.B. **umfunktioniert, herausposaunt** und
- alle übrigen zusammengesetzten Verben, die nicht auf dem ersten Glied betont sind, z.B. **übersetzt, durchleuchtet**

³Die unvollständige Bezeichnung "der Infinitiv" meint stets den Infinitiv Präsens.

⁴Die ursprüngliche Unterscheidung starke Verben/schwache Verben, die sich an der Veränderung des Stammvokals festmacht, ist weniger systematisch und wird deshalb heute als überholt angesehen. Es fallen nämlich nicht alle unregelmäßigen Verben in die Klasse der starken Verben.

Das eindeutige Klassenmerkmal der schwachen Verben ist die Tatsache, daß jede Form im Präteritum aus der korrespondierenden Form des Präsens dadurch hervorgeht, daß die Endung vorn mit einem t ergänzt wird, wobei aus phonetischen Gründen manchmal auch noch ein zusätzliches e eingefügt werden muß, also *ich lieb-e* → *ich lieb-te* oder *ich hast-e* → *ich hast-e-te*. Wann genau das geschieht, siehe (D193-195).

Numerus	Person	Präsens		Präteritum	
		Indikativ	Konjunktiv I	Indikativ	Konjunktiv II
Singular	1.	-e	-e	-(e)te	-(e)te
	2.	-(e)st	-est	-(e)test	-(e)test
	3.	-(e)t	-e	-(e)te	-(e)te
Plural	1.	-en	-en	-(e)ten	-(e)ten
	2.	-(e)t	-et	-(e)tet	-(e)tet
	3.	-en	-en	-(e)ten	-(e)ten

Das geklammerte e bei manchen der obigen Endungen wird eingefügt, wenn dies zur Aussprache sinnvoll ist (D193-195).
 Der Konjunktiv II fällt bei den regelmäßigen Verben in allen Formen mit dem Präteritum zusammen.
 Die Endungen der infiniten Formen lauten -(e)n (Infinitiv), -(e)nd (1. Partizip oder Partizip Präsens) und -(e)t (2. Partizip oder Partizip Perfekt).
 Die Endung des Imperativs ist -(e) (Singular) oder -(e)t (Plural).

Tabelle 2.1: Die einteiligen Formen regelmäßiger Verben

Es ergeben sich insgesamt 29 Formen, für die jedoch nur 13 verschiedene Endungen verwendet werden (D189). Das System wird dadurch mehrdeutig. So läßt sich zum Beispiel der Konjunktiv I nur in der dritten Person Einzahl garantiert vom Indikativ unterscheiden. Diese Mehrdeutigkeit wirkt sich jedoch nicht so stark aus, wie man angesichts des Mißverhältnisses in der Formenanzahl meinen sollte, da zur Unterscheidung der Formen noch andere Merkmale benutzt werden können; allerdings verkompliziert dieses Verfahren eine automatische Analyse doch erheblich.

Eine andere Komplikation ergibt sich aus der Mischförmigkeit des deutschen Beugungssystems: eine große Zahl von Formen ergibt sich nicht allein durch Beugung, sondern durch Zusammensetzung einer gebeugten Form mit einem Hilfsverb. Solche *zusammengesetzten Formen* haben Futur I und II, Perfekt, Plusquamperfekt, die Infinitive außer im Präsens und das Passiv. Die Beschreibung dieser Formenbildung ist jedoch nicht mehr der Beugung zuzurechnen und erfolgt daher bei der Beschreibung des Prädikats in Abschnitt 2.4.

2.2.1.2 unregelmäßige Verben

Bei den *unregelmäßigen Verben* lassen sich zwar auch eine ganze Reihe von Bildungsmethoden für die Verbformen angeben (D203-215), es ist aber, wie der Name schon sagt, nicht möglich, allgemein anzugeben, welche davon jeweils zutreffen. Glücklicherweise ist die Zahl unregelmäßiger Verben verhältnismäßig gering, es gibt nur wenige Hundert. Außerdem entstehen neue Verben stets nur in regelmäßiger Beugung und viele der unregelmäßigen Verben werden heute bereits auch regelmäßig gebeugt. Das zur Bildung der Zeiten aus den Präsens- und Präteritumformen gesagte gilt auch hier.

Es bietet sich angesichts der geringen Zahl unregelmäßiger Verben zu ihrer Behandlung ein

Vollformenlexikon an, also ein Verzeichnis, in dem einfach alle vorkommenden Verbformen einzeln abgespeichert sind.

Eine Sonderrolle nehmen die Verben **sein**, **haben** und **werden** ein, die als Hilfsverben eine zentrale Stellung für das Beugungssystem haben. Deren Formen sind sehr unregelmäßig und werden in den Tabellen 2.2 und 2.3 dargestellt.

Num.	P	Präsens		Präteritum	
		Indikativ	Konjunktiv I	Indikativ	Konjunktiv II
Sing.	1.	bin/habe	sei/habe	war/hatte	wäre/hätte
	2.	bist/hast	seist/habest	warst/hattest	wärest/hättest
	3.	ist/hat	sei/habe	war/hatte	wäre/hätte
Plur.	1.	sind/haben	seien/haben	waren/hatten	wären/hätten
	2.	seid/habt	seied/hab(e)t	wart/hattet	wär(e)t/hättet
	3.	sind/haben	seien/haben	waren/hatten	wären/hätten

Die zusammengesetzten Formen von **sein** werden mit **sein**, die von **haben** mit **haben** gebildet. Die infiniten Formen lauten **sein/haben** (Infinitiv), **seiend/habend** (1. Partizip oder Partizip Präsens) und **gewesen/gehabt** (2. Partizip oder Partizip Perfekt). Der Imperativ ist **sei/habe** (Singular) oder **seid/hab(e)t** (Plural). Oft wird bei 2. Person Plural Präsens und Konjunktiv I und beim Imperativ Plural von **sein** fälschlich **seid** bzw. **seiet** benutzt.

Tabelle 2.2: **Die Beugungsformen von sein und haben**

Numerus	Person	Präsens		Präteritum	
		Indikativ	Konjunktiv I	Indikativ	Konjunktiv II
Singular	1.	werde	werde	wurde	würde
	2.	wirst	werdest	wurdest	würdest
	3.	wird	werde	wurde	würde
Plural	1.	werden	werden	wurden	würden
	2.	werdet	werdet	wurdet	würdet
	3.	werden	werden	wurden	würden

Die zusammengesetzten Formen von **werden** werden mit **sein** gebildet (**Er ist geworden**). Ursprünglich lauteten die Formen im Präteritum Singular **ward**, **wardst**, **ward**. Diese sind jedoch heute allenfalls noch literarisch gebräuchlich. Die infiniten Formen lauten **werden** (Infinitiv), **werdend** (1. Partizip oder Partizip Präsens) und **worden** (2. Partizip oder Partizip Perfekt bei Verwendung als Hilfsverb) beziehungsweise **geworden** (2. Partizip oder Partizip Perfekt bei Verwendung als Vollverb). Der Imperativ ist **werde** (Singular) oder **werdet** (Plural).

Tabelle 2.3: **Die Beugungsformen von werden**

Ebenfalls eine besondere Stellung nehmen die Modalverben **dürfen**, **können**, **mögen**, **müssen**, **sollen**, **wollen** und **wissen** ein. Siehe dazu (D216).

2.2.2 Substantiv, Artikel und Pronomen

Die bei der Beugung des Substantivs (*Deklination*) veränderlichen Merkmale sind

- *Zahl (Numerus)* (D362) mit den Werten Einzahl (Singular) und Mehrzahl (Plural)

- *Fall (Kasus)* (D372) mit den Werten Nominativ (Wer–Fall), Genitiv (Wessen–Fall), Dativ (Wem–Fall) und Akkusativ (Wen–Fall)

Als eine weitere die Beugung wesentlich bestimmende Eigenschaft haben Substantive ein *Geschlecht (Genus)*, nämlich männlich (maskulinum), weiblich (femininum) oder sächlich (neutrum). Das Geschlecht unterliegt jedoch nicht der Beugung, das heißt es ist für jedes einzelne Wort immer gleich. Allerdings gibt es Substantive, die in mehreren Geschlechtern gebraucht werden (D346-350). In diesem Fall sollte jeder Sprecher einmal das von ihm verwendete festlegen. (Beispiel: **der/das Gummi**). Es gibt zwar eine Reihe von Regeln, die das Geschlecht in bestimmten Fällen festlegen⁵, im Allgemeinen handelt es sich jedoch um eine weitgehend willkürlich festgelegte Eigenschaft. Ein System zur Verarbeitung natürlicher Sprache muß also die Geschlechter der Substantive mit im Wörterbuch führen.

Dem Geschlecht zugeordnet gibt es verschiedene Artikel, diese werden gemäß Tabelle 2.4 mitgebeugt (D352). Eine spezielle Rolle nehmen dabei die unbestimmten Artikel ein: sie sind im Plural das leere Wort, das heißt die Verwendung eines unbestimmten Artikels im Plural besteht darin, daß man gar nichts schreibt.

Zahl	Fall	männlich	weiblich	sächlich
Singular	Nominativ	der/ein	die/eine	das/ein
	Genitiv	des/eines	der/einer	des/eines
	Dativ	dem/einem	der/einer	dem/einem
	Akkusativ	den/einen	die/eine	das/ein
Plural	Nominativ	die/ –	die/ –	die/ –
	Genitiv	der/ –	der/ –	der/ –
	Dativ	den/ –	den/ –	den/ –
	Akkusativ	die/ –	die/ –	die/ –

Tabelle 2.4: **Beugung des bestimmten und des unbestimmten Artikels**

Außerdem kann das Wort **ein** in Verbindung mit mengenanzeigenden Wörtern wie **wenig** oder **paar** als Quantifizierer vorkommen; in diesem Falle wird es nicht gebeugt — es handelt sich dann nicht um einen Artikel. Der Artikel kann mit einer Präposition verschmelzen (D358), wie bei **an dem** → **am**, **durch das** → **durchs**.

2.2.2.1 Singular

Bei der Beugung von Substantiven im Singular unterscheidet man drei Arten, die man zugleich zur Bildung der *Substantivklassen* S1, S2 und S3 heranzieht (D378). Aus mnemonischen Gründen werden die Klassen hier mit Ss, Sn und S bezeichnet.

Für die Beugungsendungen siehe Tabelle 2.5. Die Klassen können wie folgt näher beschrieben werden (D381):

- Zur Klasse S1 (Ss), dem s–Singular, gehören sächliche Substantive mit Ausnahme von **Herz**, das in einer unregelmäßigen Weise gebeugt wird, sowie alle männlichen

⁵Dabei gibt es bedeutungsbasierte Regeln, z.B. bei Abkürzungen und Schiffen, syntaktische Regeln, nämlich Endungen, die ein Geschlecht festlegen, z.B. -heit (stets weiblich) und Mischformen, z.B. bei geographischen Namen.

Fall	Ss: s-Singular	Sn: n-Singular	S: Nullsingular
Nominativ	der Mann	der Mensch	die Frau
Genitiv	des Mann-(e)s	des Mensch-en	der Frau
Dativ	dem Mann-(e)	dem Mensch-en	der Frau
Akkusativ	den Mann	den Mensch-(en)	die Frau

Bei den Sn-Endungen entfällt das e immer dann, wenn das Wort ohnehin auf e endet.

Tabelle 2.5: **Beugung der Substantive in der Einzahl**

Substantive, die nicht in die Klasse S2 fallen und viele Fremdwörter. Für das Auftreten oder Nichtauftreten des **e** in den Endungen von Genitiv und Dativ gibt es eine ganze Reihe von Regeln (D379-380); in sehr vielen Fällen ist beides erlaubt. Allerdings hat mit Ausnahme einiger formelhafter Wendungen wie **im Grunde genommen** das Dativ-e heute einen altmodischen Klang.

- Die Klasse S2 (Sn), der n-Singular, enthält als Ausnahme das Wort **der Fels** und ferner alle männlichen Substantive, die im Plural nach der Form P3 dekliniert werden und ein Lebewesen bezeichnen, sowie einige Fremdwörter.
- In S3 (S), dem Nullsingular, sind genau alle weiblichen Substantive.

Auch hier gibt es, wie überall, einige Extrawürste (D382,396-398), die jedoch kaum ins Gewicht fallen.

2.2.2.2 Plural

Es gibt im Deutschen (nach meiner Zählung, basierend auf (D383) mit den Klassen P1 – P5, jedoch präzisiert und um Fremdwortplurale und die unregelmäßige Pluralbildung erweitert) folgende Methoden der Pluralbildung bei Substantiven:

1. *e-Plural (Pe)*: **Tage**
2. *e-Plural mit Umlautung (PUe)*: **Söhne**
3. *Nullplural (P)*⁶: **Schüler**⁷
4. *Nullplural mit Umlautung (PU)*: **Mütter, Töchter**
5. *n-Plural (Pn)*: **Kammern**
6. *en-Plural (Pen)*: **Menschen**
7. *er-Plural (Per)*: **Bilder**
8. *er-Plural mit Umlautung (PUer)*: **Wälder**
9. *s-Plural (Ps)*: **Omas, Opas**

⁶Der Nullplural heißt zwar auch *endungsloser Plural*, hat aber im Dativ dennoch eine Endung -n, wenn nicht das Wort ohnehin so endet: den Schüler-n, aber: den Mädchen.

⁷Man beachte: hier liegt *keine* Umlautung vor, da das Wort Schüler schon im Singular einen Umlaut aufweist.

10. *s-Plural mit Verdopplung (Pss)*: Bildnisse
11. *i-Plural (Pi)*: Soli
12. *u/e-Fremdwortplural (Pue)*: Konsilien, Textilien⁸
13. *unregelmäßiger Plural* (ohne Klasse) für manche Fremdwörter z.B. Atlanten, Kakteen, Lexika, Textilien⁹

Außer beim unregelmäßigen Fremdwortplural, der deshalb auch keine Kurzbezeichnung trägt, lassen sich für alle diese Klassen vollständige Regeln zur Formenbildung angeben, siehe Tabelle 2.6 und folgende.

Fall	e-Plural		Nullplural	
	Pe	PUe	P	PU
Nominativ	die Tag-e	die Söhn-e	die Mädchen	die Töchter
Genitiv	der Tag-e	der Söhn-e	der Mädchen	der Töchter
Dativ	den Tag-en	den Söhn-en	den Mädchen	den Töchter-n
Akkusativ	die Tag-e	die Söhn-e	die Mädchen	die Töchter

Die Endung -n im Dativ des Nullplurals steht dann, wenn der Stamm nicht auf -n, -s oder Vokal endet.
 Beachte, daß bei Mädchen keine Umlautung vorliegt, da das Wort bereits in seinem Stamm einen Umlaut hat.

Tabelle 2.6: Beugung der Substantive in der Mehrzahl (Pe,PUe,P,PU)

Fall	n-Plural		er-Plural	
	Pn	Pen	Per	PUer
Nom	die Kammer-n	die Mensch-en	die Bild-er	die Räd-er
Gen	der Kammer-n	der Mensch-en	der Bild-er	der Räd-er
Dat	den Kammer-n	den Mensch-en	den Bild-ern	den Räd-ern
Akk	die Kammer-n	die Mensch-en	die Bild-er	die Räd-er

Tabelle 2.7: Beugung der Substantive in der Mehrzahl (Pn,Pen,Per,PUer)

Allerdings kann ein Substantiv mit oder ohne Bedeutungsunterschied in mehr als einer solchen Pluralform gebräuchlich sein (vorwiegend umgangssprachlich oder im Dialekt), z.B. die Bälge, die Bälger, die Bänke, die Banken. Eine weitere rein umgangssprachliche Veränderung ist ein zusätzliches Anhängen von -s an den Plural. Dies geschieht in allen Pluralformen außer dem s-Plural, z.B. Die Jungens und Mädchens, wobei teilweise zusätzlich die originalen Pluralformen verbogen werden, z.B. im Akkusativ den Häusers (D387-391).

Die Beugung der diversen Pronomen ist sehr unregelmäßig und wird hier aus Platzgründen ausgelassen; sie kann bei Bedarf dem Anhang A entnommen werden.

⁸hier: Textilien von Textilium.

⁹hier: Textilien von Textil.

Fall	s-Plural		Fremdwortplural	
	Ps	Pss	Pi	Pue
Nominativ	die Oma-s	die Bildnis-se	die Sol-i	die Konsili-en
Genitiv	der Oma-s	der Bildnis-se	der Sol-i	der Konsili-en
Dativ	den Oma-s	den Bildnis-sen	den Sol-i	den Konsili-en
Akkusativ	die Oma-s	die Bildnis-se	die Sol-i	die Konsili-en

Für Pi und Pue ist es einfacher, den Wörtern einzeln einen Singular- und einen Pluralstamm zuzuordnen (Solo-, Soli-, Konsilium-, Konsilien-) und diese dann in s-Singular und Nullplural einzuordnen, so daß die Klassen Pi und Pue wegfallen. Andernfalls muß der Stamm zwischen Singular und Plural modifiziert werden, und zwar durch Streichen der Endung o bei Pi bzw. um bei Pue.

Tabelle 2.8: **Beugung der Substantive in der Mehrzahl (Ps,Pss,Pi,Pue)**

2.2.3 Adjektiv

Das Adjektiv tritt gebeugt auf, wenn es attributiv bei einem Substantiv steht (*des roten Pullovers*) und wenn es als Gleichsetzungsglied verwendet wird (*Die Gefahr war eine ungeheure.*). Nicht gebeugt wird es hingegen bei der Verwendung als Satzadjektiv (*Die Gefahr war ungeheuer.*) oder als Attribut bei einem anderen Adjektiv oder einem Adverb (*Er war ein unangenehm ordentlicher Mensch.*).

Bei der Beugung richtet sich das Adjektiv nach den Eigenschaften des Substantivs, zu dem es in Beziehung steht, also nach dessen Geschlecht, Zahl und Fall (D474).

Zahl	Fall	männlich	weiblich	sächlich
Singular	Nominativ	-er/-e/-er	-e	-es/-e/-es
	Genitiv	-en	-er/-en/-en	-en
	Dativ	-em/-en/-en	-er/-en/-en	-em/-en/-en
	Akkusativ	-en	-e	-es/-e/-es
Plural	Nominativ	-e/-en/-en	-e/-en/-en	-e/-en/-en
	Genitiv	-er/-en/-en	-er/-en/-en	-er/-en/-en
	Dativ	-en	-en	-en
	Akkusativ	-e/-en/-en	-e/-en/-en	-e/-en/-en

Tabelle 2.9: **Beugung der Adjektive (Typ I/II/III)**

Die dabei verwendeten Endungen hängen jedoch nicht wie beim Substantiv allein vom Wort ab, sondern wechseln je nach Art der Verwendung. Es gibt die Typen I, II und III (siehe Tabelle 2.9), die wie folgt verwendet werden (D475-500):

- Der *Typ I* steht primär, wenn das Adjektiv ohne einen Artikel verwendet wird (*Er fühlte kalten Schweiß.*); ferner auch nach endungslosen Zahladjektiven (*drei grüne Spitzen*) und den endungslosen Formen *etwas*, *manch*, *mehr*, *solch*, *viel*, *welch*, *wenig* (D475). Dieser Typ enthält Endungen mit viel Information und muß deshalb immer dort verwendet werden, wo diese Information noch nicht anderweitig geliefert wird. Da seine Formen weitgehend¹⁰ mit denen des Pronomens *dieser*, *diese*, *dieses* überein-

¹⁰Ausnahmen: der männliche und der sächliche Genitiv Singular.

stimmen, wird er auch *pronominale* oder *determinierende Deklination* des Adjektivs genannt.

- Der *Typ II* steht primär nach einem bestimmten Artikel (**Der kalte Schweiß brach ihm aus**) und ferner nach den Formen von **derselb-**, **derjenig-**, **dies-**, **jed-**, **jedwed-**, **jeglich-**, **jen-** (D476). Diese Formen enthalten, wie der bestimmte Artikel, bereits viel Information. Deshalb kommen in der Beugung des Adjektivs nur noch zwei verschiedene Formen vor, nämlich im weiblichen und sächlichen Nominativ und Akkusativ Singular das **-e**, in allen anderen Formen das **-en**.
- Der *Typ III* steht primär nach dem unbestimmten Artikel und dessen Verneinungsform (**Er fühlte keinen kalten Schweiß.**) und ferner nach den Formen der Possesivpronomen **mein**, **dein**, **sein** (D477). Dieser Typ stellt eine Mischform dar aus den informationsreichen Formen von Typ I und den informationsarmen von Typ II. Dies erklärt sich daher, daß die erwähnten Begleitworte unvollständige Beugungsinformation gegenüber dem Demonstrativpronomen **dieser**, **diese**, **dieses** aufweisen.

2.2.3.1 Komparativ

Der Komparativ wird im Normalfall durch Anhängen von **-er** an die Grundform gebildet (D513). Bei manchen Adjektiven erfolgt zusätzlich eine Umlautung, wobei auch ungeschlüssige Fälle vorkommen. Die Sprache tendiert heute aber zu den nicht umlautenden Formen. Andere Adjektive werden unregelmäßig gesteigert (**gut**, **besser**). Endet das Adjektiv auf **-el**, **-er** oder **-en** so entfällt unter bestimmten Umständen das **e** dieser Stammendung:

- bei **-el** immer: **dunkler** statt **dunkeler**
- bei **-er** mit einem Diphthong (Zweivokal) vor der Endung auch immer: **teurer** statt **teurerer**
- bei **-er** und **-en** in gebeugten Formen wahlweise, um ein drittes unbetontes **e** zu vermeiden: **der heit(e)rere**
- bei **-er** und **-en** in ungebeugten Formen selten: **heit(e)rer**

Der Komparativ wird wie ein normales Adjektiv gebeugt.

2.2.3.2 Superlativ

Der Superlativ entsteht aus der Grundform durch Anhängen von **-st** oder **-est** (D518). Welche der beiden Arten gewählt wird, hängt von Auslaut und Silbenzahl der Grundform ab:

- Immer mit **-est** stehen einsilbige und endbetonte Adjektive auf **-s**, **-sk**, **-ß**, **-x**, **-z**, **-los**, **-haft**
- Vorwiegend mit **-est** stehen einsilbige und endbetonte Adjektive auf **-d**, **-t**, **-sch**
- Vorwiegend mit **-st** stehen Adjektive die mit Diphthong oder mit Vokal plus **h** enden.

- Die restlichen, vor allem die mehrsilbigen nicht endbetonten Adjektive stehen mit -st

Ferner werden die im Komparativ umlautenden Adjektive auch im Superlativ mit Umlaut gesteigert und die Adjektive mit unregelmäßigem Komparativ haben auch unregelmäßigen Superlativ.

2.2.4 Sonstige

Konjunktionen, Präpositionen und Adverbien werden nicht gebeugt.

2.3 Kongruenz

Ein Hauptwerkzeug zur Analyse deutscher Sätze liefert das Phänomen der *Kongruenz*. So bezeichnet man eine formale Abstimmung in den Beugungsparametern innerhalb bestimmter Wortgruppen oder zwischen mehreren Wortgruppen (D1158). Die Kongruenz besteht darin, daß im Regelfall bestimmte Beugungsparameter übereinstimmen müssen, damit eine vermutete syntaktische Beziehung erfüllt sein kann; leider kann die Kongruenz in vielen Fällen durchbrochen werden — dann fällt dieses Instrument zur Analyse weg und die Verarbeitung muß auf Basis von Wortstellungsannahmen oder semantischen Einschränkungen erfolgen.

2.3.1 Subjekt—Prädikat

Die wesentliche Kongruenz im deutschen Satz ist diejenige zwischen dem Subjekt, das an vielen Stellen stehen kann, und dem Prädikat. Subjekt und Prädikat müssen in Person und Zahl übereinstimmen. Ferner bedingt ein Prädikat im Aktiv ein Subjekt im Nominativ und ein Prädikat im Passiv ein Subjekt im Dativ mit **von** oder im Akkusativ mit **durch**.

Diese Kongruenz ermöglicht die Ermittlung des Subjekts, das normalerweise (d.h. im Aktiv) die wichtigste Rolle, nämlich die des Agenten, besetzt.

Allerdings gibt es eine Reihe von Ausnahmen oder Spezialfällen:

2.3.1.1 Nichtkongruenz der Person

Ein Subjekt kann aus mehreren Teilen bestehen, die durchaus nicht dieselbe Person haben müssen, diese Subjektteile sind dann meist mit Konjunktionen verbunden — in der Regel mit **und**. In solchen Fällen steht das Prädikat normalerweise im Plural (D1160-1161) und zwar in der “kleinsten” Person, die im Subjekt vorkommt, d.h. möglichst in der ersten, alsdann in der zweiten und nur wenn alle Subjektteile in der dritten Person sind in der dritten. Dabei ist es gleichgültig, ob ein Subjektteil in der Einzahl oder der Mehrzahl steht.

Ich und du (wir) gehen schwimmen.
Ihr und er (ihr) geht schwimmen.

Von dieser Regel wird manchmal abgewichen, indem eine der anderen in den Subjektteilen vorkommenden Personen gewählt wird.

Du und dein Kaugummi gehen (statt: geht) ins Kino.

Bei Passivsätzen, die kein Subjekt haben, steht das Prädikat in der dritten Person Einzahl (D1163). Man denkt sich quasi ein künstliches, unbestimmtes Subjekt *es*.

2.3.1.2 Nichtkongruenz der Zahl

Für die Sonderfälle bei der Zahl-Kongruenz gibt es drei unterschiedliche Ausgangssituationen:

1. das Subjekt steht im Singular (D1165).

- Bei der Verwendung von Maß- oder Mengenangaben schwankt die Verwendung fast beliebig. Dies betrifft sowohl konkrete Angaben wie **ein Dutzend**, als auch unbestimmte wie **eine Menge** oder **ein Haufen**. Also nach Belieben

Ein Dutzend Schüler macht Lärm.

was grammatikalisch korrekt ist, oder

Ein Dutzend Schüler machen Lärm.

was die sogenannte *Konstruktion nach dem Sinn* darstellt.

- Das singularische Platzhaltersubjekt *Es*, das dazu dient die erste Stelle im Satz zu besetzen, wenn das Subjekt nach hinten geschoben werden soll, hat keinen Einfluß auf die Zahl des Prädikats. Dieses richtet sich nach dem tatsächlichen Subjekt.

Es gehen (nicht: geht) Lutz und sein Kaugummi vorbei.

- Steht beim Subjekt eine Apposition in der Mehrzahl, so schwankt das Prädikat ebenfalls, die Neigung richtet sich nach der Betonung.

Kapitel 2, die Grammatikregeln, enthält (oder enthalten) besonders viele Fehler.

2. das Subjekt steht im Plural (D1166).

- Analog zum Singularfall ist auch bei pluralischen Mengenangaben häufig ein Schwanken des Prädikats zu beobachten:

120 Kilo sind (oder ist) eine Menge Übergewicht.

- Ebenfalls analog zum Singularfall ist auch die (umgekehrte) Anwendung der Appositionsregel möglich, wirkt jedoch holprig:

Die ersten Seiten, meine liebste Spielwiese, sind (selten: ist) besonders geisteserleuchtet.

3. das Subjekt ist mehrteilig. Dieser Fall wurde bereits im letzten Abschnitt erklärt. Das Prädikat steht normalerweise im Plural, Abweichungen ergeben sich vor allem, wenn die Subjektsteile ohne Konjunktion hintereinandergestellt sind oder mittels **oder** verbunden werden (D1167). Genauere Regeln aufzustellen erscheint unnötig, weil Abweichungen, die über diese beiden Fälle hinausgehen, literarischem Stil zuzurechnen sind. Ohnehin dürfte es notwendig sein, bei der Feststellung der Zahlkongruenz großzügig zu verfahren.

Da mit den oben angegebenen Fällen längst nicht alle Situationen abgedeckt sind und zudem auch eine fehlerhafte Abstimmung von Subjekt und Prädikat nicht selten ist, muß die Analyse, um robust zu sein, zusätzliche Abweichungen irgendwie behandeln können.

2.3.2 Nominalphrasen

Als eine *Nominalphrase* bezeichne ich im folgenden eine Wortgruppe aus Artikel (oder einem der bei der Beschreibung der Adjektivbeugung erwähnten Vertreterpronomen), Adjektiv(en) oder Partizip(ien) und Substantiv. Dabei ist nur das Substantiv zwingend vorhanden. Die Kongruenz besteht bei Nominalphrasen darin, daß im Normalfall alle Komponenten in dieser zusammenhängenden Gruppe in Geschlecht, Zahl und Fall (manchmal auch Person) übereinstimmen müssen (D1176-1177). Diese Eigenschaft ermöglicht es, Nominalphrasen aufzuspüren und einzugrenzen; diese stellen meist (eventuell im Verbund mit einer Präposition o.ä.) jeweils einen Fall im Sinne der Fallschablonentheorie dar.

Eine Nominalphrase kann im Adjektivbereich zusätzlich Adverbien enthalten. Solche Adverbien dienen zum Beispiel der Gradabschattung des Adjektivs. Das Problem daran ist, daß Adverbien nicht gebeugt werden und dadurch die kongruente Gruppe nicht mehr zusammenhängend erscheint.

Ansonsten tritt hier der überaus angenehme Fall ein, daß es kaum Ausnahmen von der Regel gibt. Diese betreffen nur das Geschlecht und können auch noch beseitigt werden, wenn man für Kompositionen von mehreren Wörtern sowie Eigennamen das Geschlecht passend zuordnet (D1178).

2.3.3 Gleichsetzungssatz

Subjekt und Gleichsetzungsglied¹¹ kongruieren in Zahl und, wo möglich, auch in Geschlecht und Fall.

Die Kongruenz in der Zahl wird nur verletzt, wenn ein Subjekt in der Mehrzahl mit einem Gleichsetzungsglied in der Einzahl verbunden wird (D1174): **Die Deutschen sind das Volk der Dichter und Klempner.**

Die Kongruenz im Geschlecht ist natürlich normalerweise allenfalls zufällig gegeben, da die deutschen Substantive kein systematisch zugeordnetes Geschlecht haben. Es kann jedoch, wenn eines der gleichgesetzten Glieder eine Person ist, oftmals Kongruenz hergestellt werden. Das andere gleichgesetzte Glied bezeichnet dann nämlich ebenfalls eine Person und fast alle Personenbezeichnungen (ausgenommen Eigennamen) sind im Deutschen männlich (**der Schüler, der Erbe, der Kaufmann**). Alle diese Bezeichnungen eignen sich aber zur Bildung einer weiblichen Form (**die Schülerin, die Erbin, die Kauffrau**). Wo dies möglich ist, kann eine Kongruenz erzielt werden, muß jedoch nicht (D1173).

Kongruenz im Fall liegt stets vor beim Gleichsetzungsnominativ. Bei Präpositionalkonstruktionen ist der Fall von der Präposition abhängig (D1175). Der Kongruenzbegriff ist dann so zu verstehen, daß dieser angepeilte Fall auch tatsächlich auftritt.

¹¹Hier werden der Gleichsetzungsnominativ und entsprechende Konstruktionen mit Präposition behandelt, nicht jedoch der Gleichsetzungsakkusativ. Da dieser nur mit den altertümlichen Formulierungen jemanden nennen, heißen, schelten, schimpfen, schmähen vorkommt, hat er hier wenig Bedeutung.

2.3.4 Substantivisches Attribut (Apposition)

Appositionen kongruieren grundsätzlich in Geschlecht, Zahl und Fall (D1179). Dabei liegen für Geschlecht und Zahl die selben Verhältnisse vor, wie bei den Gleichsetzungskonstruktionen.¹² Demgegenüber gibt es für den Fall eine allgemeine Kongruenz, die jedoch in folgenden Situationen verletzt wird (D1180-1182):

- Eine Apposition ohne Artikel, die einem attributiven Genitiv folgt, steht fast immer im Nominativ: *Der Kamm jenes Hahns, Behüter des Misthaufens, war...*
Die Ausnahme entfällt, sobald ein Artikel verwendet wird oder die Konstruktion bei fehlender Kongruenz mehrdeutig wird.
- Beinhaltet die Apposition eine Zusammenfassung, die mit *alle* oder *jede-* eingeleitet wird, so kann sie im Nominativ stehen. Man betrachtet dann die Apposition als Ellipse eines Gleichsetzungssatzes: *Die Reihe bestand aus Enten, Corollas und Bambinos, alles fahrbare Untersätze (statt: alles fahrbaren Untersätzen) aus richtigem Blech.*
- Manchmal wird die Apposition (fälschlich) in den Dativ gesetzt. Dies geschieht vor allem anstatt des Genitivs¹³: *Am Beispiel der Ente, dem erfolgreichsten (statt: des erfolgreichsten) der kleinen Flitzer...*
- Appositionelle Glieder mit *als*, die ohne Artikel verwendet werden, stehen nach Bezugsworten im Genitiv oft im Nominativ: *Der Einfluß der Ente als Flugtier (statt: als Flugtiers)*
- Appositionelle Glieder mit *wie* werden oft als Ellipsen empfunden und deshalb mit dem Nominativ konstruiert: *An einem Tag wie jeder andere (statt: jedem anderen)*

2.4 Prädikat

(D1024-1026) Das Prädikat bildet den Kern des Satzes und wird üblicherweise nicht als ein Satzglied betrachtet.

Ein Prädikat kann einteilig sein und enthält dann eine gebeugte (finite) Form eines Verbs, das sogenannte *Finitum*. In vielen Fällen ist es jedoch mehrteilig und kann dann enthalten: finite und infinite Formen von Hilfsverben und Vollverben, finite Formen von Modalverben und abgetrennte Präfixe von Vollverben. Solche *zusammengesetzten Formen* haben Futur I und II, Perfekt, Plusquamperfekt, die Infinitive (außer im Präsens), modale Formen und das Passiv.

2.4.1 Zeitenbildung

Die Formenbildung bei den nicht modalen Formen erfolgt in gleicher Weise für regelmäßige wie für unregelmäßige Verben so (D197):

¹²Kein Wunder. Im Prinzip ist eine Apposition ja auch eine Gleichsetzung.

¹³Diesem Effekt liegt das Motto "Rettet dem Dativ" zugrunde, das auch an anderen Stellen in der deutschen Gegenwartssprache Wirkung zeigt.

- *Präsens*: Es steht die Präsensform des Vollverbs, wobei ein abtrennbares Präfix abgetrennt wird und an die allerletzte Stelle im Hauptsatz gestellt. Dies gilt im Indikativ wie im Konjunktiv I.
- *Präteritum*: analog zum Präsens, jedoch mit den Präteritumformen.
- *Futur I*: das Präsens (bei Konjunktiv II das Präteritum) von **werden** steht mit dem Infinitiv Präsens, z.B. **(es) wird gehen**.
- *Futur II*: dito, jedoch Infinitiv Perfekt, z.B. **(es) wird gegangen sein**.
- *Perfekt*: Präsens von **haben** oder von **sein**. Welches der beiden Hilfsverben im Perfekt (genauso im Plusquamperfekt) verwendet wird, hängt vom Verb ab: Mit **haben** stehen transitive Verben, die meisten reflexiven Verben und “diejenigen intransitiven Verben, die ein Geschehen in seinem unvollendeten Verlauf, in seiner Dauer ausdrücken” (D198). Die übrigen intransitiven Verben stehen mit **sein** und dem 2. Partizip, z.B. **(es) hat funktioniert, (es) ist gegangen**.
- *Plusquamperfekt*: Präteritum von **haben** oder von **sein** mit 2. Partizip, z.B. **(es) hatte funktioniert, (es) war gegangen**.
- *Infinitiv Präsens*: Dies ist die Grundform des Verbs. Der Infinitiv Präsens mit **zu** wird gebildet, in dem man bei Verben ohne abtrennbares Präfix **zu** voranstellt, bei Verben mit abtrennbarem Präfix **zu** zwischen Präfix und Rest einfügt (**nach-zu-geben**).
- *Infinitiv Futur I*: Infinitiv Präsens gefolgt von **werden**, z.B. **gehen werden**. Für Infinitiv Futur I mit **zu** benutze entsprechend den Infinitiv Präsens mit **zu**.
- *Infinitiv Futur II*: Infinitiv Perfekt mit **werden**, z.B. **gegangen sein werden**. Für Infinitiv Futur II mit **zu** benutze entsprechend den Infinitiv Perfekt mit **zu**.
- *Infinitiv Perfekt*: 2. Partizip mit **haben** oder **sein**, z.B. **gegessen haben** oder **gegangen sein**. Das **zu** wird gegebenenfalls dazwischen eingefügt.
- *Vorgangspassiv*: Formensystem von **werden** (siehe Tabelle auf Seite 40) durch alle Zeiten mit 2. Partizip, wobei **geworden** aber durch **worden** ersetzt wird, z.B. **(es) wird gewaschen, (es) ist gewaschen worden**.

2.4.2 Modale Färbung

Von diesen Formbildungen unterscheiden sich die Sätze mit modaler Färbung oder modifizierenden Verben. *Prädikate mit Modalverb* werden wie folgt konstruiert:

- im *Präsens* und im *Präteritum* steht die entsprechende Form des Modalverbs, und zwar an der Stelle, wo auch das Hilfsverb z.B. beim nicht modalen Perfekt stehen würde.
- in allen *anderen Zeiten* steht das Hilfsverb in gleicher Form und an gleicher Stelle wie im nicht modalen Satz. Der infinite Teil des Prädikats wird gebildet vom Infinitiv des Hauptverbs gefolgt vom Infinitiv¹⁴ des Modalverbs. Dies kann jeder Infinitiv sein, nicht nur der Infinitiv Präsens.

¹⁴Gelegentlich auch dem 2. Partizip; das ist jedoch falsch.

- von den Modalverben kann weder ein *Imperativ* noch ein *Passiv* gebildet werden — was natürlich manche Leute durchaus nicht daran hindert, das dennoch zu tun (zum Beispiel im Amtsdeutsch). Diese Formen sollen jedoch hier nicht betrachtet werden.

2.4.3 modifizierende Verben

Semantisch ähnlich sind *Prädikate mit modifizierendem Verb*, z.B. *Er pflegte nachzugeben*. Es ist nur das Aktiv möglich; die Formen werden gebildet als wäre das modifizierende Verb (im Beispiel *pflegen*) alleiniges Verb und das Prädikat wird mit einem Infinitiv mit **zu** ergänzt. Dieser Infinitiv mit **zu** steht bei ansonsten einteiligen Prädikaten am Satzende, ansonsten vor dem infiniten Teil bzw. vor dem abgetrennten Präfix. Modifizierende Verben können beispielsweise sein *pflegen*, *scheinen*, *vermögen*, *ermächtigen* etc. Das Wort *brauchen* nimmt eine Zwitterstellung ein: es wird sowohl als modifizierendes als auch als Modalverb gebraucht.

2.5 Satzteile

Hier nun die wichtigsten Satzteile zur Konstruktion deutscher Sätze (D1031-1051). Kennzeichnende Eigenschaft von Satzgliedern ist es, sich nur geschlossen verschieben zu lassen. Grundsätzlich ist festzustellen, daß sich die einzelnen Satzglieder nicht in allen Fällen voneinander unterscheiden lassen. Bei der Betrachtung ergibt sich eine Einteilung in vier Klassen, nämlich das Subjekt, das den Ansatzpunkt des Geschehens bezeichnet, die Objekte, welche die Zielpunkte angeben, die adverbial gebrauchten Satzteile, die die Umstände bestimmen und die sonstigen Satzteile, welche auf den Satz als Ganzes wirken.

Es wurden in dieser Auflistung diejenigen Satzteile außer acht gelassen, die in dem hier beschriebenen Rahmen keine oder nur eine sehr geringe Bedeutung haben. Dazu zählen:

- der Gleichsetzungsakkusativ
Er nennt seinen Hund *einen Idioten*.
- der Anredenominativ
Herr Müller, halten Sie doch bitte den Mund!
- der absolute Nominativ
Schließlich hielt er den Mund, *ein guter Entschluß*.
- der absolute Akkusativ
Er saß also da, *die Augen geschlossen*, und lauschte nur noch.

2.5.1 Subjekt

(D1031-1032) Das Subjekt ist der *formale Ansatzpunkt des Geschehens*. Es steht im Nominativ, wenn das Glied einfach und im Fall bestimmt ist und kann mit **Wer** oder **Was?** erfragt werden; stehen wegen eines Gleichsetzungsnominativs zwei Nominative, ist in der Regel der erste das Subjekt. Eine kompliziertere Möglichkeit ist die, daß die Stelle des Subjekts von einem Teilsatz oder einer Infinitivkonstruktion besetzt wird, wie in

Wer andern in der Nase bohrt, ist selbst ein Schwein.
Andern in der Nase zu bohren, ist nicht einfach.

Bei der Umwandlung eines aktiven Satzes ins Passiv wird aus dem Subjekt entweder **von** und Dativ oder **durch** und Akkusativ.

Eine Sonderrolle nimmt der Platzhalter **Es** ein. Dieser Platzhalter dient dazu die erste Stelle im Satz zu besetzen, wenn das Subjekt hinter das Prädikat (oder dessen ersten Teil) geschoben werden soll.

Ein Mann ging unter.
Es ging ein Mann unter.

In diesem Falle handelt es sich bei **es** im zweiten Satz nicht um ein Pronomen — *Es* wird nicht gebeugt und kongruiert nicht (oder nur scheinbar) mit dem Prädikat.

Es ging ein Mann unter.
Aber: Es gingen zwei Männer unter.

In der automatischen Analyse ist dieser Platzhalter nur dann von einem Pronomen zu unterscheiden, wenn eben keine scheinbare Kongruenz vorliegt. Die Lage wird dadurch noch weiter kompliziert, daß **es** noch eine dritte Verwendung hat. Es kann nämlich auch noch ein Quasi-Subjekt für unpersönliche Verben sein: **Es regnet**. In dieser Verwendung ist es zwar erfragbar, kann aber nur schwer ersetzt werden; es stellt jedoch grammatikalisch ein echtes Subjekt dar.

Im aktiven Satz ist das Subjekt im weitesten Sinne immer in der Rolle des Agenten. Im passiven Satz ist es ein *affiziertes Subjekt*.

2.5.2 Objekte

(D1035) Die Objekte lassen sich als die Zielpunkte des verbalen Geschehens auffassen. Sie sind in ihrem Kasus direkt durch das Verb oder (seltener) durch ein Satzadjektiv bestimmt und ihre Stelle kann immer durch ein Pronomen besetzt sein. (Bei Präpositionalobjekten bleibt natürlich die Präposition mit dem Pronomen stehen.)

2.5.2.1 Akkusativobjekt

(D1035) Das häufigste Objekt ist nun das Akkusativobjekt. Seine Stelle ist meist durch ein Glied im Akkusativ besetzt: **Ich sehe den Tisch**. Es kann jedoch auch ein Teilsatz an der Stelle eines Akkusativobjektes stehen: **Ich sehe, daß dort ein Tisch ist**. Das Akkusativobjekt kann mit **Wen?** oder **Was?** erfragt werden; im Fall des Teilsatzes ist die Frage als **Welchen Sachverhalt?** zu verstehen.

2.5.2.2 Dativobjekt

(D1036) Ein Dativobjekt ist meist von einem Glied im Dativ besetzt: **Das gehört mir**. Alternativ kann ein Teilsatz, nämlich ein Relativsatz, stehen: **Er hilft, wem er kann**. Das Erfragen eines Dativobjektes ist mit **Wem?** möglich.

2.5.2.3 Präpositionalobjekt

(D1038-1039) Ein Präpositionalobjekt besteht aus einer Substantivgruppe (oder einem Pronomen) und einer vorangestellten Präposition. Welche Präpositionen stehen können, wird vom Verb des Satzes bestimmt (**stehen auf, hoffen auf, stehen zu**, (aber z.B. nicht:) **hoffen zu**). Die Präposition und das Verb zusammen wiederum legen fest, in welchem Fall die Substantivgruppe steht; es läßt sich danach ein Präpositionalobjekt im Akkusativ und eines im Dativ unterscheiden (**stehen auf wem?, hoffen auf wen?**). Im Genitiv kommt es nicht vor.

Kennzeichnend für Präpositionalobjekte gegenüber den adverbialen Präpositionalgefügen ist, daß die Präposition bei letzteren beliebig (mit entsprechender Bedeutungsänderung) aus einer ganzen Gruppe ausgewählt werden kann: **Er wartet auf das Rendezvous.** gegenüber **Er wartet auf/neben/hinter/vor dem Denkmal.** Ein Präpositionalgefüge kann auch ohne die Verwendung einer Präposition erfragt werden (**Wo wartet er?**), wohingegen bei Präpositionalobjekten die Präposition zwingend Bestandteil der Frage ist (**Worauf/Auf was wartet er?**).

2.5.2.4 Genitivobjekt

(D1037) Das Genitivobjekt ist relativ selten geworden. Es ist durch ein Glied im Genitiv besetzt und kann mit **Wessen?** erfragt werden. **Er erinnerte sich seiner Schulzeit.**

2.5.2.5 Gleichsetzungsnominativ

Der Gleichsetzungsnominativ entspringt der semantischen Sonderstellung von Verben wie **sein, scheinen, bleiben, werden**. Diese setzen nämlich etwas dem Subjekt gleich. Dieser Begriff der Gleichsetzung darf allerdings nicht zu eng verstanden werden. Meist wird hier nur eine Klasse oder Gattung angegeben, zu der das Subjekt gehört. Dieses gleichgesetzte Etwas steht grammatikalisch im Gleichsetzungsnominativ (D1033): **Die Lage bleibt ein Drama. Sie ist ein Genie. Sie wird Erfinderin.** Im Sinne der oben erwähnten Einteilung wird der Gleichsetzungsnominativ in die Gruppe der Objekte gerechnet.

2.5.3 Adverbial gebrauchte Satzglieder

(D1053-1057) Adverbiale Bestimmungen beziehen sich nicht immer, wie der Name vermuten ließe, auf das Verb. Sie kommen vielmehr in zwei semantisch unterschiedlichen Verwendungsarten vor.

Sie können sich gewissermaßen von außen, also ohne direkt dazuzugehören, auf die Aussage als Ganzes beziehen, z.B. **Wahrscheinlich/Meines Erachtens/Nach meinem Eindruck ist das falsch.**

Oder sie beziehen sich von innen, also als Bestandteil der Aussage auf das Verb oder die Aussage als Ganzes, z.B. **Sie trug das Paket vorsichtig. Heute bin ich dabei.** Diese Gruppe läßt sich gut weiter unterteilen in die Bestimmungen des Raumes, der Zeit, der Art und Weise und des Grundes.

- Die *adverbialen Bestimmungen des Raumes* lassen sich unterteilen in die Bestimmung
 - des Ortes (**Wo?**),
 - der Richtung (**Wohin?**),
 - der Herkunft (**Woher?**) und
 - der räumlichen Erstreckung (**Wie weit?**).
- Ganz ähnlich sind die *adverbialen Bestimmungen der Zeit*. Sie beschreiben
 - des Zeitpunkts (**Wann?**),
 - der Wiederholung (**Wie oft?**),
 - der zeitlichen Erstreckung (**Wie lange?**),
 - des Beginns (**Seit wann?**) oder
 - des Endes (**Bis wann?**).
- Die *adverbialen Bestimmungen der Art und Weise* kommen vor
 - in ihrer Urform (**Wie?**), sowie als Bestimmung
 - der Quantität (**Wieviel?**),
 - der Intensität oder des Grades (**Wie sehr?**),
 - der graduellen Differenz (**Um wieviel?**)
 - der stofflichen Beschaffenheit (**Woraus?**),
 - des Mittels (**Womit/Wodurch?**) und
 - der Begleitung (**Mit wem?**).
 - des Resultats (**Mit welchem Ergebnis?**).
- Schließlich gibt es die *adverbiale Bestimmung des Grundes*, die
 - in ihrer Urform vorkommt (**Warum?**), oder als Bestimmung
 - der Bedingung (**In welchem Fall, Unter welchen Umständen, Unter welcher Bedingung?**),
 - der Folgerung (**Aufgrund wessen?**),
 - des Zwecks (**Wozu?**) und
 - eines Hindernisses (**Trotz wessen?**).

2.5.3.1 Adverbiales Präpositionalgefüge

(D1040) Die adverbialen Präpositionalgefüge kommen in der ganzen Vielfalt, wie in der Einleitung beschrieben, vor. Zur Abgrenzung siehe den Abschnitt 2.5.2.3 über die Präpositionalobjekte auf Seite 53. Man kann die adverbialen Präpositionalgefüge einteilen nach dem Kasus mit dem sie stehen:

- am häufigsten ist der Dativ: *Seit dem Essen* habe ich nichts mehr gegessen.
- ebenfalls recht häufig kommt es mit dem Akkusativ vor: *Dieser Zustand bleibt nur für kurze Zeit*.

- seltener steht es mit Genitiv: Dies gilt *wegen meines Hungers*. Die Verwendung des Genitivs ist heute auch deshalb selten, weil mit den Präpositionen, die eigentlich mit Genitiv stehen müssen, oft der Dativ verwendet wird: Dies gilt *wegen meinem Hunger*. Dies ist aber genaugenommen falsch.

2.5.3.2 Adverbialgenitiv

(D1042) Der Adverbialgenitiv ist (wie das Genitivobjekt) von einem Glied im Genitiv besetzt. Dieses ist jedoch nicht durch ein Pronomen ersetzbar und also auch nicht erfragbar. Sein Kasus ist auch nicht durch ein Element seiner Umgebung bestimmt, sondern als feste Wendung vorgegeben: *Eines Abends* sah ich es. Im Gegensatz zum Genitivobjekt kann ein Adverbialgenitiv durch ein Glied ersetzt werden, das nicht im Fall bestimmt ist: *Neulich* sah ich es. Es nur wenige solche Wendungen, von denen die meisten eine Zeit angeben. Eine Ausnahme ist z.B. *meines Erachtens*.

2.5.3.3 Adverbialakkusativ

(D1041) Ganz Ähnliches gilt für den Adverbialakkusativ. Auch er ist ein im Fall bestimmtes Glied, das nicht pronominal ersetzbar ist und auch nicht erfragt werden kann. Er war *den ganzen Tag* baden. Und auch er läßt sich durch im Fall unbestimmte Glieder ersetzen. Er war *lange* baden. Leider gibt es im Falle des Adverbialakkusativs wesentlich mehr mögliche Konstruktionen, so daß die automatische Analyse schwierig ist, denn eine Unterscheidung vom Akkusativobjekt ist oft weder nach der Form noch nach der Stellung möglich:

Er hat den ganzen Tag den ganzen Wagen gewienert.
 Er hat den ganzen Wagen den ganzen Tag gewienert.
 Den ganzen... (etc.)

2.5.4 Zugeordnete Glieder

(D1043) Diese Satzglieder kann man auch als verkürzte Nebensätze auffassen. Es handelt sich um im Kasus bestimmte Glieder, die durch Satzteilkonjunktionen (vor allem *wie* und *als*) eingeleitet werden und oft eine Erweiterung eines an sich schon vollständigen Satzes bewirken. Im Gegensatz zu Präpositionen fordern die Satzteilkonjunktionen keinen bestimmten Fall des Gliedes, das sie einleiten. Das zugeordnete Glied kann nun auf ein Satzglied des gleichen Satzes bezogen werden und steht dann im gleichen Fall wie dieses:

Ich vertraute ihm *wie einem Freund*.
 Ich vertraute ihm *wie ein Dummkopf*.
 Ich schätze Nudeln *als eine Delikatesse*.
 Ich schätze Nudeln *als ein Delikatessenliebhaber*.

Sie können jedoch auch ohne einen solchen Bezug vorkommen:

Ich schätze Nudeln *wie die Italiener*.
 Hier sieht es aus *wie im Saustall*.

Dies geschieht vor allem, wenn es sich (wie im letzten Fall) um zugeordnete Präposition-
algefüge handelt.

2.5.5 Satzadjektiv

(D1045-1047) Das Satzadjektiv ist ein im Kasus unbestimmtes Satzglied, das mit einem
Adjektiv oder einem 1. Partizip (das als Adjektiv gebraucht wird) besetzt ist. Das Adjektiv
ist dabei ungebogen. Es kann sich auf ganz unterschiedliche Glieder des Satzes beziehen:

- auf das Subjekt. Diese Konstruktion ähnelt dem Gleichsetzungsnominativ und ist
deshalb auch mit den gleichen Verben wie dieser möglich: *Es ist groß. Es hatte gut
ausgesehen.*
- auf ein Objekt. Dies ist vor allem bei Verben möglich, die ein Wahrnehmen oder ein
Urteilen bezeichnen: *Ich finde meine Nase doof.*
- auf das Prädikat. Das Satzadjektiv dient zu dessen näherer Bestimmung: *Er schläft
tief. Deshalb flucht sein Lehrer laut.* Hier kann man das Satzadjektiv auch als ein
Adverb auffassen und dem Prädikat unmittelbar beordnen.

Wie man sieht, ist der Bezug schlecht an formalen Merkmalen abzulesen; es ist Wissen
nötig, um ihn zu erkennen. Dies wird besonders deutlich an folgendem Beispiel, in dem
das Satzadjektiv durch bloßes Austauschen, ohne jede Änderung der Satzkonstruktion jede
der drei Anwendungsarten einmal einnimmt:

Der Beamte verlangte den Ausweis zerstreut.
Der Beamte verlangte den Ausweis aufgeschlagen.
Der Beamte verlangte den Ausweis lautstark.¹⁵

Zu den Satzadjektiven gehören außerdem solche Adjektivkonstruktionen, die mit Präposi-
tionen gebaut werden und meist feste Redewendungen darstellen, die sogenannten *präpo-
sitionalen Satzadjektive* wie *von klein auf, über kurz oder lang* und die damit verwandten
durch eine Satzteilkonjunktion, z.B. *wie* oder *als* abgetrennten *zugeordneten Satzadjektive*
wie *Es sah aus wie neu.*

2.5.6 Satzpartikel

Satzpartikel nennt man ein im Fall unbestimmtes Satzglied mit Partikel (genauer: Adverb)
im Kern (D1048-1050), z.B. *Gestern kam sie. Dort war es.*

Analog zur Situation bei den Satzadjektiven gibt es auch bei den Satzpartikeln solche,
die mit Präpositionen stehen und *präpositionale Satzpartikeln* genannt werden, z.B. *Bis
gestern ging es. Alles Gute kommt von oben.* Auch die *zugeordneten Satzpartikel* existieren
in exakt analoger Weise: *Mir geht es wie immer.*

¹⁵Dies sind der Reihe nach Subjektbezug (Grenzfall), Objektbezug, Prädikatsbezug.

2.6 Satz

2.6.1 Grundsätzliches

Der Satz stellt, verglichen mit dem bisher Besprochenen, eine neue Stufe sprachlicher Einheiten dar. Ein Satz ist grammatikalisch aus den bisher beschriebenen kleineren Einheiten aufgebaut und (mit gewissen Einschränkungen) inhaltlich abgeschlossen. Angesichts der daraus erwachsenden Vielfalt der Struktur *Satz* stellt sich die Frage nach einer Grobunterteilung. Eine solche Unterteilung könnte nach formalen oder inhaltlichen Merkmalen vorgenommen werden. In der deutschen Sprache bieten sich dazu der formale Begriff der Satzform und der formale und inhaltliche Begriff der Satzart an.

Bei den *Satzformen* unterscheidet man (D1012)

- den *einfachen Satz*, der in der Regel um ein einziges Verb gebildet wird. **Ich betrachte meinen Fuß.** Zum einfachen Satz kann man sinnvollerweise auch bestimmte Konstruktionen mit mehreren Verben rechnen, zumeist einer finiten und einer infiniten Form. Dazu zählen vor allem Sätze mit Modalverb. **Ich möchte meinen Fuß betrachten.**
- die *Satzverbindung oder Nebenordnung (Parataxe)*, die aus aneinandergereihten Sätzen besteht, die jeder auch einzeln stehen könnten und etwa gleichen Rang haben. **Der Mann war schlaff, er hatte eine grünliche Gesichtsfarbe und er sah die Welt durch die viereckigen Augen eines Informatikers.** Eine solche Verbindung ist recht lose.
- das *Satzgefüge (Hypotaxe)*, einem mehrteiligen Satz bei dem die einzelnen Bestandteile ineinander verfugt sind. Sie bestehen aus mindestens einem Hauptsatz und einem Nebensatz der nicht allein stehen kann.

Eine andere Unterteilung, die nicht nur vom grammatikalischen, sondern auch vom semantischen Standpunkt aus von Interesse ist (D1008), ist jene nach der *Satzart*¹⁶:

- Die häufigste Satzart in der deutschen Sprache ist der *Aussagesatz* (D1009); er wird darum als Standardform aufgefaßt. Ein Aussagesatz behauptet einen Sachverhalt oder teilt ihn mit. Das Finitum steht an zweiter Stelle. **Er betrachtet seinen Fuß.**
- Damit verwandt ist der *Ausrufesatz* (D1009). Er hat im Prinzip den gleichen semantischen Grundton, es wird jedoch zusätzlich eine starke innere Bewegung ausgedrückt, etwa Erstaunen oder Bewunderung. Das Finitum kann hier an erster, zweiter oder letzter Stelle stehen. **Betrachtet er aber intensiv seinen Fuß! Oh, er betrachtet seinen Fuß! Wie schön er seinen Fuß betrachtet!** Die Analyse solcher Sätze ist wegen der freien Verbstellung eventuell schwierig; aber sie haben für die Verarbeitung natürlicher Sprache ohnehin kaum eine Bedeutung, oder können einfach wie ein Aussagesatz verstanden werden, und werden deshalb hier nicht weiter betrachtet.
- Ein *Wunschsatz* drückt etwas aus, das vom Sprecher erwünscht oder angestrebt wird (D1010), sei es nun erreichbar oder nicht. Ein Wunschsatz muß sich nicht notwendig an eine bestimmte Person richten. Wunschsätze stehen im Konjunktiv, das Finitum

¹⁶Für Bemerkungen zur pragmatischen Interpretation siehe Abschnitt 1.2.1.3 über die Pragmatik auf Seite 20

nimmt die erste, zweite oder letzte Stelle ein. Fiele mir doch nur ein besseres Beispiel ein! Mein Gehirn helfe mir! Wenn mir doch nur eine Idee käme! Für die Schwierigkeit und Wichtigkeit gilt das zu Ausrufesätzen gesagte.

- Ein *Aufforderungssatz* enthält ebenfalls einen Wunsch (D1010). Dieser ist jedoch nicht ausdrücklich dargestellt, sondern der Satz richtet eine Aufforderung zu dessen Erfüllung an ein bestimmtes Gegenüber. Das Finitum ist ein Imperativ und steht an erster Stelle. **Betrachte deinen Fuß!**
- Mit einem *Fragesatz* (D1011) wird eine Information angefordert; er ist also im Normalfall an ein bestimmtes Gegenüber gerichtet. Dabei ergibt sich eine weitere Unterteilung (D1258) in
 - *Ja/Nein-Fragen*, deren Beantwortung eine Einschätzung von Wahrheit oder Falschheit ist. Hier steht das Finitum an der ersten Stelle. **Betrachtest Du Deinen kleinen Zeh ?**
 - *W-Fragen*, deren Beantwortung die Information einer Art erfordert, die durch das jeweilig in der Frage vorkommende Fragewort (oder eine Frageformel) bestimmt ist. Diese Frageworte fangen zumeist mit **W** an¹⁷, daher der Name. Hier steht das Finitum an der zweiten Stelle. **Wer bist Du? Welches Buch meinst Du?** Die *W-Fragen* sind zu unterteilen in *schwach gerichtete Fragen*, die keine Objektangabe enthalten (**Wer?**) und *stark gerichtete Fragen*, die das erfrage Objekt mittels einer Substantivphrase mit angeben (**Welches Buch?**). Die Frageformel kann statt mit einem Fragewort auch mit einer Präposition beginnen (**In welche Stadt? Auf wen?**).

2.6.2 Nebensätze

Für die Einteilung von Nebensätzen bieten sich drei verschiedene Möglichkeiten an. Zwei davon sind syntaktischer Natur, nämlich die äußere Form der Nebensätze und die Position, die sie im Satz einnehmen. Das dritte ist von semantischer Natur, nämlich die inhaltliche Beziehung zwischen dem Haupt- und dem Nebensatz.

Die Unterscheidung aufgrund der Form (D1192) orientiert sich vor allem an den Nebensatzeinleitungen; man unterscheidet

- *Konjunktionalsätze*, die durch eine Konjunktion eingeleitet werden: Ich sehe, **daß** dies ein Konjunktionalsatz ist.
- *Relativsätze*, die durch ein Relativpronomen oder eine Relativpartikel eingeleitet werden: Dies ist ein Satz, **der** einen Relativsatz enthält.
- *indirekte Fragesätze*, die durch ein Fragepronomen oder eine Fragepartikel eingeleitet werden: Dort, **wo** das Komma steht, beginnt er.
- *uneingeleitete Nebensätze*
- *satzwertige Infinitivkonstruktionen*: Ich entscheide mich, **einen Infinitiv zu verwenden**.
- *satzwertige Partizipialkonstruktionen*: Ich sitze hier, **ein Beispiel erfindend**.

¹⁷siehe Abschnitt 2.5.3 ab Seite 53

Die Unterscheidung nach der Position (D1193) beschreibt, was für eine Stelle es ist, die der Nebensatz besetzt. Dies kann die Stelle eines Satzgliedes (*Gliedsatz*) oder eines Attributs sein. Man nennt diese

- *Subjektsatz*, der Nebensatz besetzt die Stelle des Subjekts: *Wer dies liest, versteht es hoffentlich auch.*
- *Objektsatz*, der Nebensatz besetzt die Stelle eines Akkusativobjekts (Ich überlege, *was mir einfallen könnte.* Ich bemerke, *daß es nichts tolles ist.*), eines Dativobjekts (Ich sehe also zu, *wie der Sekundenzeiger voranrückt.*), eines Genitivobjekts (Ich erinnere mich dabei, *daß ich weiterarbeiten sollte.*) oder eines Präpositionalobjekts (Ich bin erstaunt, *daß ich das schon getan habe.*).
- *Adverbialsatz*, der Nebensatz ersetzt eine adverbiale Bestimmung: *Wo eben noch nichts war, stehen jetzt Beispiele.*
- *Attributsatz*, der Nebensatz ist ein erweitertes Attribut: *Die Frustration, noch so wenig geschafft zu haben, sinkt jetzt ab.*

Die Unterscheidung nach der Beziehung schließlich eröffnet ein großes Feld von semantischen und syntaktischen Beobachtungen. Diese werden hier nur kurz angedeutet.

Anmerkung: die seltsamen Bezeichnungen 'Relativbeziehung' und 'Verhältnisbeziehung' stammen aus dem Duden.

1. *Relativbeziehungen* (D1198-1212) liegen vor, wenn der Nebensatz mit einem Relativpronomen oder einer Relativpartikel eingeleitet wird und eine gemeinsame *semantische Stelle* mit dem Hauptsatz hat. Es gibt verschiedene Arten von Relativbeziehungen:

- *uncharakterisiert* mit *der, welcher, wer, was*, z.B. Dieser Satz, der einen Relativsatz enthält, *welcher* noch einen enthält, enthält also zwei Relativsätze.
- *modal* mit *wie, als ob, wie wenn, als, wie um*, z.B. Dieser Satz enthält, *als ob* er nicht anders könnte, zwei Relativsätze, *wie um* uns zu gefallen.
- *kausal* mit *weswegen, weshalb, manchmal auch woher*, z.B. Hier liegt nur ein Relativsatz vor, *weswegen* es nicht zwei sind.
- *instrumental* mit *wodurch, womit*, z.B. Hier auch, *wodurch* sich eine genauere Erläuterung erübrigt.
- *lokal* mit *wo, wohin, woher*, z.B. Hier ist er, *wo* man ihn erwartet.

Dabei wird der Relativsatz stets als Nebensatz mit Finitum am Ende konstruiert.

2. *Inhaltsbeziehungen* (D1213-1231) sind nicht ganz so einfach zu beschreiben. Sie lassen sich am besten eingrenzen, indem man ihre Anschlußmittel beschreibt; diese erschöpfen sich nicht im Austausch von Einleiteworten, sondern es kommen auch Infinitivkonstruktionen und direkte Rede vor. Nur etwas mühsam lassen sich auch semantische Kategorien bilden:

- *rein faktisch* mit *daß* oder einfachem Infinitiv: Es ist schön, *daß*... , Es ist schön dies gesagt zu haben.
- *faktisch, verlaufsdarstellend* mit *als*: Als ich dies sagte,...

- *faktisch, hypothetisch* mit Infinitiv oder wenn: Es wäre schön, das zu sagen. Ich fände es schön, wenn ich das sagte.
 - *rein beziehend* mit Infinitiv oder daß: ...sagt, daß..., ...sagt,...zu haben
 - *beziehend auf indirekte/direkte Rede* Er sagte, er könne. Er sagte: "Ich kann."
 - *beziehend, verlaufsdarstellend* mit wie oder Akkusativ mit Infinitiv: Ich sah sie kommen. Ich merkte, wie...
 - *beziehend auf undeutliche Wahrnehmung* mit als, als ob: ..., als wüßte er nicht weiter. ..., als ob er nicht weiter wüßte.
 - *modal* mit daß (Es ist wichtig, daß...), mit Infinitiv (...vermochte dies nicht einzuschätzen.) oder mit Akkusativ und Infinitiv (...ließen ihn gehen).
 - *völlig fraglichkeitsdarstellend* mit ob oder direkter Rede: Er fragte, ob ich komme. Er fragte: "Kommst Du?"
 - *völlig fraglichkeitsaufhebend* mit ob: Ich weiß, ob ich komme.
 - *völlig alternativsetzend* mit ob: Es ist egal, ob ich komme.
 - *gezielt fraglichkeitsdarstellend* mit W-Anschluß oder direkter W-Frage: Er fragte, wann ich komme. Er fragte: "Kommst Du?"
 - *gezielt fraglichkeitsaufhebend* mit W-Anschluß: Ich weiß, warum ich komme.
 - *gezielt alternativsetzend* mit W-Anschluß: Es ist egal, wohin ich komme.
 - *umreißend* mit W-Anschluß: Ich habe gehört, wer alles kommen soll.
3. *Verhältnisbeziehungen* (D1332-1256) stellen die vielfältigste Nebensatzklasse dar, nicht nur im Hinblick auf die syntaktischen Erscheinungsformen, sondern auch, was die semantischen Interpretationen angeht.
- *Kausalsätze* geben einen Grund (*da/weil*) einen minderen Grund (*zumal* etc.) oder einen Ausgleich (*dafür, daß*) an.
 - *Konsekutivsätze* stellen eine Folge dar und zwar eine allgemeine (*so daß*), eine spezielle (*so...daß*), eine mögliche (*zu+Infinitiv*), eine negative (*zu...als daß*) oder eine modalisierte.
 - *Konzessivsätze* benennen einen unzureichenden Gegengrund oder eine Einräumung mit *obgleich, obwohl, obschon, obzwar, wenn auch, wengleich, wenschon, trotzdem*.
 - *Folgerungssätze* (unwichtig)
 - *Temporalsätze* beschreiben eine Vorzeitigkeit (*nachdem, als, seit, seitdem, sobald, sowie, wenn, kaum daß*) oder eine Nachzeitigkeit (*bis, bevor, ehe*) oder eine Gleichzeitigkeit (*während, indem, indes, solange, sobald, sowie, sooft, als, wenn*).
 - *Konditionalsätze* beschreiben eine Voraussetzung oder ein Urteilsgewicht (*wenn, wofern, sofern, falls* oder uneingeleiteter Nebensatz), eine Ausnahmeporaussetzung (*außer wenn, es sei denn*) oder ein Verhältnisgesetz (*je...um so, je...desto*).
 - *Relevanz/Irrelevanz-Sätze* erklären die Irrelevanz einer Sachverhaltsklasse (*was/wie/wo/...auch immer*), einer Sachverhaltsalternative (*ob...oder*) oder eines einzelnen Sachverhalts (*auch wenn, selbst wenn*) oder die Relevanz einer Sachverhaltsalternative (*je nachdem ob*) oder einer Sachverhaltsskala (*je nachdem wer/wie/wo/...*).

- *Finalsätze* beschreiben eine Motivation (Zweck), die kausal oder konditional dargestellt sein kann (**damit**, **daß**, **auf daß**, **um...zu**), oder eine Eignung (Zweckmäßigkeit)
- *Modalsätze* enthalten eine Handlungsausführung (**indem**, **dadurch daß**, **so daß**) eine Handlungsbegleitung (**wobei**) oder einen fehlenden Begleitumstand (**ohne zu** mit Infinitiv oder **ohne daß**).
- *Konfrontationssätze* beinhalten Gegenüberstellungen und zwar gegensätzliche (**während**, **wogegen**, **wohingegen**), ersetzende (**statt**) oder ausgrenzende (**außer daß**, **außer zu**+Infinitiv).
- *Präzisierungssätze* ergänzen eine vorherige vollständige Aussage, indem sie zusätzliche Information liefern und zwar einen Vorbehalt (**außer daß**, **nur daß**), eine Eingrenzung (**was...betrifft**, **insofern**, **dafür daß**) oder eine Begleitung (**wobei**).

Es dürfte klar sein, daß angesichts der Vielzahl von Formen und Bedeutungen eine vollständige Behandlung deutscher Nebensatzstrukturen so gut wie ausgeschlossen ist. Denn erstens sind die Unterschiede zur Klassifizierung oft nur subtiler Art (die äußeren Formen fallen zu einem großen Teil übereinander), und zweitens ist auch nicht ersichtlich, wie ein solches Analyseergebnis sinnvoll weiterverarbeitet werden könnte. Es ist also nötig, zumindest einen großen Teil der möglichen Nebensatzkonstruktionen aus der Analyse herauszulassen.

Eine ausreichend große syntaktische Kompetenz ist für die allermeisten Fälle aber schon erreicht, wenn Relativsätze, sowie Final-, Konsekutiv- und Kausalsätze (für Erklärungen) und Konditionalsätze (für Regeln) analysiert werden können. Dabei ist auch eine Einschränkung auf die gängigsten Erscheinungsformen zu verschmerzen.

2.6.3 Satzzeit

Als die Standardzeit wird das Präsens betrachtet. Die Zeit eines Satzes spiegelt sich vollständig und einzig im Prädikat wider, das in seiner Form verändert wird. Insbesondere verändert sich mit der Satzzeit oft auch die Anzahl von Teilen, aus denen das Prädikat besteht. Eine besondere Stellung nehmen Befehlssätze ein: sie sind zeitlos.

Zur Berücksichtigung der Zeit im Satz reicht es unter dieser Voraussetzung aus, die Zeit des Prädikates zu ermitteln und in der Ausgabe anzugeben. Dies geschieht für jeden Nebensatz einzeln, wobei gegenseitige Abhängigkeiten nicht unbedingt berücksichtigt zu werden brauchen. Die Angabe entfällt bei Befehlssätzen.

2.6.4 Satzmodus

Als der Standardmodus wird der Indikativ betrachtet. Eine Modusveränderung kann sich stark oder schwach auswirken.

Bei schwacher Auswirkung verändern lediglich die Einzelteile des Prädikates ihren Modus, sie bleiben jedoch in Anzahl und Stellung gleich. Dieser Fall tritt auf bei korrekter¹⁸

¹⁸„korrekt“ bedeutet hier die Verwendung der grammatikalischen Formen des Konjunktivs ohne Umschreibung mit *mögen* oder *würden*.

Verwendung des Konjunktiv I oder II, z.B. für die indirekte Rede oder als einfachem Irrealis. Eine wichtige Sonderstellung nimmt die Darstellung von Definitionen mittels des Konjunktivs **sei** ein, die vor allem im wissenschaftlichen Umfeld sehr gebräuchlich ist.

Bei starker Auswirkung tritt zusätzlich eine Veränderung der Anzahl oder Stellung der Prädikatsteile auf. Die wichtigsten dieser Fälle sind erstens die Stirnstellung beim Befehlssatz und zweitens die Umschreibung des Konjunktiv II mit **würden**. Bei Befehlssätzen ergibt sich die zusätzliche Modifikation, daß sie kein Subjekt aufweisen.

In beiden Fällen erschwert sich nur das Finden und die Analyse des Prädikats; der Rest des Satzes bleibt, ausgenommen bei Befehlssätzen, unbeeinflußt. Es genügt also auch hier meist, den Modus zu ermitteln und in der Ausgabe anzugeben. Bei Befehlssätzen muß darüberhinaus die Suche nach einem Subjekt unterbleiben.

2.6.5 Satzgenus

Die Standardaussageweise ist das Aktiv¹⁹. Im Prinzip kann ein Satz in den meisten Fällen vom Passiv ins Aktiv übertragen werden. Von dieser Übertragung wird allerdings nicht nur das Prädikat betroffen.

Manche Rollen wechseln ihre äußere Form, so wird zum Beispiel die Rolle des Agenten im Passiv als Präpositionalobjekt vorgefunden, während sie im Aktiv vom Subjekt (im Nominativ) gefüllt wird. Durch die Darstellung als Objekt wird es möglich, im Passiv den Agenten auszulassen (*Im Lokal wird getanzt.*), so daß bei der Übertragung ein Subjekt (und damit ein Agent) künstlich erfunden werden muß (*Man tanzt im Lokal.*). In manchen Fällen ist die Einführung eines *virtuellen Substantivs* mit der Bedeutung "irgendjemand oder irgendetwas" zur Darstellung dieses Subjekts notwendig. In der Tat besteht die Motivation zur Verwendung des Passiv gerade darin, die Angabe eines Agenten zu umgehen²⁰

Auch der Sinn kann sich bei diesem Übergang verändern: Aus dem Satz

Kein Programm wird von mehreren Informatikern gemacht.

der besagt, daß Informatiker beim Programmieren nicht zusammenarbeiten, wird im Aktiv

Mehrere Informatiker machen kein Programm.

was besagt, daß es einige faule oder enthaltsame Informatiker gibt. Kurzum: Die mechanische Umwandlung von Passiv nach Aktiv hat einige Tücken.

2.6.6 Aussagesatz

Ein Aussagesatz liegt vor, wenn kein Frage- oder Befehlssatz vorliegt. Eine Fallschablone beschreibt eine Menge von Aussagesätzen. Alle anderen Satzarten werden bei der Analyse normalisiert und auf die Form des Aussagesatzes zurückgeführt; zusätzliche Angaben ermöglichen die korrekte Interpretation des Ergebnisses. In (D1088-1141) finden sich detaillierte Beschreibungen aller üblichen Satzbaupläne für einfache deutsche Aussagesätze.

¹⁹Im Deutschen wird heute etwa 93% Aktiv, 5% Vorgangspassiv und 2% Zustandspassiv verwendet (D294).

²⁰Etwa bei Große Intelligenz ist ihm gegeben. Wer war es denn nun: Gott? Die Natur? Oder vielleicht Mama und Papa? Man braucht keine Stellung zu nehmen — dem Passiv sei gedankt.

2.6.7 W-Frage

Eine W-Frage liegt vor, wenn die erste Stelle im Satz von einem Frageglied besetzt ist. Der Satz hat ansonsten die Form eines Aussagesatzes; allerdings ist die Stellung der Satzteile eventuell anders als normal, da das erfragte Glied vorn stehen muß. Dabei besetzt das Frageglied den Platz genau einer Rolle. Welche Rolle dies ist, hängt von dem Frageglied ab; dieses kann sein

- ein einzelnes Fragewort (**Wie, Wo, Wann?**). Dies ist die klassische W-Frage. Für jedes solche Fragewort kommt nur eine kleine Zahl von Rollen in Frage.
- ein Fragepronomen mit einem Restsatzglied (**Welches Buch?**). Dabei ersetzt das Fragepronomen quasi einen Artikel und das Restsatzglied ist ein Substantivglied, dem lediglich dieser Artikel entfernt wurde.
- eine Frageformel, zum Beispiel aus Präposition, Fragewort und manchmal einem Substantiv- oder Partikelanhang (**Von wo? Von wo aus? Auf welche Weise? In wessen Sinn?**).

2.6.8 Ja/Nein-Frage

Eine Ja/Nein-Frage liegt vor, wenn erstens das Prädikat an der ersten Stelle im Satz steht und es zweitens keine Imperativform hat. Die Form, auf die eine Ja/Nein-Frage zurückgeführt wird, ist die exakt passende Aussage, die eine Beantwortung der Frage mit **Ja** erlaubt. Also: die Frage **Ist Informatik schwierig?** kann zumindest dann mit **Ja** beantwortet werden, wenn die exakt passende Aussage **Informatik ist schwierig** als gegeben vorliegt und genau auf diese Aussage wird die Frage abgebildet.

2.6.9 Befehlssatz

Ein Befehlssatz liegt vor, wenn erstens das Prädikat an der ersten Stelle im Satz steht und es zweitens Imperativform hat. Ein Befehlssatz hat kein Subjekt, die Rolle des Agenten bleibt also unbesetzt. Eine Umformung zur Abbildung auf einen Aussagesatz ist nicht notwendig.

2.7 Besondere Themen

2.7.1 Ellipsen

Bei der *Einsparung von Redeteilen (Ellipse)* fallen Satzteile weg, ohne daß der Satz dadurch unverständlich wird (D1142-1144). In der extremen Form entstehen dabei Einwortsätze, meist mit feststehender Bedeutung: **Hilfe! Feuer! Guten Tag. Willkommen.**

In abgeschwächter Form bleiben weitere Satzteile stehen: **Laß das. Wird erledigt.** Man kann folgende wiederkehrenden Formen der Ellipse erkennen:

- Bei der *Einsparung von Redeteilen in gleichkonstruierten Sätzen* werden gemeinsame Redeteile ausgelassen: **Ich gehe ins Wohnzimmer und du ins Bett**. Diese Zusammenfassung kann mit oder ohne Konjunktion erfolgen. Es darf in einer Reihe von Fällen jedoch nicht ausgelassen werden. So immer dann, wenn die auszulassenden Teile nicht kongruent sind (wechselnder Fall bei Pronomen²¹, unterschiedliche Zahl bei Hilfsverben²² oder verschiedene Bedeutung des Teils an den Verwendungsstellen²³). Dennoch findet sich oft auch in diesen Fällen fälschlich eine Auslassung, im Falle des Bedeutungsunterschieds oft bewußt angewandt, um eine komische Wirkung zu erzielen.
- Die *Einsparung von Attributen* umfaßt Adjektive vor Substantivzusammensetzungen (**Französisches Essen und Trinken**), Adverbien vor Verbzusammensetzungen (**Französisch essen und trinken**) und gleiche Pronomina (**Sein Schnarchen und Röcheln verhinderte meinen Schlaf**). Auch hier kommen in ähnlicher Weise Regelverletzungen vor.
- Genau umgekehrt erfolgt auch eine *Einsparung von Attributiertem*: **Französisches und deutsches Essen**. **Französisch oder deutsch essen**. **Sein und ihr Schnarchen verhinderte meinen Schlaf**.
- Schließlich gibt es die *Teilworteinsparung* bei sich wiederholenden Teilen von Substantiven oder Verben:

Groß- und Kleinbauern müssen an Privatzugang und -fahrt ihres Hofes ein Hinweisschild ankleben oder -nageln, um Besucher ein- oder abzuweisen.

Diese Konstruktion betrifft also Sinnanteile von zusammengesetzten Substantiven sowie Präfixe von Verben, wird jedoch auch oft fälschlich z.B. für Suffixe von Substantiven gebraucht: **Wahr- und Falschheit, Mutter- und Vaterschaft**.

2.7.2 Negation

Die elementarste logische Operation, die Verneinung, ist verständlicherweise ein sprachlich recht vielgestaltiges Phänomen und birgt erhebliche Schwierigkeiten, was die Erkennung und korrekte Interpretation in der deutschen Sprache angeht.

Negierende Elemente können sein (D1148-1155)

- Worte mit negierender Bedeutung wie **bestreiten, verfehlen**
- Wendungen mit negativer oder negierender Bedeutung wie **sich den Teufel darum kümmern**
- *Negationspréfixe* an Wörtern wie **un-, a-, des-, in-, miß-, nicht-**
- *Negationswörter* wie die Negationspronomen **keiner, niemand, nichts** und die Negationspartikel **nicht, nie, nirgend...**, **keines...**
- das Satzäquivalent **Nein**.

²¹Jene, die uns (uns im Akkusativ) liebten, ja Bonbons geschenkt (uns im Dativ) hatten.

²²Es wurde geschossen und (es wurden) Häuser abgebrannt.

²³Ich ging schwimmen und dabei meine Uhr kaputt.

Negative Wendungen sind für die Verarbeitung natürlicher Sprache sowieso kaum von Bedeutung und können ansonsten, ebenso wie Worte mit negativer oder negierender Bedeutung, ganz im Rahmen der sonstigen Fähigkeiten von **SARA** behandelt werden, indem man nämlich passende Fallschablonen vorsieht. Problematischer stellen sich die Negationswörter dar:

- Die *Negationspronomen* **keiner**, **niemand** und **nichts** (D1149) können an der Position eines Subjektes oder Objekts stehen (**Niemand ist zu sehen. Ich sehe niemanden.**) und haben dort auch eine entsprechende Bedeutung: das semantische Subjekt oder Objekt wird als nicht vorhanden angegeben.
Sie werden auch wie Attribute gebraucht und negieren dann das Substantiv mit dem sie stehen: **Niemand anderes kann das erledigen, aber mir fällt nichts Gescheites ein und ich habe keine Lust.**
- Die *Negationspartikel* **nicht**, **nie** (und **nimmer**), **niemals**, **nirgends**, **nirgendwo**, **nirgendwoher**, **nirgendwohin**, **keinesfalls**, **keineswegs** (D1150) können ebenfalls allein eine syntaktische Stelle ausfüllen. Sie negieren dort die Gesamtheit aller Bestimmungen, die sonst dort gestanden haben könnten (**Ich saufe nie/manchmal/von 10 bis 12, und zwar nirgendwo/zuhause/unter dem Bett.**). Es ist aber auch möglich, daß sie zu einem Glied hinzutreten (**Er säuft nie viel.**); sie stehen dann davor und negieren genau dieses Glied, das z.B. ein Satzadjektiv, ein Präpositionalgefüge oder ein Satzpartikel sein kann.
- Alle diese Negationswörter können zusätzlich verstärkt werden²⁴ durch die Wörter **gar**, **überhaupt** und **absolut** (in der Jugendsprache auch **echt**) oder zeitlich modifiziert durch **noch** oder **mehr** (D1150), z.B. **Ich freue mich echt absolut überhaupt gar nicht mehr.**

2.7.3 Zahlen und Zahlwörter

Zahlen nehmen für Computeranwendungen naturgemäß eine besonders wichtige Stellung ein. Bei der Betrachtung der Zahlwörter kommt man zu der überraschenden Feststellung, daß der Wortschatz offenbar unendlich groß ist. Selbst wenn man nur einen begrenzten Zahlbereich benötigt, empfiehlt es sich, eine systematische Behandlung von Zahlen zu ermöglichen, da das Eintragen aller Zahlen ins Wörterbuch unrealistisch ist.

Die *Kardinalzahlen* (*Grundzahlen*) zählen zu den Adjektiven (**Elf kleine Negerlein**). Sie werden allerdings nur schwach gebeugt: einzig die (seltene) Genitivform weicht vom Nominativ ab (**Dies ist die Geschichte zwölfer kleiner Negerlein.**). Diese Tatsache macht es nicht allzu unangenehm, die Eingabe von Zahlwörtern in Ziffern zu verlangen. Dies ermöglicht eine einfache algorithmische Behandlung und wird vom Endbenutzer gut akzeptiert werden, da eine Eingabe in Buchstaben ohnehin aufwendiger wäre. Um die wichtigsten Ausnahmefälle abzudecken, sollten die Zahlwörter **eins** bis **zwölf** ins Wörterbuch aufgenommen werden. Die genaue Behandlung der Beugung ist dabei allerdings kompliziert (D457-459).

Die Deutung von Kardinalzahlen kann tatsächlich in der Regel als Adjektiv bei einem Substantiv oder als Satzadjektiv oder Adverb erfolgen (**Das Ergebnis lautet 11.**). Eine besonders wichtige Anwendung von Kardinalzahlen ist die Angabe von Zeit (D461-462).

²⁴Na, wenn *das* kein glänzendes Beispiel für den logischen Aufbau der Sprache ist. . .

Die *Ordinalzahlen (Ordnungszahlen)* (D464) haben einen sehr ausgeprägt adjektivischen Charakter, so daß ihre Behandlung als Adjektive keine Schwierigkeiten aufwirft; sie können auch genau wie die Adjektive substantiviert werden. Etwas erschwert ist hier allerdings die Eingabe mit Ziffern, da Ordinalzahlen dann mit einem Abschlußpunkt gekennzeichnet werden, der nicht mit dem Satzendeypunkt verwechselt werden darf. Analog zur Lage bei den Kardinalzahlen sollte man auch hier **erster** bis **zwölfter** extra ins Wörterbuch aufnehmen.

Die *Vervielfältigungszahlwörter* (D467) entstehen durch Anhängen der Nachsilbe **-fach** an Kardinalzahlen; statt **zweifach** wird auch **doppelt** gebraucht. Eine Mögliche Kurzschreibweise in der Zifferndarstellung ist ein angehängtes **x**. Die Vervielfachung wird oft substantiviert (**das ...fache**) oder umschrieben (**...mal so oft/so viel**).

Bruchzahlen (D465-466) kommen ebenfalls alleinstehend als Substantive oder adjektivisch verwendet vor. Meist sind Bruchzahlen heute (zumal bei Computeranwendungen) von den Dezimalzahlen verdrängt, so daß nur wenige und ganz bestimmte Bruchzahlen vorkommen (halb/Hälfte, drittel, viertel, zweidrittel, dreiviertel, ...). Es kann deshalb wohl von einer algorithmischen Handhabung abgesehen werden, indem man die nötigen Einzelfälle ins Wörterbuch einträgt.

Dezimalzahlen müssen sinnvollerweise auch behandelt werden. Sie verhalten sich grammatikalisch genau wie die Kardinalzahlen, können aber sinnvollerweise ausschließlich in Ziffernform verstanden werden; dabei ist sowohl ein Punkt als auch ein Komma als Dezimalzeichen zu erwarten. Eventuell kann man hier auch die halblogarithmische Schreibweise zulassen.

unbestimmte Zahladjektive (D469-471) sind zum Beispiel **alle**, **andere**, **einige**, **einzelne**, **etliche**, **ganze**, **gesamte**, **gewisse**, **manche**, **mehrere**, **ungezählte**, **übrige**, **vereinzelte**, **verschiedene**, **viele**, **weitere**, **wenige**, **zahllose**, **zahlreiche**. Der genaue Wert, der von ihnen bezeichnet wird, ist im Allgemeinen nicht feststellbar, kann aber manchmal aus dem Zusammenhang hervorgehen (vor allem bei **alle**, **übrige** und **andere**).

Kapitel 3

Analyse mit Fallschablonen

*Es ist nur natürlich,
sich eine Methode vorzunehmen
und sie auszuprobieren.
Wenn es schiefgeht, gib es offen zu
und versuche etwas anderes.*

Franklin D. Roosevelt

In diesem Kapitel werden einige allgemeine Aspekte von **SARA** besprochen, nämlich der Begriff der Fallschablone und des Fallschablonenzerteilers und im Zusammenhang damit ein Großteil der Terminologie, die bei der Beschreibung der Architektur und Implementation verwendet wird. Dann die Philosophie der sogenannten toleranten Zerteilung, die bei **SARA** zur Anwendung kommt, die Menge der Rollen, aus denen für die Beschreibung von Fallschablonen typischerweise ausgewählt wird und schließlich das Verfahren des Chart-Parsing, das dem Zerteiler zugrundeliegt.

3.1 Fallschablonen und Zerteiler

3.1.1 Was ist eine Fallschablone ?

1. Eine *Fallschablone* ist ein Datenobjekt, das eine Klasse von Äußerungen in einer natürlichen Sprache beschreibt.

Das kann zum Beispiel so aussehen:

```
Fallschablone 'geben' :  
Rolle wer_gibt          ein_Mensch  
Rolle wem_wird_gegeben ein_Mensch  
Rolle was_wird_gegeben physikalisches_Objekt
```

Diese Beschreibung erfasst beispielsweise alle folgenden Sätze:

```
Ich gebe dir ein Buch.  
Du gibst Sara das Buch.  
Sara gibt das Buch ihrer Oma.  
Die Oma gibt das Buch an ihre Vermieterin.
```

2. Zur Beschreibung dieser Satzklasse enthält die Fallschablone einen *Kopf*, sozusagen der Kern der Äußerung (z.B. **geben**). Dieser Kopf ist das eindeutig identifizierende Merkmal der Fallschablone. Der Kopf der Fallschablone definiert eine syntaktische und eine semantische Einschränkung der “erlaubten” Äußerungen (z.B. “Vollständige deutsche Sätze mit **geben** als Hauptverb”).
3. Außerdem enthält eine Fallschablone eine Menge von *Fallschablonenrollen*. Eine Fallschablonenrolle (z.B. **Rolle wer_gibt ein_Mensch**) besteht aus einer *Rolle* (**wer_gibt**) und einer Einschränkung für ihren *Rollenfüller* (**ein_Mensch**).
4. Eine *Rolle* hat einen sie eindeutig definierenden Namen (z.B. **wer_gibt**) und beschreibt eine syntaktische Einschränkung, d.h. eine Menge von grammatikalischen Konstruktionen (z.B. “Nominalgruppe im Nominativ”). Normalerweise vergibt man allgemeinverwendbare, d.h. in verschiedenen Fallschablonen zu gebrauchende Rollennamen; so heißt das Subjekt als Rolle typischerweise **Agent**.
5. Ein *Rollenfüller* (oft einfach *Füller* genannt) ist ein Sinnobjekt der betrachteten Sprache, z.B. ein Substantiv. Eine Einschränkung an einen Rollenfüller zu machen, wie es in einer Fallschablone für jede Rolle geschieht, bedeutet also, z.B. (bei dem Füller **ein_Mensch**) zu sagen “Hier ist nicht jedes beliebige Substantiv erlaubt, sondern nur solche, die einen Menschen bezeichnen.” Solche Einschränkungen werden auch *Konzepte* genannt. Mögliche Arten von Rollenfüllern sind Substantive (z.B. **Mensch**), Lokaladverben (zur Ortsangabe, z.B. **oben**), Modaladverben (zur Angabe der Art und Weise, z.B. **lässig**), Temporaladverben (zur Angabe der Zeit, z.B. **gestern**) und ganze Sätze (z.B. Nebensätze mit **weil**, zur Angabe eines Grundes).
6. Neben der bisher beschriebenen Form von Fallschablonen für ganze Sätze, den sogenannten *verbalen Fallschablonen*, kann man auch Fallschablonen angeben, deren Kopf ein Substantiv ist, die sogenannten *nominalen Fallschablonen*. Manche Rollen lassen sich nämlich nicht nur Sätzen zuordnen, sondern auch einzelnen Substantiven. Ein Beispiel hierfür ist die Rolle **Ort**: **Der Garten hinter dem Haus**. Es ist stets möglich, dort, wo als Füller ein Substantiv erwartet wird, statt eines Substantivs eine Instanziierung (s. unten) einer nominalen Fallschablone als Rollenfüller zu verwenden.

3.1.2 Was ist der Zweck von Fallschablonen ?

Man erhält als Ausgabe eines Analyseprozesses mit Fallschablonen eine Fallschablone, deren Rollen mit irgendwelchen Teilen des Eingabesatzes ausgefüllt sind, eine sogenannte *Instanziierung*. Diese ist nun verhältnismäßig leicht weiterzuverarbeiten: Für die weitere Verarbeitung kann man sich beispielsweise zu jeder Fallschablone eine Prozedur zugeordnet denken, deren Parameter die Rollenfüller sind. Von jedem Parameter liegt die Bedeutung bereits fest, nur der Wert kann sich von Mal zu Mal verändern. Außerdem erlauben Fallschablonen, Wortmehrdeutigkeiten anhand der von jeder Rolle definierten Kombination von syntaktischen und semantischen Einschränkungen aufzulösen.

Der Zweck von Fallschablonen bei der Verarbeitung natürlicher Sprache besteht also darin, eine Zerlegung einer (Menge von) Äußerung(en) in solche Teile zu beschreiben, die eine gewisse, vorher bekannte Beziehung zur Gesamtäußerung haben. Dabei wird gleichsam von alleine eine Zwischendarstellung definiert, die gut von nachfolgenden Verarbeitungsstufen

(z.B. Kontextinterpretation) weiterverarbeitet werden kann und leicht die Beantwortung bestimmter Fragen (eben nach den einzelnen Rollen) erlaubt.

3.1.3 Was ist ein Zerteiler ?

Der Begriff des Zerteilers kommt eigentlich von der Theorie der formalen Sprachen her.¹ Eine solche Sprache wird zum Beispiel durch ein Produktionensystem beschrieben, also ein System von Regeln, durch deren Anwendung man aus einem einzigen Startsymbol jede in der Sprache mögliche Äußerung erhalten kann (s. [GooWai84]). Ein Zerteiler ist in diesem Zusammenhang ein Programm, das eine gegebene Äußerung auf eine Folge von Regelanwendungen zurückführt, was die innere Struktur der Äußerung widerspiegelt. Je nach Art der Sprache kann dieser Prozess ein oder mehrere Ergebnisse haben und der Prozess selbst kann direkt zum Ziel führen oder auf Rücksetzen angewiesen sein.

Im Bereich der Verarbeitung natürlicher Sprache handelt es sich bei einem Zerteiler um ein Programm, das eine Analyse von natürlichsprachlichen Äußerungen vornehmen soll. Wie dies vor sich geht und welche Form die Ausgaben haben, hängt jedoch stark von der Art des Systems ab. Bei SARA wird zur Analyse ein Chart-Parser verwendet (s. Abschnitt 3.4) und die Ausgaben haben die Form von Instanziierungen von Fallschablonen. Der Zerteiler hat die Aufgabe die eingegebene Äußerung anhand der Fallschablonen in eine oder mehrere Instanziierungen umzusetzen.

3.1.4 Was ist ein Fallschablonenzerteiler ?

Ein Fallschablonenzerteiler ist ein Zerteiler, der neben dem Eingabesatz noch eine zweite Datenmenge benötigt: die Fallschablonen. Ob man diese Fallschablonen nun als Teil des Zerteilers betrachtet, oder als Teil der Eingabe an ihn, ist eine reine Geschmackssache. Ein Fallschablonenzerteiler versucht, den Eingabesatz auf eine dieser Fallschablonen abzubilden, das heißt, die Eingabe wird zu Wortgruppen zusammengefasst und jede dieser Wortgruppen wird einer Rolle oder dem Kopf der Fallschablone zugeordnet. Dabei können sich Mehrdeutigkeiten ergeben, weil sich dem gefundenen Verb mehr als eine Fallschablone zuordnen läßt, oder weil sich manche der Wortgruppen als Füller für mehr als eine der Rollen in der jeweils betrachteten Fallschablone eignen. Das Analyseergebnis eines Fallschablonenzerteilers ist also im Allgemeinen eine Menge von Instanziierungen und verändert sich nicht nur mit veränderter Eingabe, sondern auch mit veränderter Fallschablonenmenge. Gerade die letztere Eigenschaft ist es, die Fallschablonensysteme für die Verarbeitung natürlicher Sprache so interessant macht: Durch geeignete Wahl von Fallschablonen kann das System auf einem bestimmten, engumgrenzten Gebiet sprachlicher Äußerungen eine recht große Sprachkompetenz haben und diese Wahl von Fallschablonen kann, da sie mit dem eigentlichen Programm nicht fest verbunden ist, leicht verändert werden. Das macht Fallschablonensysteme recht anwendungsfreundlich.

3.2 Tolerante Zerteilung

Dem Konzept der *toleranten Zerteilung* (*relaxed parsing*²) liegt folgende Haltung zugrunde:

¹Zerteiler heißt auf englisch *parser*, was aus dem Lateinischen stammt: *pars orationis* bedeutet "Teil der Rede" oder auch "Teil der Sprache".

²*relaxed*, engl.: entspannt, freundlich, locker, mild

Für ein tatsächlich brauchbares System zur Verarbeitung natürlicher Sprache kommt es darauf an, auch solche Eingaben noch “vernünftig” analysieren zu können, die nicht exakt den Grundsätzen der Sprachgrammatik entsprechen oder in der Systemgrammatik nicht vorgesehen sind.

Dies läßt sich zum Teil dadurch erreichen, daß man die Genauigkeit der Analyse etwas schleifen läßt, das heißt einige Tests, die möglich wären, nicht oder nicht vollständig durchführt. Im Falle von Deutsch bezieht sich das zum Beispiel auf die Überprüfung der verschiedenen Kongruenzen. Dadurch werden oft auch inkorrekte³ Eingaben akzeptiert — das System wird *robust*.

Der Preis, den man dafür zahlen muß, ist der einer erhöhten Zahl von Mehrdeutigkeiten. Es werden für mehrdeutige Eingaben noch mehr mögliche Bedeutungen aufgefunden, es werden manche eindeutigen Eingaben mehrdeutig und es werden für manche eigentlich abzuweisende Eingaben Interpretationen “erfunden”.

Einige dieser Deutungen sind nun falsch und man versucht die Trennung zwischen richtigen und falschen Interpretationen anhand der Ergebnisse vorzunehmen⁴, also aus dem syntaktischen Analyseprozeß hinaus- und in die Auswertung der Resultate hineinzunehmen. Dies läßt sich zum Teil in den Zerteiler selbst vorverlagern, indem man intern eine Gütebewertung der (Teil)Lösungen einführt, die beispielsweise besagt, daß eine korrekte Lösung jeweils allen solchen Lösungen vorzuziehen ist, die die Toleranz ausgenutzt haben.

Tolerantes Zerteilen bedeutet also, die Anwendung harten Wissens in syntaktischer Hinsicht teilweise aufzugeben zugunsten einer großzügig akzeptierenden Haltung (“Nehmen wir mal an, es war *so* gemeint.”), um die Analyse robust zu machen.

Auf das Zeitverhalten ergeben sich zwei gegenläufige Auswirkungen: Einerseits spart das tolerante Zerteilen etwas Zeit, indem es bestimmte Prüfungen nicht durchführt. Andererseits erfordern die sich dabei ergebenden zusätzlichen Mehrdeutigkeiten gegebenenfalls auch zusätzliche Verarbeitungsschritte, was (möglicherweise erheblich) Zeit kostet.

SARA verwendet an folgenden Stellen die Philosophie des toleranten Zerteilens

1. Die Kongruenzprüfung bei Adjektiven entfällt völlig. Es wird nur zwischen überhaupt gebeugten und ungebeugten Formen unterschieden. Diese Maßnahme verkleinert auch das Wörterbuch.
2. Folgen von (Modal-)Adverbien und Adjektiven werden auch nicht auf ihre semantische Plausibilität geprüft, sondern nur aufgereiht als Attribute eines Substantivs angegeben.
3. Die Stellung von Wortgruppen unterliegt nur sehr wenigen Einschränkungen, die allerdings unvermeidlich sind, z.B. um Frage- und Imperativsätze von Aussagesätzen unterscheiden zu können.

³Dies ist ein relativer Begriff. Korrektheit läßt sich erst auf der pragmatischen Ebene sinnvoll festlegen. Hier ist eine Inkorrektheit in Bezug auf eine strenge syntaktische Analyse gemeint.

⁴Angesichts des geringen Weltwissens, das Systeme zur Verarbeitung natürlicher Sprache heute haben, sind falsche Interpretationen von Eingaben nicht zu vermeiden. Der Ansatz der milden Zerteilung ist also insofern geschickt, da eine Plausibilitätsprüfung der Analyseergebnisse ohnehin notwendig ist. Eine *nachträgliche* Prüfung ist auch aus dem Grunde sinnvoll, weil man damit das nicht sachgebundene syntaktische Wissen besser von dem sachgebundenen semantischen trennt.

4. Die Zuordnung von abgetrennten Präfixen zu Verben unterliegt keinen Beschränkungen. *SARA* könnte durch Änderungen bei der Formenerzeugung des Wörterbuchs leicht noch toleranter gemacht werden.

3.3 Rollen

In diesem Abschnitt werden nun die typischen Rollen zusammengestellt, die zur Bildung der Fallschablonen verwendet werden.⁵ Diese Sammlung kann im Bedarfsfall natürlich durchaus verändert oder erweitert werden.

Die Aufstellung hat folgende Form: Zuerst steht immer der Name der Rolle; diese Namen sind zudem teilweise in Gruppen sortiert. Hinter einem Trennstrich folgt die Aufzählung der grammatikalischen Konstruktionen, die für einen Füller dieser Rolle vorkommen können. Dabei werden die grammatikalischen Fälle Nominativ, Genitiv, Dativ und Akkusativ in der Regel mit Nom, Gen, Dat und Akk abgekürzt. Eine Präpositionalkonstruktion wird (zum Beispiel) in der Form “für+Akk (für wen?)” geschrieben; das heißt die Rolle kann gefüllt werden von der Präposition *für*, gefolgt von einem Glied im Akkusativ und wird erfragt mit der Frageformel “für wen?”⁶. In Klammern folgt jeweils die Angabe, wie die Rolle erfragt werden kann. Dabei kann die angegebene unspezifizierte Frage auch durch eine spezifizierte (mit *Welcher/Welche/Welches*) ersetzt werden. Andere grammatikalische Konstruktionen sind gegebenenfalls ausformuliert beschrieben.

Die Rollen sind:

- Personen oder wie Personen beteiligte Objekte
 - (1) *Agent* — im aktiven Satz Nom (*Wer?, Was?*), im passiven Satz von+Dat (*Von wem?*) oder durch+Akk (*Durch wen?*), bei nominalen Instanzen von aus Verben abgeleiteten Substantiven Gen (*Wessen?*)
 - (2) *Koagent* — mit+Dat (*Mit wem?*)
 - (3) *Patient* — Dat(*Wem?*)
 - (4) *Nutznießer* — Dat (*Wem?*), für+Akk (*Für wen?*), an+Akk (*An wen?*)
 - (5) *Eigentümer* — Gen (*Wessen?*) meist vorangestellt, oder von+Dat (*Wessen?*) stets nachgestellt
- Allgemeine Rollen⁷
 - (6) *Thema oder Betroffenes (affiziertes Objekt)* — Akk(*Wen?*)
 - (7) *Genitiv* — Gen (*Wessen?*)⁸

⁵Für die genaueren grammatikalischen Daten sei verwiesen auf den Abschnitt 2.5 über die Satzglieder und dort vor allem:

die Absätze über Objekte (ab Seite 52) und über adverbial gebrauchte Satzglieder (ab Seite 53), den Abschnitt 2.1.6 über die Konjunktionen (ab Seite 35) und den Abschnitt 2.6.2 über Nebensätze (ab Seite 58).

⁶Diese Abkürzungen sollen verhindern, daß die Aufzählung durch ständige langatmige Wiederholungen unlesbar wird.

⁷Die Tatsache, daß sich diese Rollen an einem grammatikalischen Fall festmachen, sollte Signal genug sein, sie so sparsam wie irgend möglich einzusetzen.

⁸Diese Konstruktion, einem grammatikalischen Fall direkt eine Rolle zu vergeben, ist zwar sehr unbefriedigend, läßt sich jedoch leider kaum umgehen. Die Konstruktionen mit dem Genitiv sind nämlich nicht in eine kleine Zahl von Rollenbeziehungen einzuordnen.

- (8) *Definition* — Nom(Was?) Gleichsetzungsnominativ.
- Rollen von Sachen (physikalischen Sachen oder Sachen im übertragenen Sinn)
 - (9) *Attribut* — mit+Dat (Mit was?, Mit wem?)
 - (10) *Material* — aus+Dat (Aus was?, Woraus?)
 - (11) *Werkzeug* — mit+Dat (Mit was?, Womit?), durch+Akk (Durch was?, Wodurch?), mittels+Gen (Womit?) oder (fälschlich) mittels+Dat (Womit?)
 - (12) *Resultat (effizientes Objekt)* — Akk(Was?)
 - Rollen des Raums
 - (13) *Ort* — eine der Präpositionen in, auf, vor, hinter, neben, bei, an, rechts, links, rechts von, links von mit dem Dativ, Frage unspezifiziert mit Wo? oder teilspezifiziert mit Worin?, Worauf?, Wovor?, Wohinter?, Woneben?, Wobei?, Woran?, Rechts wovon?, Links wovon?
 - (14) *Startpunkt* — von+Dat (Von wo?, Von wo aus?, Von wo ab?, Von wo her?)
 - (15) *Zielpunkt oder Richtung* — selbe Konstruktion wie beim Ort (Frage jedoch immer Wohin?) oder zu+Dat (Zu wem?, Zu was?, Wohin?)
 - (16) *Bahn* — über+Akk (Worüber?, Wo entlang?), durch+Akk (Wo durch?, Wo entlang?), entlang+Gen (Wo entlang? Entlang wessen?), entlang+Dat (Wo entlang?)
 - (17) *Erstreckung* — bis+Dat (Wohin?, Bis wohin?), entlang+Gen (Entlang wessen?, Wo entlang?, Wohin?, Wo?), entlang+Dat (Entlang wessen?, Wo entlang?, Wohin?, Wo?)
 - Rollen der Zeit
 - (18) *Zeitpunkt* — Zeitadverbien, um mit Uhrzeitangabe, am mit Tagesangabe, im mit Monatsangabe, Nebensätze mit nachdem, bevor, während, sobald, sowie andere Konstruktionen, wie im Jahr..., relative Angaben mit in etc. Frage immer Wann?
 - (19) *Beginn* — selbe Konstruktion wie beim Zeitpunkt, jedoch mit den Präpositionen seit, ab, von...an und Nebensätze nur mit nachdem oder sobald. Fragen Wann? oder Ab wann? Seit wann?, Von wann an?
 - (20) *Ende* — dito, mit der Präposition bis, oder Nebensatz mit bevor. Fragen Wann?, Bis wann?
 - (21) *Dauer* — Zeitintervallangabe mittels zweier Zeitpunktsangaben mit einer Formulierung wie von...bis... oder rein quantitatives Zeitintervall (z.B. 10 Minuten, eventuell ergänzt durch lang o.ä.) oder Nebensatz mit während. Frage immer Wie lange? oder Wielange?
 - Rollen der Art und Weise (im weitesten Sinne)
 - (22) *Art und Weise* — Partizipien, Adjektive und Adverbien, als Adverbien beim Prädikat oder mit+Dat (.) Fragen Wie? Auf welche Weise?. Substantivisch durch Vergleich Nom (wie ein...)
 - (23) *Grad oder Intensität* — bestimmte Adverbien beim Prädikat oder indirekte Beschreibung in Nebensätzen. Frage Wie? Wie sehr?

(24) *Folge* — Nebensätze mit *so daß, weswegen, weshalb, womit, wodurch*. Fragen *Mit welcher Folge? Mit welchem Ergebnis?* etc.

- Rollen zur Begründung

(25) *Grund* — Nebensatz mit *weil, da* oder Hauptsatz mit *denn*. Frage *Wieso? Weshalb? Warum?*

(26) *Zweck* — Nebensatz mit *um...zu*. Fragen wie bei Grund und dazu *Zu welchem Zweck?, Wozu?, Wofür?* etc.

(27) *Vorbedingung* — Nebensätze mit *falls, wenn, sobald*. Fragen *Wann?, In welchem Fall?, Unter welcher Bedingung? Unter welchen Umständen?*

3.4 Chart-Parsing

SARA baut auf einer Analyse mit einer zentralen Datenstruktur namens “Active Chart” auf⁹.

Motivation für die Konstruktion eines Charts ist die Beobachtung, daß sowohl Bottom-Up als auch Top-Down Verfahren für die Zerteilung bei geradliniger Verwendung erhebliche Mengen an redundanter Arbeit leisten müssen, sobald ein einmal eingeschlagener Weg sich als falsch erweist, also ein Rücksetzen notwendig wird.¹⁰ In diesem Fall gehen nämlich alle bisher gewonnenen Informationen verloren; es bleibt nur das Wissen um den fehlgeschlagenen Versuch an sich. Das heißt, daß die Arbeit, die nötig war, um irgendwelche Teilstrukturen zu erkennen, ignoriert wird und eventuell noch einmal (oder mehrmals) neu geleistet werden muß. Nun ist aber das Rücksetzen nur bei der Analyse speziell gestalteter Sprachen zu vermeiden — bei natürlichen Sprachen jedenfalls nicht — und so sinnt man auf Abhilfe.

Eine Möglichkeit ist, alle bisher erzeugten Teilergebnisse in einer Tabelle (*well formed substring table*) abzuspeichern. Ein *Chart* ist nun lediglich eine besondere Darstellungs- und Betrachtungsweise dieser Tabelle: man betrachte den Eingabetext als gerichteten Graphen in der Weise, daß man jedes Eingabewort als die Markierung einer Kante dieses Graphen auffaßt und die Eingabewörter in ihrer tatsächlichen Reihenfolge hintereinander anordnet. Die Knoten des Graphen liegen also vor dem ersten und hinter dem letzten Wort und in allen Wortzwischenräumen. Alle Informationen für einen Chart-Parser werden in den Kanten dieses Graphen gehalten. Anfänglich sind dies nur die Bedeutungen der einzelnen Eingabeworte (evtl. mehrere je Wort); im Verlauf der Analyse werden auch alle in Wortgruppen erkannten Teile des Eingabesatzes als je eine Kante im Chart eingetragen.

Etwas genauer: Jedes Teilergebnis der Analyse hat die Form, daß eine Reihe von aufeinanderfolgenden Eingabewörtern (mindestens eines) unter einer bestimmten Strukturbezeichnung zusammengefaßt wird. Liegt nun ein solches Teilergebnis vor, so kann man es

⁹Wenn jemandem eine Übersetzung für dieses blöde Wort einfällt, möge er oder sie sich bitte schleunigst bei mir melden. In der Zwischenzeit behelfe ich mir damit, den Ausdruck 1:1 ins Deutsche zu übernehmen. Dabei entsteht dann noch das leicht peinliche Problem, daß man sich für eines von drei (gleichmäßig unpassend wirkenden) Geschlechtern entscheiden muß. Ich wähle nach der Regel “Im Zweifel sächlich” die Form “Das aktive Chart”.

¹⁰In diesem Abschnitt schwingt immer mit, daß fast alles, was man über Zerteilung weiß, auf Produktionssystemen basiert. Insbesondere wird (auch beim Chart-Parsing) meist ein kontextfreies Regelsystem angenommen. Die Analyse mit Fallschablonen ist damit jedoch nur weitläufig verwandt. Daher ist es notwendig, so weit wie irgend möglich von kontextfreien Systemen zu abstrahieren, weshalb hier manches etwas diffus klingt.

im Chart abspeichern, indem man eine zusätzliche Kante einträgt, die die betreffenden Wörter überspannt und als Markierung die jeweilige Strukturbezeichnung trägt. Eine solche Kante enthält außerdem Information darüber, welche der von ihr überspannten Kanten in ihr zusammengefaßt werden. Das ist notwendig, weil es zwischen zwei Knoten (d.h. durch einen Teilbereich des Graphen) meist mehrere mögliche Pfade gibt.

Man erhält so einen gerichteten, azyklischen Graphen, mit je genau einer Quelle und Senke, der nach erfolgreicher Analyse des Eingabesatzes eine Kante enthält, die alle Eingabewörter überspannt.

Nun ist allerdings bei dem bis hierher beschriebenen Vorgehen immer nur von einer Zusammenfassung von Kanten die Rede gewesen. Es ist also soweit nur ein Bottom–Up Verfahren möglich. Um nun auch umgekehrt Top–Down arbeiten zu können, muß man den Typ der *aktiven Kante* einführen. Eine aktive Kante ist eine solche, die noch keine Strukturbezeichnung aufweist, sondern nur eine Strukturhypothese, weil nämlich noch nicht alle zu der anvisierten Struktur gehörenden Teile aufgefunden worden sind. So eine Kante muß also Information darüber enthalten, was bereits gefunden ist, was noch gefunden werden muß und wie das bisher versucht wurde (um endlose Wiederholung zu vermeiden). Dies kann zum Beispiel durch Zuweisung eines inneren Zustands erfolgen.

Der Zerteilungsvorgang bei einem solchen Chart–Parser durchläuft eine Schleife, in der immer eine der aktiven Kanten ausgewählt und dann bearbeitet wird. Bei der Bearbeitung können neue aktive Kanten “vorgeschlagen” werden und die bearbeitete Kante kann mit einer anderen, nicht aktiven, verschmelzen, wobei eine neue Kante entsteht, die ihrerseits aktiv sein kann oder “erfolgreich”. Bei Mehrdeutigkeiten können auch mehrere aktive und erfolgreiche Kanten in einem Schritt entstehen. Schließlich besteht die Möglichkeit, daß die Kante sich nicht wie geplant fortführen läßt: Sie schlägt fehl.

Entsteht in diesem Prozeß eine Kante, die die gesamte Eingabe überspannt und nicht mehr aktiv ist, so ist die Zerteilung erfolgreich; die besagte Kante beschreibt das Analyseergebnis.

Die Bearbeitungsschleife endet spätestens, sobald keine aktive Kante mehr vorhanden ist. Sie kann auch vorzeitig abgebrochen werden, wenn ein zufriedenstellendes Ergebnis vorliegt und keine weiteren Mehrdeutigkeiten ausgewertet werden sollen.

Ein Chart–Parser eignet sich recht gut für die Zerteilung mit Fallschablonen. Die Rollen stellen auf natürliche Weise Teilergebnisse mittlerer Komplexität dar. Die (Teil)ergebnisse höherer Komplexität sind Instanzen von Fallschablonen für den ganzen Satz oder für Nebensätze und kleine Teilergebnisse werden von (syntaktisch zusammengehörigen) Wortgruppen oder einzelnen Worten gebildet.

Mit dieser Aufteilung ist es möglich, gleichzeitig syntaktisch und semantisch zu zerteilen. Die Zusammenfassung von Wörtern zu Wortgruppen ist eine weitgehend syntaktisch gesteuerte Arbeit, die bottom–up erledigt werden kann. Das Einfüllen solcher Wortgruppen in Fallschablonen erfolgt vor allem anhand semantischer Kriterien und wird top–down durchgeführt.

Kapitel 4

Architektur

*Dies ist das Kapitel
für das ein illustrierendes
oder erheiterndes Motto
nicht zu finden war.*

Der Verfasser

Dieses Kapitel beschreibt den Aufbau von **SARA**. Dazu gehören die Eingabeformate der externen Datenstrukturen und deren Semantik, der Leistungsumfang und die Struktur des Zerteilers sowie sein Algorithmus und schließlich das Ausgabeformat der Analyseergebnisse.

Die Beschreibung der Bedienung von **SARA** befindet sich in Kapitel 7.

4.1 Generelles

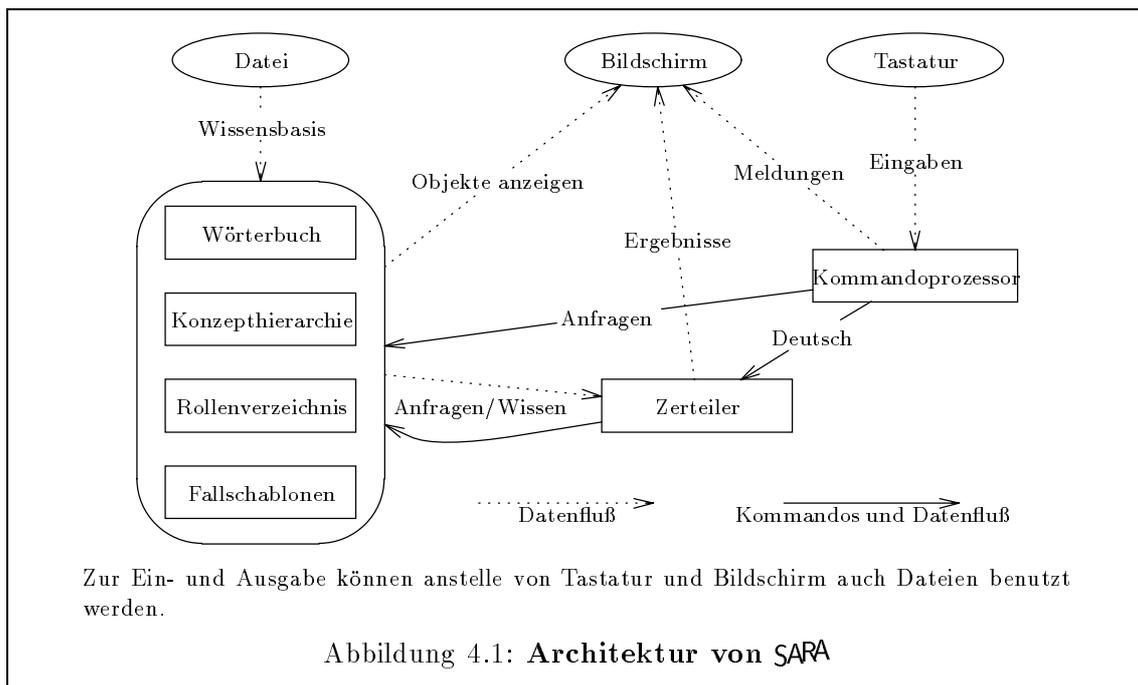
Im Wesentlichen besteht **SARA** aus drei Teilen:

Erstens der *Wissensbasis*, in der das auswechselbare und sachgebietsspezifische Wissen gehalten und verwaltet wird. Dazu gehören das Wörterbuch, die Konzepthierarchie, das Rollenverzeichnis und die Fallschablonen.

Zweitens dem *Zerteiler*, der zwar auch Wissen enthält, jedoch nur solches von fachgebietenunabhängiger Bedeutung, nämlich Wissen über den Aufbau und die Auswertung der Wissensbasis und Wissen über die deutsche Grammatik. Die Aufgabe des Zerteilers ist es, unter Verwendung der Wissensbasis Eingabesätze in Analyseergebnisse umzuformen.

Drittens schließlich aus dem *Kommandoprozessor*, der die Eingaben des Benutzers entgegennimmt, die Operationen des Zerteilers und der Wissensbasis aufruft und ihre Ergebnisse und Meldungen handhabt. Die wichtigsten Daten- und Kommandoflüsse zwischen diesen Komponenten sind in Bild 4.1 dargestellt.

Da der Kommandoprozessor in Anbetracht meiner Aufgabenstellung ein uninteressanter Teil ist, wird er hier nicht weiter beleuchtet. Im Rest dieses Kapitels wird also auf die externe Sicht der Wissensbasis und auf den Zerteiler näher eingegangen.



4.2 Ein-/Ausgabekategorien

Zur Kategorisierung von Eingaben und Ausgaben bei SARA, kann man folgende Einteilung vornehmen:

Eingaben sind

1. Kommandos. Diese gehen zum Kommandoprozessor und werden deshalb in Kapitel 7 beschrieben.
2. Deutsch. Dies sind die zu analysierenden Eingabesätze. Eine solche Eingabe wird an den Zerteiler übergeben, der sie analysiert und je nach gewählten Parametern daraus Instanzierungen oder Hilfeausgaben erzeugt. Welche Arten von deutschen Eingaben der Zerteiler verarbeiten kann, ist in Abschnitt 4.8.1 auf Seite 93 beschrieben. Welche konkreten Eingaben er verarbeiten kann, hängt darüberhinaus vom aktuellen Inhalt der Wissensbasis ab.
3. Wissensbasis. Die Eingaben an die Wissensbasis umfassen das Wörterbuch, die Konzepthierarchie, das Rollenverzeichnis und die Fallschablonen. Die Formate dieser Eingaben sind in den folgenden Abschnitten beschrieben.

Ausgaben sind

1. Meldungen. Dies umfaßt normale Betriebsmeldungen, Warnungen und Fehlermeldungen. Die genauere Beschreibung erfolgt in Kapitel 7.
2. Hilfeausgaben. Wird SARA als Hilfesystem benutzt, so wird als Reaktion auf die deutschen Eingaben nur jeweils ein fest mit der betroffenen Fallschablone verbundener Text ausgegeben.

3. Instanziierungen. Dies ist die eigentliche Nutzausgabe von **SARA**. Hier wird die vollständige Form aller aus einem Eingabesatz gewonnenen Informationen ausgegeben. Das Format ist in Abschnitt 4.9 auf Seite 109 beschrieben.

4.3 Die Wissensbasis

Die Wissensbasis besteht aus dem Wörterbuch, der Konzepthierarchie, dem Rollenverzeichnis und den Fallschablonen. Diese liegen in einer externen, menschenlesbaren Form als Textdatei vor und werden von **SARA** beim Einlesen in eine interne Darstellung überführt. Die externe Form soll nun beschrieben werden.

In den folgenden Abschnitten werden verschiedentlich formale Spezifikationen der Eingabeformate für die verschiedenen Wissensbereiche auftauchen.

Die an die Backus–Naur–Form angelehnte Syntax dieser Spezifikationen wird als bekannt vorausgesetzt (für leere Alternativen steht jeweils ε). Die Grundelemente der Eingabesprache (*token*) sind die Schlüsselwörter, die jeweils mit einem Doppelpunkt beginnen, sowie die Zeichenkette **#include** und Objekte vom Typ `wort`, `string` oder `zahl`. Die Token werden durch beliebig viele Zwischenraumzeichen getrennt. Zwischenraumzeichen sind das Leerzeichen, das Zeilenendezeichen, das Tabulatorzeichen und der Kommentar. Kommentare sind Zeichenfolgen, die in `/*` und `*/` eingeschlossen sind und `*/` nicht enthalten.

`wort`, `string` und `zahl` sind Ausdrücke, die folgende Eigenschaften haben:

wort: ein `wort` ist eine Folge über die Zeichen `a` bis `z`, `A` bis `Z`, `@` und `-`. Dabei darf `@` nur unmittelbar den Vokalen `a,o,u,A,O,U` vorausgehen; dies zeigt an, daß es sich bei dem darauf folgenden Vokal um einen umlautenden Stammvokal handelt. Diese Information wird ausschliesslich bei Substantiven verwendet, es wird dann bei der Bildung des Pluralstammes der Vokal durch seinen korrespondierenden Umlaut ersetzt. Umlaute werden als Vokal mit `e` eingegeben. Beispiele: `Vorschl@ag`, `uebel`. Querstriche werden 1:1 übernommen.

string: Ein `string` beginnt mit einem Begrenzer und enthält ab dort eine beliebige Folge von Zeichen, einschließlich Zeilenendezeichen, bis derselbe Begrenzer wieder auftaucht. Begrenzer können sein das Vorwärts- und das Rückwärtsapostroph sowie das Doppelanführungszeichen. Der Wert des strings ist die Zeichenfolge zwischen den Begrenzern. Beispiele:

```
"Dies ist ein 'richtiger' String"
'Dies ist ein "richtiger"String'
```

zahl: Eine `zahl` beginnt mit einem optionalen Querstrich (Minuszeichen) gefolgt von einer Folge der Ziffern `0` bis `9`, die als eine dezimale Ganzzahl interpretiert werden. Ein Dezimalkomma oder Dezimalpunkt ist nicht erlaubt.

Die Gestalt der Textdatei, die die Wissensbasis enthält, genügt folgender Grammatik:

```
wissensbasis →
    wissensbasis wissensbasiseintrag
    |
    ε
wissensbasiseintrag →
```

```

    wörterbucheintrag
  | konzepteintrag
  | rolleneintrag
  | fallschablone
  | #include string

```

Die Form von `wörterbucheintrag`, `konzepteintrag`, `rolleneintrag` und `fallschablone` ist in den folgenden Abschnitten definiert und erklärt.

Der string bei einem `#include`-Eintrag gibt eine Datei an. Die Bedeutung eines solchen Eintrags ist, daß die Verarbeitung der Wissensbasis am Anfang der angegebenen Datei fortgesetzt wird und nach deren Ende an die Stelle nach dem Eintrag zurückkehrt. Diese Anweisung darf auch verschachtelt vorkommen.

In den Einträgen werden noch folgende Objekte verwendet:

```

wortliste →
    '(' ')
  | '(' wortlisteninnern ')'
wortlisteninnern →
    wortfolge
  | wortfolge ',' wortlisteninnern
wortfolge →
    '(' ')
  | wort
  | '_'
  | '(' wortfolgeninnern ')'
wortfolgeninnern →
    wort
  | '_'
  | wort ',' wortfolgeninnern
  | '_' ',' wortfolgeninnern

```

Dabei dürfen die Kommata wahlweise auch weggelassen werden. Der Querstrich ist nach Definition auch ein Wort, die Unterscheidung wurde hier nur zur größeren Klarheit explizit gemacht. Wortfolgen und Wortlisten werden zu verschiedenen Zwecken eingesetzt; insbesondere zur Angabe von Oberflächenformen für Wörter und zur Angabe von Verweismengen auf andere Objekte der Wissensbasis.

Es folgt nun die Beschreibung der eigentlichen Wissensbasiseinträge.

4.4 Das Wörterbuch

4.4.1 Grundgedanke

Grundlage für eine Sprachanalyse ist die Bedeutungszuordnung für die einzelnen Wörter. Eine solche wird möglich, indem die *Oberflächenform* des Wortes (Buchstabenfolge) in einen Verweis auf einen Wörterbucheintrag umgesetzt wird. In diesem Wörterbucheintrag lassen sich dann beliebige Daten zur Verarbeitung des Wortes unterbringen.

4.4.2 Vorüberlegungen und Entwurfsentscheidungen

Die Umsetzung der Oberflächenform in einen Verweis auf einen Wörterbucheintrag wäre ein ganz einfacher Prozeß, wenn diese Oberflächenform der Wörter immer völlig gleich wäre. Leider ist es aber gerade im Deutschen so, daß sich diese Oberflächenform stark und vielfältig verändern kann (Beugung). Es muß also zur Herstellung des Verweises zuerst die gegebene Wortform in die zugehörige Grundform umgesetzt werden.

Dies kann grundsätzlich auf zwei Arten erfolgen: mittels algorithmischer Umwandlung der gegebenen Form in die Grundform oder mittels eines Formenverzeichnisses, in dem zu jeder Oberflächenform die zugehörige Grundform angegeben ist.

Da im Deutschen wie gesehen die Gesetze zur Bildung der Oberflächenformen sehr umfangreich sind und zudem eine große Zahl von Unregelmäßigkeiten vorkommen, ist eine algorithmische Umwandlung sehr aufwendig; sie kann im Rahmen dieser Arbeit nicht konstruiert werden und kann auch im Idealfall nicht alle Formen abdecken. SARA verwendet aus diesem Grund ein sogenanntes *Vollformenlexikon*, in dem alle Oberflächenformen, die das System erkennen soll abgespeichert sind. Das heißt das Wörterbuch enthält sowohl die Einträge für die Grundformen (Wortnamen), die zur Bildung von Konzepten benützt werden, als auch alle möglichen Oberflächenformen, die jeweils die Beugungsbeschreibung und einen Verweis auf die Grundform als Eintrag erhalten. Diese Methode ist zwar speicherintensiv, aber schnell. Man kann unabhängig von der Größe des Wörterbuches durch Hashtabellen jeden Wörterbucheintrag in gleichbleibender Zeit erhalten.

Es wäre nun nicht nur lästig, sondern schlicht unpraktikabel, diese Formen alle von Hand eingeben zu müssen. Glücklicherweise ist jedoch die Vorwärtsverarbeitung (d.h. die Erzeugung) der Formen leichter als die Rückwärtsverarbeitung (d.h. die Analyse, Rückführen auf die Grundform). Deshalb kann man bei der Eingabe des Wörterbuches neben der Grundform eine kleine Zahl von grammatischen Attributen mit angeben, und daraus in den meisten Fällen alle Formen automatisch erzeugen.

Für alle ungebeugten Wortarten sind ohnehin keine Zusatzangaben erforderlich.

Für Adjektive reicht die Angabe der Wortstämme für Positiv, Komparativ und Superlativ aus. Dies enthält, wo nötig, gleichzeitig die Information "nicht steigerbar", wenn man erlaubt, diese Stämme einzeln wegzulassen.

Bei Substantiven gibt es eine Kollektion von 3 Beugungsarten im Singular und 11 im Plural, was fast alle Fälle (mit Ausnahme einiger Fremdworte) abdeckt. Von diesen theoretisch 33 Möglichkeiten treten nicht alle auf. Es bietet sich aufgrund der geringen Zahl von Möglichkeiten an, eine einzige Typangabe mit einer endlichen Menge von Werten zur Beschreibung der Beugung bei Substantiven zu verwenden. Ich habe mich jedoch entschlossen, den Singular und den Pluralbeugungstyp einzeln anzugeben. Dies hat folgende Vorteile

1. Für Substantive, die nur im Singular oder nur im Plural auftreten muß nicht extra jeweils ein eigener Typ oder ein Zusatzangabe vorgesehen werden. Die Beschreibung ergibt sich, indem man einfach eine der Typangaben wegläßt.
2. Es sind auch solche Fälle zu erfassen, in denen ein Wort im Sprachgebrauch zwischen zwei Klassen schwankt, indem man die Angabe von mehr als je einem Typ für Singular und Plural zuläßt.

3. Bei der Erstellung des Wörterbuchs sind die Angaben durch Betrachtung von nur zwei Formen des aktuellen Wortes zu erhalten: die Endung im Genitiv Singular liefert den Singulartyp und die Form des Nominativ Plural den Pluraltyp. Diese Arbeitsweise senkt die Fehlerhäufigkeit bei der Erstellung des Wörterbuches ab und erleichtert es, das Eingabeformat zu erlernen.

Bei Verben gibt es drei Klassen: regelmäßige Verben, Verben mit Stammvokaländerung und völlig unregelmäßige.

Da die nicht regelmäßigen Verben nur einen sehr geringen Teil des Verbwortschatzes ausmachen, habe ich mich entschlossen, eine automatische Formenerzeugung nur für die echt regelmäßigen Verben vorzusehen.

Alle übrigen Wortarten sind vollkommen unregelmäßig. Für sie gibt es (wie auch bei Bedarf für Adjektive, Verben und Substantive) eine Möglichkeit, für jede einzelne Form eine Liste von alternativ möglichen Oberflächenformen von Hand einzugeben.

Das gewählte Eingabeformat ist recht statisch; ein Vertauschen der Teile ist innerhalb eines Worteintrags nirgends erlaubt. Diese Eigenschaft fördert ein einheitliches Erscheinungsbild der Wörterbücher und senkt damit die Fehleranfälligkeit.

4.4.3 Aufbau

Hier also eine BNF-Grammatik für die Einträge ins Wörterbuch, der Eintrag ε steht zur besseren Kennzeichnung leerer Alternativen:

```

wörterbucheintrag →
    '{' name wortbeschreibung '}'
name →
    wort
wortbeschreibung →
    :verb allgemeines :regelmaessig
    | :verb allgemeines verbformen
    | :praefix
    | :hilfsverb allgemeines verbformen
    | :modalverb allgemeines verbformen
    | :substantiv allgemeines substantivtyp
    | :substantiv allgemeines substantivformen
    | :adjektiv allgemeines steigerungsstämmen
    | :adjektiv allgemeines adjektivformen
    | :artikel allgemeines adjektivformen
    | :personalpronomen allgemeines substantivformen
    | :reflexivpronomen allgemeines substantivformen
    | :relativpronomen allgemeines adjektivformen
    | :interrogativpronomen allgemeines adjektivformen
    | :demonstrativpronomen allgemeines substantivformen
    | :possessivpronomen allgemeines substantivformen
    | :praeposition
    | :konjunktion

```

| :lokaladverb
| :temporaladverb
| :modaladverb

verbformen →
 praesens praeteritum konjunktivI konjunktivII
 partizip imperativ

praesens →
 :praesens wortliste
 | ε

praeteritum →
 :praeteritum wortliste
 | ε

konjunktivI →
 :konjunktivI wortliste
 | ε

konjunktivII →
 :konjunktivII wortliste
 | ε

partizip →
 :partizip wortliste
 | ε

imperativ →
 :imperativ wortliste
 | ε

substantivtyp →
 :typ wortfolge

substantivformen →
 singular plural

singular →
 :singular wortliste
 | ε

plural →
 :plural wortliste
 | ε

steigerungsstämme →
 :steigerungsstämme wortliste

adjektivformen →
 maskulinum femininum neutrum plural

maskulinum →
 :maskulinum wortliste
 :maennlich wortliste
 | ε

femininum →
 :femininum wortliste
 :weiblich wortliste
 | ε

```

neutrum →
    :neutrum wortliste
    :saechlich wortliste
    | ε
allgemeines →
    stamm_oder_infinitiv_mit_zu geschlecht person steigerung
stamm_oder_infinitiv_mit_zu →
    :stamm wort
    | :zu_infinitiv wort
    | ε
geschlecht →
    :geschlecht wortfolge
    | ε
person →
    :person wortfolge
    | ε
steigerung →
    :steigerung wortfolge
    | ε

```

(Bei den Schlüsselworten sind alternativ zu `:substantiv`, `:adjektiv` und `:verb` auch `:sub`, `:adj` und `:vb` zu verwenden.)

Diese Grammatik bedarf zu ihrer korrekten Anwendung noch einiger Erläuterungen:

1. **‘name’**: Der Name des Wörterbucheintrags ist für Substantive, Adjektive, Pronomen und Artikel im Prinzip willkürlich und wird zunächst als die Grundform angenommen. Falls dort nicht die Grundform angegeben ist, muß diese im allgemeinen Teil mit `:stamm` aufgeführt sein. Bei den ungebeugten Wortarten muß der Name das Wort selbst sein, bei Verben der Infinitiv Präsens.
2. **‘-’**: Ein `-` in einer Wortfolge oder Wortliste ist ein Platzhalter. Dieser heißt in einer Liste von Wortformen, daß die betreffende Form nicht erzeugt werden soll. An anderen Stellen ist ein Platzhalter nicht erlaubt.
3. **Eingeschachtelte Wortfolge**: Eine Wortfolge innerhalb einer Wortliste (die immer eine Liste von Wortformen ist) bedeutet eine Menge von wahlweise möglichen Formen, die alle eingetragen werden.
4. **‘allgemeines’**: Die Angaben im allgemeinen Teil eines Wörterbucheintrags werden wie folgt verwendet: Vor der Analyse des allgemeinen Teils werden alle für die jeweilige Wortart relevanten Einträge (s. unten) auf ihre jeweiligen Standardwerte gesetzt. In der Analyse können sich nun alle Einträge beliebig ändern. Nach der Analyse werden alle für die jeweilige Wortart nicht relevanten Einträge auf Leerwerte gesetzt. Im Einzelnen:
5. **‘stamm’**: Die Stammangabe kann entfallen, wenn sie mit dem Namen des Wörterbucheintrages übereinstimmen würde, ausgenommen bei Verben mit abtrennbarem Präfix. Der Wortstamm für automatisch erzeugte Formen ist bei Substantiven der Nominativ Singular und bei Adjektiven die ungebeugte Form. Bei Substantiven kann gegebenenfalls der umlautende Stammvokal mit einem vorangehenden `@` markiert

sein; gibt es eine solche Markierung nicht, wird der erste Vokal als Stammvokal angenommen. Bei Verben ist die Stammangabe stets notwendig, wenn sich nicht der Stamm durch Abtrennen von **-en** (oder **-n**, falls kein **e** vorhanden) vom Namen des Eintrags ergibt. Die Angabe des Verbstammes darf einen eventuell zum Wort gehörenden abtrennbaren Präfix nicht umfassen. Nur dann kann die nötige Information zur Verarbeitung von Verben mit abgetrenntem Präfix erzeugt werden, z.B. Name **aufmachen** hat **:stamm mach**; der dann erzeugte Wortname lautet **machen** und dies wird ggf. als Synonym in der Konzepthierarchie angegeben (siehe dort). In anderen Fällen sind Angaben zum Stamm nicht nötig.

6. **‘geschlecht’**: relevant für Substantive (und teilweise die Pronomen). Standardwert ist bei regelmäßigen Substantiven mit Singulartyp S ‘weiblich’, bei Pluraltyp Pue ‘saechlich’, sonst ‘maennlich’. Aufgrund dieser Festlegungen findet zwar nur für männliche und weibliche Substantive immer eine richtige Festlegung des Geschlechts statt, leider ist aber im Deutschen eine automatische Unterscheidung von Maskulina und Neutra nur selten möglich, so daß diese Einschränkung unvermeidlich ist; außerdem gibt es ja auch mehrgeschlechtliche Wörter. Für diese Fälle muß also eine Geschlechtsangabe erfolgen. Die automatische Bestimmung des Geschlechts legt aber schon mit diesen simplen Methoden in etwa drei Vierteln der Fälle den richtigen Wert fest. Eine Geschlechtsangabe ist eine beliebige Auswahl aus den Wörtern **maennlich**, **weiblich**, **saechlich**, **maskulinum**, **femininum** und **neutrum**, die jeweils durch ihren Anfangsbuchstaben abgekürzt werden können. Ist mehr als ein Geschlecht angegeben, so behandelt der Zerteiler das Wort als mehrgeschlechtlich und reagiert entsprechend großzügiger bei der Kontrolle der Kongruenz.
7. **‘person’**: relevant für Substantive (insbesondere Pronomen), Standardwert ‘dritte’. Eine Geschlechtsangabe ist eine beliebige Folge der Wörter **erste**, **zweite** und **dritte**, die jeweils durch ihren Anfangsbuchstaben abgekürzt werden können.
8. **‘steigerung’**: relevant für Adjektive, Standardwert ist ‘positiv’. Die Steigerungsangabe ist eine beliebige Auswahl der Wörter **positiv**, **komparativ** und **superlativ**, die jeweils durch ihren Anfangsbuchstaben abgekürzt werden können. In der Regel wird nur einer dieser Werte angegeben.
9. **Verben**: Die meisten Vollverben sind regelmäßig und benötigen überhaupt keine zusätzlichen Angaben zur Erzeugung ihrer Formen; diese werden gemäß Tabelle 2.1 (Seite 39) aus dem Verbstamm gebildet. Hilfs- und Modalverben sind immer unregelmäßig. Bei unregelmäßigen Verben sind folgende Formen anzugeben: Die Wortlisten für Präsens, Präteritum und Konjunktiv I/II bestehen immer aus 6 Einträgen: erst für 1./2./3. Person Singular, dann Plural. Die Wortliste **:partizip** enthält Partizip I und Partizip II, diese werden als Singular-und-Plural und erste-bis-dritte-Person eingetragen. Die Wortliste **:imperativ** enthält die Formen für Imperativ Singular und Imperativ Plural. Als Infinitiv wird der Name des Wörterbucheintrags benutzt. Bei unregelmäßigen Verben mit abtrennbarem Präfix kann der Infinitiv mit ‘zu’ an der Stelle einer Stammangabe angegeben werden, er wird dann als einteilige Form mit in das Wörterbuch aufgenommen. Die Partizipien müssen zur Verwendung als Adjektiv, falls gewünscht, einzeln als solche eingetragen werden.
10. **Substantive**: Substantive sind fast immer in eine aus einer Reihe von Klassen einzuordnen. Dazu wird bei **:typ** die Art der Singularbildung und die Art der

Pluralbildung angegeben. Bei Substantiven die nicht im Singular oder nicht im Plural auftreten können oder teilweise unregelmäßig sind, ist die entsprechende Klasse wegzulassen und der Rest ggf. in einem weiteren Eintrag zu spezifizieren. Für die möglichen Klassen und ihre Bedeutung siehe Tabellen 2.5 (Singular), 2.6 bis 2.8 (Plural) und die zugehörigen Textabschnitte (ab Seite 40). Dabei werden im allgemeinen nicht mehr als ein Singular- und ein Pluraltyp angegeben; falls doch, ist zu beachten, daß die umlautenden Pluraltypen nur mit anderen umlautenden Pluraltypen verträglich sind, die nicht umlautenden nur mit nicht umlautenden und die Fremdwortplurale Pi und Pue mit keinem anderen Plural. Bei den seltenen unregelmäßigen Substantiven und den als Substantive zu behandelnden Pronomen sind bei **:singular** und bei **:plural** jeweils 4 Formen für Nominativ, Genitiv, Dativ und Akkusativ anzugeben. Bei Pronomen gewinnt auch die Angabe einer Person Bedeutung, bei normalen Substantiven ist dies immer 'dritte'.

11. **Adjektive:** Adjektive sind in aller Regel vollkommen regelmäßig. Sie haben unterschiedliche Stämme für Positiv, Komparativ und Superlativ (wenn sich das Adjektiv überhaupt steigern läßt) und bilden aus diesen ihre Formen gemäß der Adjektivbeugung vom Typ I, II und III, wie in Tabelle 2.9 auf Seite 44 angegeben. Die Steigerungsstämme (oder Platzhalter) werden bei **:komparation** angegeben. Im unregelmäßigen Fall muß jede Steigerung einzeln eingetragen werden, ebenso jeder Beugungstyp. Dabei enthält jede Wortliste 4 Einträge für Nominativ, Genitiv, Dativ und Akkusativ. Es gibt drei Wortlisten für jedes der drei Geschlechter im Singular und eine Wortliste für den Plural, wo bei allen Adjektiven und adjektivisch verwendeten Pronomen die Formen für alle Geschlechter zusammenfallen. Dieses Eingabemuster findet jedoch seine eigentliche Berechtigung nur bei Pronomen, die als Adjektiv behandelt werden. Normale Adjektive sind immer regulär. Dabei gibt es einen einzigen Sonderfall, bei dem diese Methode nicht mehr unmittelbar zur Darstellung ausreicht, und zwar die Possessivpronomen. Hierbei liegt nämlich der Fall vor, daß Geschlecht und Numerus sozusagen doppelt vorhanden sind: einmal für das Wort mit dem das Pronomen steht und einmal für die Person auf die es sich bezieht. Da jedoch sämtliche Pronomen ohnehin im Zerteiler oder von den nachfolgenden Verarbeitungsstufen speziell behandelt werden müssen, genügt es, je einen Wörterbucheintrag für jede Kombination von Geschlecht und Numerus der Bezugsperson zu machen.
12. **Artikel:** Artikel werden ins Wörterbuch eingegeben wie unregelmäßige Adjektive. Sie bekommen jedoch eine eigene Wortart zugewiesen, um eine spezielle Behandlung im Zerteiler zu ermöglichen.
13. **Relativpronomen:** dito
14. **Präpositionen, Konjunktionen und Adverbien:** Diese Wortarten unterliegen nicht der Beugung. Es sind deshalb keinerlei Zusatzangaben zu machen und darum muß der Name des Eintrags der Wortform entsprechen.
15. **Personalpronomen und Reflexivpronomen:** Diese Wortarten werden wie (unregelmäßige) Substantive eingetragen, jedoch mit eigener Wortart, um dem Zerteiler eine Unterscheidung zu ermöglichen. Hier kommt die Möglichkeit zum Tragen, im allgemeinen Teil eine Person angeben zu können.
16. **Interrogativpronomen** Die Interrogativpronomen werden eingetragen wie unregelmäßige Adjektive, jedoch mit eigener Wortart, um eine spezielle Behandlung

im Zerteiler zu ermöglichen.

17. **Demonstrativpronomen:** Das Demonstrativpronomen *der, die, das* kann im Prinzip als Substantiv eingetragen werden. Es empfiehlt sich jedoch sehr, das nicht zu tun, da die große Menge von Mehrdeutigkeiten durch Zusammenfallen mit dem bestimmten Artikel, die sich dabei ergibt, beim Zerteilen einen stark erhöhten Aufwand und schlechtere Ausgaben nach sich zieht. Die Demonstrativpronomen *dieser, jener, derselbe, derjenige* und ihre weiblichen und sächlichen Gegenstücke können sowohl als Artikel als auch als Substantive gebraucht werden und werden entsprechend für beide Wortarten eingetragen.
18. **Possessivpronomen:** Die Possessivpronomen kommen ebenfalls im Gebrauch als Artikel und als Substantive vor (allerdings in unterschiedlich gebeugter Form) und werden dementsprechend für beide Wortarten eingetragen. Siehe oben bei Adjektiven.

Zur Veranschaulichung, siehe die Standardwissensbasis (Seite 135ff) und die Beispielwissensbasis (Seite 144ff).

4.5 Die Konzeptionshierarchie

4.5.1 Grundgedanke

Ein Fallschablonensystem wäre wenig leistungsfähig, wenn man zu jeder Rolle genau das Wort angeben müßte, das sie füllen soll. Es ist vielmehr nötig, an jeder Stelle statt nur eines Wortes einen beliebigen Vertreter einer bestimmten Wortmenge, der Synonymmenge dieses Wortes, zuzulassen. Anstatt nun diese Synonymmenge jedesmal in der Fallschablone anzugeben, kann man das Wörterbuch um Synonymlisten bei jedem Wort erweitern; das so entstehende Synonymverzeichnis nennt man einen *Thesaurus*. Auf diese Weise braucht man die Synonyme eines Wortes nur einmal anzugeben und sie werden dennoch stets berücksichtigt. Die Synonyme sind quasi die Inkarnationen eines Konzepts.

Nun möchte man natürlich bei den Fallschablonen nicht nur synonyme Begriffe automatisch behandeln, sondern möglichst alles, was an der jeweiligen Stelle sinnvoll sein kann. Es sollten also in den Fallschablonen nicht syntaktische Kategorien, sondern semantische Kategorien (Konzepte) eingetragen sein. Eine solche semantische Kategorie bilden im Allgemeinen zu einem gegebenen Wort die Synonyme, zuzüglich aller Unterbegriffe, zuzüglich¹ aller Oberbegriffe. Man ergänzt also den Thesaurus auch noch bei jedem Eintrag um Verweise auf die Ober- und Unterbegriffe. Eine solche Struktur stellt eine Hierarchie von sinnverwandten Wortklassen dar; sie ordnet semantische Konzepte. Aus diesem Grund heißt diese Struktur *Konzeptionshierarchie*.

4.5.2 Vorüberlegungen und Entwurfsentscheidungen

Bei der Bildung der Synonymliste möchte man nicht gern jede einzelne Wortform hinschreiben, die als Inkarnation des Konzepts zulässig ist. Zudem muß ja ohnehin eine Verbindung zwischen Wörterbuch und Konzeptionshierarchie hergestellt werden. Aus diesem

¹meistens, je nach Fall

Grund werden in der Synonymliste Wortnamen angegeben. Es sind dann jeweils alle Wortformen erlaubt, die im Wörterbuch mit diesem Wortnamen eingetragen sind.

Zur Bildung der Ober-/Unterbegriffshierarchie werden nur die direkten Oberbegriffe eines jeden Konzeptes angegeben. Die Unterkonzepte werden aus diesen Verweisen berechnet. Man könnte es natürlich auch genau andersherum machen, ich gehe jedoch davon aus, daß die Menge der Oberkonzepte kleiner und einfacher geistig zu erfassen ist, als die Menge der Unterkonzepte.

4.5.3 Aufbau

Ein Eintrag in der Konzepthierarchie enthält den Namen des Konzeptes, eine Wortartangabe, eine Definition, eine Synonym- und eine Oberbegriffsliste, sowie, (bei Verbkonzepten) eventuell einen Querverweis auf ein semantisch korrespondierendes Substantivkonzept.

Die Unterbegriffe werden aus der Gesamtheit der Einträge rechnerisch abgeleitet.

Hier also die BNF-Grammatik für die Konzepthierarchieeinträge:

```

konzepteintrag →
    '<' konzeptname wortart definition
    synonymliste oberbegriffsliste substantivquerverweis '>'

konzeptname →
    wort

wortart →
    :verb
    | :substantiv
    | :adjektiv
    | :praeposition
    | :konjunktion
    | :lokaladverb
    | :temporaladverb
    | :modaladverb

definition →
    :definition string
    | ε

synonymliste →
    :synonyme wortfolge
    | ε

oberbegriffsliste →
    :oberbegriffe wortfolge
    | ε

substantivquerverweis →
    :substantiv wort
    | ε

```

(Bei den Schlüsselworten sind alternativ zu `:synonyme`, `:oberbegriffe` und `:substantiv` auch `:syn`, `:ober` und `:sub` zu verwenden.)

Erläuterungen zur Grammatik:

1. Der Name des Eintrags ist beliebig. Er ist insbesondere in keiner Weise an die Namen von Wörterbucheinträgen gekoppelt.
2. Die Identifikation eines Konzepteintrages erfolgt durch seinen Namen.
3. Der Eintrag der Definition wird vom Zerteiler auf Wunsch ausgegeben, wenn das betreffende Konzept den Kopf einer Fallschabloneninstanziierung bildet, die die ganze Eingabe abdeckt.
4. Die Synonymliste stellt die einzige Verbindung zwischen den Konzepteinträgen und den Wörterbucheinträgen her. Aus diesem Grund muß in der Synonymliste auch ein mit dem Konzeptnamen übereinstimmendes Synonym (falls existent) mit angegeben werden. Als Synonym gibt man jeweils einen Wortnamen an, nicht eine Wortform. Beachte, daß der Wortname von Verben mit abtrennbarem Präfix diesen nicht mit enthält; es braucht deshalb für eine beliebige Anzahl von Verben, die sich nur im Präfix unterscheiden, nur ein Synonymeintrag gemacht zu werden.
5. Die Wortartangabe wird herangezogen, um die Menge der betrachteten Wörterbucheinträge bei den Synonymen einzuschränken. Das heißt die Angabe der Wortart in der Synonymliste ist unnötig, weil diese immer mit der Wortart des Konzepteintrags selbst übereinstimmt.
6. Die Oberbegriffsliste stellt eine Liste von Konzepteintragsverweisen dar. Dabei wird der Name jeweils direkt angegeben und die Wortart wird vom aktuellen Eintrag übernommen.
7. Der Substantivquerverweis stellt einen Eintrag dar, der nur bei Verben erlaubt ist. Er enthält den Namen eines Konzepts, das als Substantivierung des Verbs aufgefaßt werden kann. Dieses wird benutzt, um bei Fallschablonen die Substantivköpfe aus den Verbköpfen errechnen zu können, wo dies gewünscht wird.
8. Verweise auf nicht existierende Wörterbuch- oder Konzepteinträge sind erlaubt und führen zu Voreinträgen.
9. Es sind nicht für alle Wortarten sinnvoll Einträge im Konzeptverzeichnis zu erstellen: für Präpositionen gibt es das Rollenverzeichnis, Hilfs- und Modalverben können nur als Vollverben eingeordnet werden und Pronomen sind als solche nicht spezifisch einordnungsfähig und müssen deshalb vom Zerteiler behandelt werden.

Zur Veranschaulichung, siehe die Beispielwissensbasis (Seite 144ff).

4.6 Das Rollenverzeichnis

4.6.1 Grundgedanke

Die Anzahl sinnvoll zu verwendender Rollen für die deutsche Sprache insgesamt ist recht groß, eine Auflistung der wichtigsten Rollen steht in Abschnitt 3.3 ab Seite 71. Dabei

gibt es bei vielen Rollen eine ganze Anzahl von möglichen Präpositionen, so daß insgesamt einige Rollen eine genau gleiche Form aufweisen. Ähnliches gilt für die Wortkombinationen, mit denen eine Rolle erfragt werden kann. Angesichts der Größe der Datenmenge sollte man die Darstellung der Rollen nicht fest in den Zerteiler einkodieren, sondern als Datenstruktur realisieren. Macht man diese Datenstruktur extern, das heißt zur Laufzeit frei veränderbar, so schafft man damit ein zusätzliches Mittel um in bestimmten Fällen durch stärkere Einschränkung des Sprachbereichs Mehrdeutigkeiten zu vermeiden oder, andersherum, die Robustheit der Analyse durch Verringerung der syntaktischen Einschränkungen zu vergrößern. Außerdem können so die Rollennamen, bei Bedarf an die Terminologie der Anwendung angepasst werden.

4.6.2 Vorüberlegungen und Entwurfsentscheidungen

Manche Rollen erlauben einige ihrer Erscheinungsformen nur, wenn sie in einem Satz verwendet werden und andere nur bei der Verwendung mit einer Nominalphrase. Deshalb muß es bei der Angabe der Rollenformen möglich sein, die Verwendung entsprechend einzuschränken. Andererseits ist diese Unterscheidung auch oftmals irrelevant. Deshalb gibt es in einem Rolleneintrag drei Listen von Rollenformen: eine mit Rollenformen nur für Sätze (d.h. für verbale Fallschablonen), eine mit Rollenformen nur für Nominalphrasen (d.h. für nominale Fallschablonen) und eine mit Rollenformen, die in beiden Situationen zulässig sind.

In Imperativsätzen gibt es kein Subjekt. Diese Tatsache kann bei der Steuerung des Zerteilers Probleme bereiten, wenn er nicht weiß, daß die Rolle, die eigentlich vom Subjekt gefüllt wird, bei einem Imperativsatz nun nicht aufzutauchen braucht. Der Zerteiler könnte nämlich bei seiner Gütebewertung die Instanziierung als schlecht ansehen, obwohl sie es gar nicht ist. Dies ist vor allem deshalb kritisch, weil das Subjekt ein ansonsten meist unverzichtbarer Teil des Satzes ist und deshalb die Gütebewertung auf sein Fehlen empfindlich reagieren muß. Es sollte also eine Möglichkeit geben, im Rollenverzeichnis zu signalisieren, wenn eine Rolle im Imperativ wegfällt. Zu diesem Zweck gibt es eine Subjektmarkierung.

4.6.3 Aufbau

Die Einträge ins Rollenverzeichnis enthalten einen Rollennamen, eine Beschreibung der Darstellungsformen in denen die Rolle auftreten kann und eine Beschreibung der Frageformeln, die es erlauben diese Rolle zu erfragen.

Hier also die BNF-Grammatik für die Rollenverzeichniseinträge:

```

rolleneintrag →
    '<<' rollenname subjekt rollenformen
    verbalrollenformen nominalrollenformen frageformen '>>'
rollenname →
    wort
subjekt →
    :subjekt
  
```

```

    | ε
rollenformen →
    rollenformen rollenform
    | ε
rollenform →
    :nominativ
    | :genitiv
    | :dativ
    | :akkusativ
    | :lokaladverbial
    | :modaladverbial
    | :temporaladverbial
    | :nominativ wortfolge
    | :genitiv wortfolge
    | :dativ wortfolge
    | :akkusativ wortfolge
    | :nebensatz wortfolge
verbalrollenformen →
    :nur_verbal rollenformen
    | ε
nominalrollenformen →
    :nur_nominal rollenformen
    | ε
frageformen →
    frageformen frageform
    | ε
frageform →
    :frage wortfolge
    | :fragenominativ wortfolge
    | :fragegenitiv wortfolge
    | :fragedativ wortfolge
    | :frageakkusativ wortfolge

```

Erläuterungen zur Grammatik:

1. Der Rollename ist völlig frei zu wählen. Er kollidiert keinesfalls mit einem Namen aus dem Wörterbuch oder der Konzepthierarchie.
2. Bei den Rollenformen bedeutet ein einfaches Kasusschlüsselwort, daß es sich um eine Rolle handelt, die durch eine Substantivphrase ohne Präposition dargestellt werden kann. Die Wortfolge gibt hinter einer Kasusangabe eine Liste von Präpositionen an, von denen jede, zusammen mit einer Substantivphrase im betreffenden Fall stehend, einen Füller für die betreffende Rolle ergibt. Hinter der Angabe **:nebensatz** liefert die Wortliste eine Menge von alternativ möglichen Konjunktionen, die einen Nebensatz zu dieser Rolle einleiten können.
3. Alle in den Wortfolgen bei Rollenformen aufgeführten Wörter werden automatisch ins Wörterbuch als Präposition bzw. Konjunktion eingetragen.

4. Bei den Frageformen ist grundsätzlich zu unterscheiden zwischen schwach gerichteten und stark gerichteten Fragen; siehe den Absatz über die W-Fragen auf Seite 58.
5. Eine Angabe für eine schwach gerichtete Frage wird mit dem Schlüsselwort **:frage** angekündigt. Die darauf folgende Wortliste gibt die exakte Folge von Wörtern an, mit denen die Rolle erfragt wird, z.B. **Wo hinein** für die Rolle Zielpunkt. Diese Wörter werden automatisch als Einträge mit der speziellen Wortart Fragewort in das Wörterbuch aufgenommen.
6. Eine stark gerichtete Frage enthält eine Substantivphrase, die dazu dient den möglichen Bereich der Antworten bereits mit der Frage einzuschränken. Es wird also das erfragte Objekt mit angegeben, um Mehrdeutigkeiten zu verringern. Dieses Objekt steht je nach Rolle in einem bestimmten Kasus, der durch die Wahl des Schlüsselwortes angegeben wird. Dem Objekt voraus geht im Satz der konstante Teil der Frageform, der hier mit der Wortliste spezifiziert wird. Beispiel: die Frage **Unter welchem Fuß...?** wird angegeben als **:fragedativ (unter)**. Auch hier werden die Fragewörter automatisch als solche ins Wörterbuch aufgenommen.
7. Man beachte, daß es einige erstarrte Formen von stark gerichteten Fragen gibt, die als schwach gerichtete aufzufassen sind, z.B. ist **Auf welche Weise?** anzugeben als **:frage (auf welche weise)**.
8. Die Eintragung **:subjekt** für eine Rolle bedeutet, daß die Füller dieser Rolle im Eingabesatz als Subjekt auftreten. Dies hat zwei Auswirkungen: Bei Imperativsätzen wird nicht verlangt, daß diese Rolle gefüllt wird und wo ein Prädikat und ein Subjekt in einem Satz vorkommen, müssen ihre Formen in Person und Numerus kongruent sein.

Zur Veranschaulichung, siehe die Standardwissensbasis (Seite 135ff).

4.7 Die Fallschablonen

4.7.1 Vorüberlegungen und Entwurfsentscheidungen

Eine Fallschablone besteht aus einer Menge von Rollen und einem Kopf. Die Rollen liefern jeweils zweierlei Einschränkungen für die erlaubten Eingaben: die Rolle selbst definiert eine Reihe von möglichen syntaktischen Erscheinungsformen, der Füller der Rolle definiert eine semantische Einschränkung. Der Füller einer Rolle ist entweder eine Substantivphrase, ein Temporal-, Lokal- oder Modaladverb oder ein Nebensatz. Es wird in der Fallschablone dafür jeweils eine Menge von Konzepten angegeben.

Da bestimmte Rollen mit gleichen Füllern in sehr vielen Fallschablonen auftauchen, wäre es praktisch, eine Art Makromechanismus zu haben, mit dem man solche Standardfälle mit weniger Schreibaufwand in die Fallschablone einbauen kann. Es gibt deshalb sogenannte kopflose Fallschablonen. Diese können nicht selbst instanziiert werden, sondern dienen ausschließlich dazu, ihre Rollen an andere Fallschablonen zu vererben. Diese Vererbung geschieht, indem die Fallschablone, welche erben möchte, alle (kopflosen) Fallschablonen, von denen sie erben möchte, in einer entsprechenden Erbklausele angibt.

Außerdem werden in Fallschablonen von Konzepten, die in der Konzepthierarchie auf demselben Unterbegriffspfad liegen, häufig dieselben Rollen verwendet. Dies geschieht meist in der Weise, daß stärker spezialisierte Kopfkongzepte alle Rollen ihrer Oberbegriffe haben und nur noch zusätzliche Rollen definieren oder für auch beim Oberkonzept vorhandene Rollen andere Füller definieren. Diesem Verhalten trägt die Wissensbasis dadurch Rechnung, daß Rollen automatisch zu Unterkongzepten vererbt werden. Sollte ein Unterkongzept für eine dieser Rollen einen anderen Füller benötigen, so kann in der Fallschablone des Unterkongzepts die Rolle nochmals neu angegeben werden. Sollte ein Unterkongzept eine Rolle ausnahmsweise nicht übernehmen wollen, so kann sie mit einem leeren Konzept als Füller neu angegeben werden. Zusätzliche Rollen werden ohne weitere Umstände angegeben. Dieser Mechanismus macht in vielen Fällen die Bildung von Fallschablonen zu Kongzepten komplett überflüssig, da eine sinnvolle Fallschablone durch die Vererbung implizit entsteht.

In manchen Fällen ist die Konzepthierarchie bei der Beschreibung der semantischen Einschränkungen an die Füller hinderlich: Es können Mehrdeutigkeiten dadurch entstehen, daß ein Begriff, der in eine Rolle eingefüllt werden soll, zugleich ein Oberbegriff des erwarteten Füllers einer anderen Rolle ist, so daß er in manchen Fällen (je nach grammatikalischer Form) auch dort eingefüllt werden könnte. Vor allem die Bildung stark spezialisierter Rollen wird durch diesen Effekt erschwert. Als Abhilfe sieht die Wissensbasis vor, bei den Füllern einer Rolle die Verwendung von Oberbegriffen des angegebenen Kongzepts zu sperren. Dies kann für jede Rolle in jeder Fallschablone einzeln erfolgen.

Zur Steuerung des Zerteilers ist es außerdem sinnvoll, bei einer Rolle angeben zu können, wie wichtig es ist, daß diese Rolle im Eingabesatz tatsächlich gefüllt wird. Im Prinzip reichen dafür meist schon wenige feste Stufen aus, es ist jedoch flexibler und durchsichtiger, die Beschreibung der Dringlichkeit durch eine ganze Zahl vorzunehmen. Dadurch steht erstens eine fast unbegrenzt feine Unterscheidungsmöglichkeit für solche Dringlichkeiten zur Verfügung und zweitens kann der Entwerfer einer Wissensbasis durch Offenlegung der mit den Dringlichkeiten betriebenen Arithmetik die Reaktionen des Zerteilers leichter verstehen und vorhersehen.

4.7.2 Aufbau

Hier also die BNF-Grammatik für die Fallschablonen:

```

fallschablone →
    '[' kopf kommentar beerbung rollen ']'
kopf →
    wort
    wort :kopflos
    | wort :verb
    | wort :substantiv
kommentar →
    string
    | ε
beerbung →
    :erbe wortfolge
    | ε

```

```

rollen →
    rollen rolle
    | ε
rolle →
    :rolle rollenname füller kein_oberbegriff dringlichkeit
rollenname →
    wort
füller →
    wortfolge
kein_oberbegriff →
    :kein_oberbegriff
    | ε
dringlichkeit →
    :dringlichkeit zahl
    | ε

```

Erläuterungen zur Grammatik:

1. Der Name einer Fallschablone ist der Name des Konzepteintrages, der den Kopf der Fallschablone bildet. Bei Fallschablonen ohne Kopf ist der Name beliebig.
2. Die Wortfolge bei `:erbe` ist eine Liste von Namen solcher Fallschablonen, die keinen Kopf haben. Alle Rollen dieser Fallschablonen werden in die aktuelle mit übernommen.
3. Ferner gibt es eine implizite Vererbung, die alle Rollen von Fallschablonen vererbt, deren Kopf Oberbegriff des hiesigen Kopfes in der Konzepthierarchie ist.
4. Der Rollenname muß einen korrespondierenden Eintrag im Rollenverzeichnis haben; andernfalls erfolgt eine Fehlermeldung.
5. Die Dringlichkeit wird zusammen mit der Distanzinformation aus der Konzepthierarchie verwendet, um bei Mehrdeutigkeiten in der Rollenzuordnung die am besten passende Interpretation auszuwählen. Die Dringlichkeit kann jeden positiven Wert haben. Der Standardwert ist (willkürlich gewählt) 20. Die Interpretation der Dringlichkeit ist folgende: Während des Zerteilungsvorganges hat jede Fallschablone einen sogenannten *Gütwert*, der sich jedesmal verändert, wenn etwas eingefüllt wird. Wird ein Füller für eine Rolle mit der Dringlichkeit D gefunden, so erhöht sich der Gütwert um D . Ist er hingegen am Ende der Analyse nicht gefunden worden, so verringert sich der Gütwert um D minus 20, vorausgesetzt D ist größer als 20. Für die Wahl des Wertes 20 war ausschlaggebend, daß so einerseits eine genügend große Zahl von Abstufungen möglich ist (nämlich 20 nach unten und beliebig viele nach oben) und zugleich die Zahlen eine handhabbare Größe behalten.
6. Die Wortliste Füller wird als eine Menge von Verweisen auf Konzeptinträge interpretiert. Wenn ein entsprechender Eintrag für einen Konzeptnamen in der Liste nicht existiert, so wird ein Vorauseintrag gemacht. Bei der Instanziierung der Fallschablone wird diese Liste als eine Disjunktion von erlaubten Füllerkonzepten behandelt, die Liste gibt also eine Auswahl von Möglichkeiten an.

7. Die Angabe von `:kein_oberbegriff` verhindert, daß Einträge in der Konzepthierarchie, die direkte oder indirekte Oberbegriffe des angegebenen Füllers sind, als korrekte Füller in Betracht gezogen werden.

Zur Veranschaulichung, siehe die Beispielwissensbasis (Seite 144ff).

4.8 Der Zerteiler

Dieses Kapitel beschreibt die Arbeitsweise des Zerteilers und die damit erzielten (qualitativen) Analyseleistungen.

4.8.1 Leistungsumfang

Dies sind die Eigenschaften der syntaktischen Konstruktionen, die vom Zerteiler erkannt werden:

- *(deklarative) Hauptsätze* decken die gesamte Eingabe ab.
Hauptsatzinstanzen können beliebig viele Rollenfüller aller Arten enthalten. Das Prädikat kann in jeder einteiligen Form stehen oder in einer zweiseitigen mit einem Modalverb. Das heißt, es kann Präsens, Präteritum und Konjunktiv I und II angegeben werden. Falls Hauptsatzkanten ein Subjekt haben, wird dessen Kongruenz (in Numerus und Person) mit dem Subjekt geprüft.
- *Nominalgruppen* können ebenfalls gesamte Eingabe abdecken oder als Füller für Rollen vorkommen.
Zu einer Nominalgruppe kann eine Präposition gehören, gefolgt von einem Artikel oder einem als Artikel verwendeten Pronomen, gefolgt von einer Reihe von Adjektiven oder Adverb/Adjektiv-Kombinationen oder einer Zahl. Nach diesem Vorspann kommt das Substantiv, das den Kern (Kopf) der Gruppe bildet und dahinter kann sich ein Relativsatz anschliessen. Dahinter können Rollenfüller folgen, wenn sie in einer Fallschablone zu dem Kopfsubstantiv angegeben sind.
- *Adverben* können alleine eine Rolle füllen.
Es gibt die Arten Modaladverb, Lokaladverb und Temporaladverb.
- *Relativsätze* beginnen mit einem Relativpronomen und enden mit dem Vollverb.
Sie müssen sich unmittelbar an ihr Bezugswort anschliessen und es wird stets die Kongruenz des Relativpronomens mit diesem Bezugswort geprüft (Numerus und Geschlecht). Dieses Bezugswort und sein Vorspann wird als Rollenfüller in den Relativsatz eingebaut, und zwar in dem Kasus, den das Relativpronomen angibt. Relativsätze füllen keine Rolle, sondern können spontan (ähnlich wie Adjektive) bei einem Substantiv angegeben werden. Für die Instanziierung eines Relativsatzes stehen alle Fallschablonen zur Verfügung.
Ansonsten gilt das zu Hauptsätzen gesagte.
- *Nebensätze* beginnen mit einer Konjunktion und enden mit dem Vollverb.
Sie füllen eine Rolle. Ansonsten gilt das zu Hauptsätzen gesagte.

- *W-Fragen mit Substantiv* beginnen mit einer Frageformel und decken die gesamte Eingabe ab.
Sie können die Form einer Nominalgruppe oder eines Hauptsatzes haben. Bei der Form als Hauptsatz (z.B. **Welches Beispiel meinst Du ?**) füllt die Frageformel eine Rolle. Bei der Form als Nominalgruppe (z.B. **Welches Beispiel ?**) ist die Frageformel bereits das gesamte Ergebnis. Die Frageformel besteht aus einer Folge von Frageworten (meist Null oder eine Präposition und das Interrogativpronomen) und dem Substantiv, nach dem gefragt wird. Ein Fragezeichen wird zum Erkennen von Fragesätzen grundsätzlich nicht benötigt.
- *W-Fragen ohne Substantiv* in Hauptsatzform werden analog behandelt, jedoch wird beim Füllen einer Rolle keine Einschränkung bezüglich des Füllerkonzeptes gemacht, da ein solches ja nicht zur Verfügung steht. Eine Interpretation als Nominalgruppe wird (mangels Substantiv) nicht vorgenommen. Die Frageformel besteht hier nur aus einer Folge von Frageworten, z.B. **Links von was steht er ?**
- *Ja/Nein-Fragen* beginnen mit einem Verb und decken die gesamte Eingabe ab. Sie werden ansonsten genau analog den Hauptsätzen behandelt. Wie bei den W-Fragen ist auch hier kein Fragezeichen notwendig.
- *Imperativsätze* beginnen mit einem Verb im Imperativ und decken die gesamte Eingabe ab.
Diejenigen Rollen in der erkannten Fallschablone, die im Rollenverzeichnis als Subjektrolle gekennzeichnet sind, werden als nicht benötigt gestrichen und bei der Bewertung der Kante unberücksichtigt gelassen. Modalverben sind in Imperativsätzen nicht möglich; ansonsten gilt das zu Hauptsätzen gesagte.

4.8.2 Zerteilungsphasen

1. Der Zerteiler erhält als Eingabe einen String. Dieser wird in Teile zerlegt. Diese Teile sind Worte, Zahlen, Texte in Anführungszeichen, Zwischenräume und Satzzeichen. Worte werden im Wörterbuch nachgeschlagen. Sind sie dort bekannt, wird eine entsprechende Kante im Chart erzeugt, andernfalls entsteht eine Kante vom Typ 'unbekannt'.
Zahlen, Texte in Anführungszeichen und Satzzeichen werden direkt ins Chart eingetragen.
Zwischenräume dienen nur zur Trennung.
2. Das Chart wird dem eigentlichen Zerteiler übergeben. Der Zerteiler erweitert das Chart um neue Kanten, bis alle Bearbeitungsmöglichkeiten erschöpft sind. Im Erfolgsfalle wird dabei auch eine Menge von Instanziierungen von Fallschablonen erzeugt.
3. Diese Instanziierungen werden als Ergebnis in einer lesbaren Form ausgegeben².

Beim Nachschlagen der Wörter im Wörterbuch werden bereits Einschränkungen gemacht, um die Mehrdeutigkeit von Kanten zu verringern:

1. Fragewörter werden nur als Kante eingetragen, wenn vom ersten Wort an bei jedem Wort bis zur betreffenden Stelle die Interpretation als Fragewort möglich war.

²Wahlweise können auch die Kommentare der erkannten Fallschablonen und die Definitionen ihrer Kopfkonzpte als Resultat ausgegeben werden.

2. Großgeschriebene Wörter (außer am Satzanfang) werden ausschließlich als Substantive, Artikel oder Fragewörter interpretiert.³

4.8.3 Zerteilungsalgorithmus

Die nächsten Abschnitte enthalten eine Beschreibung des Zerteilungsalgorithmus. Diese Beschreibung will auf möglichst wenig Raum die wesentlichen Ideen des Algorithmus vermitteln, ohne so etwas wie eine genaue Spezifikation der Arbeitsweise auch nur anzustreben. Sie ist deshalb in einem sehr informellen Pseudocode verfaßt, der noch von verbalen Anmerkungen ergänzt wird. Ein großer Teil der in diesem Pseudocode verwendeten Operationen wird überhaupt nicht näher beschrieben; eine Detailierung erfolgt nur, wo sie zur Einsicht in die Funktionsweise unverzichtbar erscheint. Ein Teil der Beschreibung ist den zur Darstellung von Chart und Chartkanten verwendeten Datenstrukturen gewidmet, in der Hoffnung, das Wissen um diese Strukturen ver helfe dem Leser trotz aller Grobheit der Beschreibung zu einer erleichterten Einsicht in die algorithmische Konstruktion als Ganzes. Für genauere Einblicke muß auf den Quellcode des Programms verwiesen werden.

4.8.3.1 Generelles

Der Zerteilungsalgorithmus stellt einen Chart-Parser dar, das heißt alle Information befindet sich in *Kanten*. Siehe dazu auch Abschnitt 3.4.

Eine Kante hat den Zustand aktiv, erfolgreich oder erfolglos (fehlgeschlagen). Dabei stellt eine erfolgreiche Kante einen vollständig gefundenen Baustein für den Zerteilungsalgorithmus dar. Eine aktive Kante ist eine Hypothese über das Vorhandensein eines solchen Bausteins, die noch untersucht werden muß und eine erfolglose Kante ist eine ehemals aktive Kante, bei deren Untersuchung sich die betreffende Hypothese als falsch herausgestellt hat.

Ferner gibt es aus organisatorischen Gründen die Kantenzustände 'inaktiv' und 'aktuell': eine Kante, die nicht mehr selbst benötigt wird, kann in den Zustand 'inaktiv' gesetzt werden. Dies bewirkt im Hinblick auf die Effizienz, daß die Kante im Chart nicht mehr vorhanden zu sein scheint; es können jedoch Verweise auf die Kante bestehen bleiben. Wegen letzterer Notwendigkeit gibt es ein echtes Löschen in einem Chart grundsätzlich nicht. Mit dieser Methode kann man zum Beispiel am Ende einer Zerteilung, also nachdem man ein Ergebnis gefunden hat, noch vorhandene aktive Kanten quasi wegwerfen. Außerdem werden so Kanten entsorgt, von denen man bei der Entdeckung einer Mehrdeutigkeit mehrere Kopien gemacht und in einen anderen Zustand gesetzt hat. 'aktuell' sind schließlich diejenigen aktiven Kanten, die wirklich gerade bearbeitet werden; meist also nur eine.

Der Verarbeitungszyklus beim Chart-Parsen besteht in der wiederholten Auswahl einer aktiven Kante und deren Bearbeitung, wobei unter anderem neue aktive Kanten ins Chart eingetragen werden können und die aktive Kante 'erfolgreich' oder 'erfolglos' gesetzt werden kann. Das Eintragen einer neuen aktiven Kante unterbleibt, wenn an derselben Stelle bereits eine erfolgreiche oder erfolglose Kante zu der gleichen Hypothese im Chart eingetragen ist.

³Die Interpretation als Artikel dient zur Beherrschung großgeschriebener Pronomina.

Bei einem ‘reinrassigen’ Chart-Parser würden nur neue aktive Kanten vorgeschlagen, die zunächst die Länge Null haben oder es wird eine aktive mit einer direkt an sie anschließenden erfolgreichen Kante verbunden, wobei eine neue Kante entsteht, die aktiv oder erfolgreich sein kann. Von dieser Arbeitsweise wird jedoch hier abgewichen: Es ist zusätzlich möglich, daß eine aktive Kante mit einer erfolgreichen verbunden wird, *ohne* daß eine neue Kante entsteht (die Kante wird also verlängert), wenn sichergestellt ist, daß dennoch alle Mehrdeutigkeiten ausgeschöpft werden.

Die Zerteilung ist beendet, sobald es keine aktiven Kanten mehr gibt. Nach Ende der Zerteilung sollte es eine oder mehrere erfolgreiche Kanten über die volle Länge des Eingabesatzes geben; diese stellen das Ergebnis oder (bei Mehrdeutigkeit) die Ergebnisse dar. Falls es keine Kante über die gesamte Länge gibt, werden die besten Teillösungen präsentiert.

Daraus ergibt sich auf der äußersten Ebene folgender Algorithmus:

zerteilen:

```

baue chart c auf;
while (noch mindestens eine aktive kante in c vorhanden)
  waehle eine aktive kante k aus;
  bearbeite die aktive kante k;
end;
gib ergebnisse aus.

```

Beim Bearbeiten des Charts werden Instanzen von Fallschablonen gesucht und daraus Instanziierungen von Fallschablonen erzeugt. Zum leichteren Verständnis folgen hier grobe Definitionen dieser Begriffe im Sinne der folgenden Abschnitte:

Definition “Fallschablone”

Eine Fallschablone ist ein Datenobjekt der Wissensbasis. Sie gibt eine abstrakte Beschreibung einer Klasse von Eingaben des Benutzers. Fallschablonen werden vom Zerteiler nur gelesen, jedoch nicht verändert. Verbale Fallschablonen beschreiben Haupt- oder Nebensätze wie **Ich sehe das Buch im Schrank**, nominale Fallschablonen beschreiben Substantive mit Rollenfüllern als Attributen, wie **das Buch im Schrank**.

Definition “Instanz”

Eine Instanz einer Fallschablone ist eine Eingabe oder ein Teil einer Eingabe des Benutzers. Sie gehört zu der Klasse von Eingabesätzen, die die Fallschablone beschreibt. Instanzen werden vom Benutzer eingegeben und vom Zerteiler lediglich gesucht, jedoch nicht verändert.

Definition “Instanziierung”

Eine Instanziierung einer Fallschablone ist ein Datenobjekt des Zerteilers. Sie beschreibt, wie eine Instanz auf eine Fallschablone abgebildet wird. Instanziierungen werden vom Zerteiler aufgebaut und verändert und bilden das Ergebnis des Analyseprozesses. Auch der Vorgang des Aufbaus einer Instanziierung wird als Instanziierung bezeichnet.

Instanziierungen sind Kanten im Chart; sie haben deshalb zu jedem Zeitpunkt einen bestimmten Anfangs- und Endpunkt im Eingabesatz.

Es gibt also eine Korrespondenz zwischen Instanzen (Benutzereingabe, externe Form) und Fallschablonen (Wissensbasis, interne Form). Diese Korrespondenz wird beschrieben von einer Instanziierung; jene aufzufinden ist die Aufgabe des Zerteilers.

4.8.3.2 Information in den Kanten

Ein Chart besteht hauptsächlich aus einer Liste von Verweisen auf die aktiven Kanten und einer Reihung von Knoten, wobei jeder Knoten eine Liste von Verweisen auf Kanten ist. Von zentraler Bedeutung für die Arbeitsweise des Zerteilers ist die in den Kanten darstellbare Information. Eine Kante enthält im Wesentlichen folgende Information:

- einen Kantenzustand **zustand**. Also einen der Werte erfolgreich, aktiv, fehlgeschlagen, aktuell, inaktiv.
- einen Verweis **eigner** auf das Chart, in das die Kante gehört.
- eine Knotennummer **start**, die den Knoten in diesem Chart angibt, an dem die Kante beginnt. (Der erste Knoten hat die Nummer 0)
- eine Zahl **laenge**, die angibt, wieviele Worte die Kante überspannt.
- eine Typkennzeichnung **typ** und ein Informationsobjekt **e1** (für “Element”), die die eigentliche Information der Kante enthalten: Der Kantentyp **typ** gibt an, welche Art von Kante vorliegt. Es gibt insbesondere z.B. die Werte **wort**, **rollenfueller** und **instanziierung**. In **e1** befindet sich die eigentliche kantenspezifische Information. Diese ist abhängig vom Kantentyp. Bei **wort**-Kanten steht hier das Wort selbst und die zugehörige Beugungsinformation (je nach Wortart einige der Angaben Numerus, Kasus, Geschlecht, Zeit, Komparationsstufe). Bei **rollenfueller**-Kanten ist dies der Name der betroffenen Rolle und bei **instanziierungs**-Kanten eine Reihe von unterschiedlichen Informationen, wie unten beschrieben.
- eine Liste **subkanten** der von dieser Kante zusammengefaßten anderen Kanten. Insbesondere ist dies für Instanziierungen die Liste der Rollenkanten (gefüllte Rollen).
- eine Zahl **guetewert**, die die Beurteilung der Qualität dieser Kante enthält. Dies ist vor allem für Instanziierungen von Bedeutung.

Nur die Chartoperationen haben die Berechtigung, Kanten zu erschaffen oder zu kopieren sowie ihren Start, ihre Länge und ihren Zustand zu verändern. Dadurch ist das Chart immer über die Lage auf dem Kantenmarkt informiert. Die so erreichte Kontrolle über die Kanten dient dazu, den Zerteilungsprozeß sauber zu *steuern*. Mit den Komponenten **subkanten** und **e1** werden die eigentlich informationsgewinnenden Operationen beim Zerteilen *durchgeführt* und die Ergebnisse bewahrt.

Die Information, die im **e1**-Feld von Instanziierungskanten steht, läßt sich etwa wie folgt zusammenfassen:

```
bool          verbal;    /* verbale oder nominale Fallschablone */
kantenverweis kopf;      /* Kante des Kopfkonzpts */
knotennummer ende;      /* Endpunkt der Instanz (spätestens) */
list(fallschablonenrolle) offene_rollen; /* noch unbenutzte Rollen */
```

```

/* nur bei verbal relevant: */
wortinfo      konjunktion; /* bei Nebensaetzen */
wortinfo      modalverb;  /* evtl. vorhanden */
satzart       art;        /* deklarativ, W-Frage etc. */
zeitangabe    zeit;       /* Tempus des Satzes */
list(kantenverweis) schwestern;
/* nur bei nominal relevant: */
wortinfo      wi;         /* (sprich: Wee-Ihh) */
wortinfo      praeposition;
list(kantenverweis) vorworte;
bool          hat_fragepronomen;

```

Es dient dabei **verbal** zur Unterscheidung verbaler und nominaler Instanziierungen von Fallschablonen. Bei verbalen tritt als **kopf** eine Verbkante (d.h. eine Kante vom Typ **wort**, deren **wortinfo** als Wortart 'Verb' angibt) auf, bei nominalen eine Substantivkante. **ende** enthält die Vermutung darüber, an welcher Stelle in der Eingabe (angegeben als Knotennummer) spätestens mit dem Instanzieren aufgehört werden muß. Die Liste der offenen Rollen enthält zu jedem Zeitpunkt diejenigen Rollen, die in der Fallschablone vorgesehen sind, jedoch bisher noch nicht gefüllt wurden.

Es gibt nun im Weiteren zwei Gruppen von Angaben, von denen eine nur bei der verbalen Instanziierung einer Fallschablone relevant ist, die andere nur bei der nominalen.

Die erste Gruppe (für verbale Instanziierungen) enthält Angaben über die globalen Eigenheiten des abgedeckten Satzes: Die Aussageweise (Deklarativsatz, Befehlssatz, W-Frage, Ja/Nein-Frage, Nebensatz), die Konjunktion (im Falle eines Nebensatzes, der kein Relativsatz ist) und eventuell ein verwendetes Modalverb.

Die zweite Gruppe (für nominale Instanziierungen) beschreibt die Nominalphrase, die den Hauptteil der Instanziierung bildet ("Kopfphrase"). Dazu gehört die grammatikalische Gesamterscheinungsform (Beugungsinformation in **wi** und ggf. eine Präposition in **praeposition**, das zentrale Substantiv (ebenfalls in **wi**) und die Liste **vorworte** der dem Substantiv vorangehenden Worte (Artikel, Adverben, Adjektive, Zahlen). Schließlich findet sich hier die Information, ob die Instanz das nachgefragte Objekt in einer Frageformel sein kann. Dies ist der Fall, wenn das zentrale Substantiv ein Interrogativpronomen vor sich hat, was durch **hat_fragepronomen** angegeben wird.

4.8.3.3 Initialisierung und Kantenauswahl

Die aktiven Kanten werden mit einigen Ausnahmen nach Art eines Stapels verwaltet. Es wird immer die zuletzt in den Zustand 'aktiv' versetzte Kante als nächste weiterbearbeitet. Beim Weiterbearbeiten erhält die Kante zunächst automatisch den Zustand 'aktuell'. Sie kann jedoch im Zuge der Bearbeitung wieder 'aktiv' gesetzt werden. Werden danach, jedoch vor Abschluss der Bearbeitung dieser Kante, noch eine oder mehrere andere Kanten 'aktiv' gesetzt (seien es nun neue oder bereits existierende oder auch solche, die bereits 'aktiv' sind) so werden also diese nach Abschluß des gerade laufenden Arbeitsschrittes als erste bearbeitet. Es kann auch sein, daß keine Kante 'aktiv' gesetzt wird, was dazu führt, daß die nächstältere aktive Kante als nächste bearbeitet wird.

Um mit diesem Prozess beginnen zu können, muß beim Aufbauen des Chart mindestens eine aktive Kante eingetragen werden; dies geht so vor sich:

baue chart c auf:

```

trage alle wortkanten ein; /* Worte, "Strings", Zahlen */
for (alle verben v in irgendeiner Reihenfolge) do
  trage verbale fallschablone ein;
end;
if (erstes wort ist fragewort)
  trage aktive frageformelkante ein;
end;
for (alle substantive s von hinten nach vorn) do
  trage nominale fallschablone ein;
end.

```

Jedes Eingabewort hat potentiell laut Wörterbuch eine Reihe von Interpretationen. Jede dieser Interpretationen kann bei manchen Wortarten, insbesondere Verb und Substantiv, laut Konzepthierarchie eine Reihe von zugehörigen Konzepten eintragen haben. Mit *v* und *s* sind oben diese Konzepte gemeint. Das Eintragen einer Fallschablonen- oder Frageformelkante bedeutet dabei jeweils die Erzeugung einer aktiven Kante mit Länge 0 und Startpunkt bei dem Konzept. Diese aktive Kante wird dann später instanziiert. Das Eintragen geschieht ungefähr so:

trage verbale fallschablone ein:

```

erzeuge fallschablonenkante f aus (v);
f.el.inst.verbal = true.

```

trage nominale fallschablone ein:

```

erzeuge fallschablonenkante f aus (s);
f.el.inst.verbal = false.

```

erzeuge fallschablonenkante f aus (w):

```

erzeuge neue kante k im chart beim startpunkt von w;
initialisiere k als leere instanziiierung;
k.el.inst.kopf = w;
k.el.inst.offene_rollen = sammle rollen (w).

```

Das Sammeln von Rollen bedeutet die Bildung der Fallschablone aus dem Kopfkonzept. Diese Fallschablone ergibt sich, indem man die unmittelbar in einer gleichnamigen Fallschablone (falls vorhanden) angegebenen Rollen nimmt, dazu alle, die dort mit der `:erbe`-Klausel geerbt werden und dazu, sukzessive die Konzepthierarchie hinaufsteigend, alle Rollen, die von Fallschablonen für die Oberkonzepte des betrachteten Kopfkonzepths angegeben werden. Dies geschieht etwa in folgender Weise:

sammle rollen (w):

```

list(fallschablonenrolle) resultat = nil;
if (w hat fallschablone)
  resultat = w.eigene_rollen;
  /* d.h. inklusiv ausdruecklich Geerbte von kopflosen Fallschablonen */
end;
ergaenze resultat in breitensuche um rollen der oberkonzepte;
return (resultat).

```

Es werden also auch dann Instanzierungskanten erzeugt, wenn das betreffende Konzept gar keine eigene Fallschablone in der Fallschablonenmenge der Wissensbasis hat. Die

Rollen werden von den Oberbegriffen des Konzepts geerbt, wobei bereits vorhandene Rollen ignoriert, d.h. weder überschrieben noch mehrfach aufgenommen werden. Im Extremfall entsteht dabei eine leere Liste von offenen Rollen; die Instanziierung ist dann sofort zu Beginn ihrer Bearbeitung vollständig. Jede Rolle enthält Information über die erwartete grammatikalische Form (durch Angabe eines Rollennamens aus dem Rollenverzeichnis) und über die Art des erwarteten Füllers (durch Angabe eines Füllerkonzepts).

Entscheidend beim Aufbau des Chart ist die Reihenfolge, in der die aktiven Kanten aktiviert werden: durch die Stapelverwaltung der aktiven Kanten werden die zuletzt aktivierten zuerst bearbeitet, das heißt es wird als erstes versucht die nominalen Instanziierungen zu vervollständigen.

Die Kantenauswahl ist trivial:

wähle eine aktive kante k aus:

```
k = pop (aktive);
setze_zustand (k, aktuell).
```

4.8.3.4 Instanzieren von Fallschablonen

Die Instanziierung von Fallschablonen wird jeweils beim Bearbeiten der ausgewählten aktiven Kante vorangetrieben. Dabei gibt es folgende Fälle:

1. Beim Besuch einer Frageformelkante wird die zugehörige Frageformel gebildet.
2. Beim ersten Besuch einer Kante, die eine nominale (oder pronominale) Instanziierung ist, muß zunächst die Nominalphrase selbst gebildet werden, d.h. es werden Adjektive, Adverbien, Artikel und Präposition (jeweils sofern vorhanden), die vor dem Kopfsubstantiv oder -pronomen der Instanziierung stehen, aufgesammelt und in der Instanziierung vermerkt. Bei diesem Prozess können durch Wortmehrdeutigkeiten mehrere Kanten aus derselben Anfangskante entstehen.
3. Beim zweiten Besuch einer nominalen Instanziierung wird untersucht, ob der Nominalphrase vielleicht ein Relativsatz folgt. Falls ja, wird dessen Instanziierung angestoßen und der Besuch beendet. Auch dies ist mehrdeutig, falls nämlich mehrere Verben als potentielle Köpfbegriffe des Relativsatzes zur Verfügung stehen.
4. Ansonsten wird versucht, die Instanziierung zu verlängern, falls nicht das Ende bereits erreicht ist. Für die Verlängerung wird jede Kante untersucht, die vom aktuellen Endknoten der Instanziierungskante ausgeht und wann immer sie sich anfügen läßt, wird eine Kopie der Instanziierung gemacht und die Kante angefügt. Auch dieser Prozeß ist also eventuell mehrdeutig.
5. Falls zwar das Ende noch nicht erreicht ist, aber dennoch keine sinnvolle Verlängerung gefunden werden konnte, wird untersucht, ob sich möglicherweise hier ein Nebensatz anschließt (wiederum evtl. mehrdeutig). Falls ja, wird dessen Instanziierung angestoßen und die aktuelle Kante solange zurückgestellt. Wenn sie dann später wieder aktuell wird, ist die Instanziierung des Nebensatzes abgeschlossen und die betreffende Kante kann als normale Verlängerung eingefügt werden.

6. Ansonsten kann die Kante nicht fortgesetzt werden und wird nun entweder verworfen, weil sie unvollständig ist, oder als erfolgreiche Instanziierung im Chart fixiert.

In der Pseudocode-Schreibweise sieht diese Prozedur folgendermaßen aus:

bearbeite die aktive kante k:

```
if (k ist fragekante)
  bilde fragekante in k;
  return;
end;
```

Das war der erste Fall, die Bildung einer Fragekante. Es folgt nun die Untersuchung auf den ersten Besuch bei einer nominalen Instanziierung:

```
if (ist erster besuch bei dieser kante)
  if (k ist verbale instanziierung)
    ; /* nichts tun */
  elsif (k ist pronominale instanziierung)
    lese pronominalphrase (k);
  else
    lese nominalphrase (k);
  end;
return;
```

Als nächstes muß im Falle eines zweiten Besuches bei einer nominalen Kante versucht werden, ggf. einen Relativsatz zu instanzieren:

```
elsif (ist zweiter besuch bei dieser kante und
       k ist nominale instanziierung)
  versuche relativsatz anzustossen;
  if (habe was angestossen)
    return;
  end;
end;
```

Hier nun der normale Fall: Die Kante soll noch fortgesetzt werden, wenn das Eingabeende oder das vorberechnete Ende noch nicht erreicht ist. Es wird dann jede Kante, die am augenblicklichen Endpunkt der Instanziierung beginnt, also als Fortsetzung in Frage kommt untersucht. Jede dieser Kanten, die von ihrem Inhalt her angefügt werden kann, wird an eine eigens dafür erstellte Kopie der aktuellen Kante angefügt. Durch dieses Kopieren wird die Mehrdeutigkeit technisch gehandhabt; das Original der Kante wird am Ende vernichtet. Folgende Unterscheidungen werden gemacht:

```
if (will noch verlaengern)
  for (alle kanten f am endknoten von k) do
    if (f ist erfolgreich und vom typ fallschablonenkante)
      fuehle fallschablone f in k ein;
    elsif (f ist wortkante fuer verb)
      fuege praedikatsteil p in k ein;
    elsif (f ist vom typ 'unbekannt')
```

```

    fuege unbekannte kante u in k ein;
  elsif (f ist erfolgreich und vom typ fragekante)
    fuehle fragekante f in k ein;
  else /* sonstige Wort- u. Konzeptkanten, erfolglose K. etc.*/
    ignoriere die kante;
  end;
end;
end;
end;

```

Es folgt nun die Untersuchung darauf, was in der obigen Schleife geschehen ist, und die entsprechende Reaktion:

```

if (ich wollte verlaengern und konnte nicht)
  versuche nebensatz anzustossen;
elsif (wollte nicht mehr verlaengern)
  bearbeite instanziierung k nach;
else /* habe verlaengert */
  setze zustand (k, inaktiv); /* Kopien werden weitergefuehrt */
end.

```

Hier nun die genauere Beschreibung der oben erwähnten Operationen. Zuerst das Bilden von Fragekanten und das Lesen der Nominalphrasen:

bilde fragekante in k:

```

while (naechstes wort w ist fragewort)
  /* Mehrdeutigkeiten gibts keine */
  verlaengere k um 1 und nimm w mit auf;
end;
if (es folgt nominalphrase np mit hat_fragepronomen)
  verlaengere k um np und nimm np mit auf;
end.

```

lese pronominalphrase (k):

```

for (jede kante v am knoten vor dem startknoten von k) do
  if (v.typ = wortkante und v.wi.wortart = praeposition)
    neu = kopie von k;
    nimm v mit in neu auf;
    setze zustand (neu, aktiv);
  end;
end;
if (mindestens eine neue kante erzeugt)
  setze zustand (k, inaktiv);
end.

```

Das Lesen von Nominalphrasen wird hier nur sehr ungenau angegeben. Es folgt unten eine verbale Erläuterung:

lese nominalphrase (k):

```

n = startknoten von k;
if (n = 0)
  return; /* mehr geht nicht */

```

```

for (jede kante w beim knoten n) do
  /* genaueres siehe untenstehenden Text */
  if (w ist wortkante und wortart ist passend und
      kongruenz nicht verletzt)
    neu = kopie von k;
    fuege wort w vorn zu neu hinzu;
    setze zustand (neu, aktiv);
    if (keine mehrdeutigkeit zu befuerchten)
      setze zustand (k, inaktiv);
    lese nominalphrase (neu);
  end;
end.

```

Etwas genauer gesagt geht `lese nominalphrase` so: Sammle vom Substantiv aus nach vorne vorgehend Paare von Adverb/Adjektiv oder einzelne Adjektive, Zahlen, dann ggf. einen Artikel und dann ggf. eine Präposition auf. Dies behandelt auch Demonstrativpronomen und Possessivpronomen. Bei Wortmehrdeutigkeiten wird die Kante entsprechend kopiert, so daß mehr als eine Kante als Ergebnis entstehen kann. Das Auf-sammeln ist beendet, wenn eine Präposition gefunden ist oder ein Wort einer nicht oder nicht mehr passenden Wortart auftaucht oder durch den vermeintlichen Artikel die Kongruenz verletzt wird.

In der Kante wird Fall und Präposition vermerkt; diese Information dient, zusammen mit dem Konzept, zu dem das Substantiv gehört, dazu, festzulegen, welche Rollen sie füllen kann. Der Gütewert der Kante wird auf die Anzahl von Worten in der Phrase gesetzt, für Phrasen die bei Knoten Null beginnen und solche, die eine Präposition enthalten, wird der Gütewert nochmals um 1 erhöht.

4.8.3.5 Das Startpunktproblem

Bei der Instanziierung der Fallschablonen besteht ein Problem darin, den Punkt im Eingabesatz zu finden, an dem die Instanz der gerade betrachteten Fallschablone beginnt und den Punkt, wo sie endet.

Der Endpunkt kann bei nominalen Instanziierungen dadurch festgestellt werden, daß alle Rollen gefüllt sind; bei verbalen Instanziierungen ist der Endpunkt entweder das Eingabeende (Hauptsatz) oder er liegt hinter dem Prädikat.

Der Startpunkt muß aber bei Beginn der Instanziierung bereits festliegen. Bei nominalen Fallschablonen ist der Beginn stets der Anfang der (s. oben) Nominalphrase. Bei verbalen Fallschablonen ist dieser Punkt jedoch nicht im Voraus festzustellen, da die Stellung des Hauptverbs keine Rückschlüsse auf den Beginn des zugehörigen Satzes zuläßt. Es gibt zwei Fälle: bei Hauptsätzen ist der Beginn immer auch der Beginn der Eingabe, bei Nebensätzen ist der Beginn hinter einer Konjunktion oder einem Relativpronomen irgendwo im Innern. Es muß bei einem Nebensatz also erst diese einleitende Konjunktion oder das Relativpronomen gefunden werden. Dies kann man nicht ohne Weiteres mit linearer Rückwärtssuche vom Satzende aus tun, da Nebensätze verschachtelt sein können und zudem Relativpronomen massiv mehrdeutig mit Artikeln sind.

Die hier verwendete Lösung des Problems ist folgende: Eine aktive Kante für die Instanziierung einer verbalen Fallschablone wird zunächst stets als Hauptsatzkante eingetragen.

Diese Kante wird zuerst zu instanziiieren versucht. Es wird jedoch die “Urform” dieser Kante, die sogenannte *Schwesterkante*, aufbewahrt und die Hauptsatzkante erhält einen Verweis darauf. Gelingt die Instanziierung der Hauptsatzkante, so wird die Schwesterkante ‘erfolglos’ gesetzt. Bis dahin steht aber noch die Schwesterkante der Fallschablone für die Instanziierung mit einem Relativsatz oder sonstigen Nebensatz zur Verfügung, wobei der Startpunkt dynamisch zu bestimmen ist.

richte schwesterkante von k ein:

```
k_neu = kopie von k;
k_neu.el.inst.schwestern = list(k); /* einelementige Liste */
k.el.inst.schwestern = list(k_neu); /* dito */
setze_start (k, 0);
setze k_neu ganz hinten in den stapel der aktiven kanten;
setze k vorn in den stapel der aktiven kanten.
```

Diese Schwesterkanten können nun auf folgende Weise in den Instanzierungsprozeß einbezogen werden: kommt eine Instanzkante an einen Punkt, wo keine passende erfolgreiche Kante zu ihrer Fortsetzung zur Verfügung steht (s. unten), so untersucht sie die noch nicht gefüllten Rollen daraufhin, ob eine von ihnen eine Fallschabloneninstanz als Füller erwartet. Ist dies der Fall, so wird angenommen, das genau diese Instanz an der aktuellen Stelle beginnt. Der Zerteiler versucht, unter den Schwesterkanten eine zu finden, die die erwartete Fallschablone (oder eine damit laut Konzepthierarchie verträgliche) trägt und setzt den Anfangspunkt einer Kopie davon auf die aktuelle Stelle. Dies alles geschieht in **versuche nebensatz anzustossen**. Nun wird die weitere Instanziierung der aktuellen Kante unterbrochen und als nächstes versucht, diese neue Kante zu instanziiieren. Dabei kann sich gegebenenfalls dieser Prozeß wiederholen.

versuche nebensatz anzustossen:

```
for (alle schwesterkanten s) do
  if (s koennte irgendeine rolle von k fuehlen)
    k_neu = kopie von s;
    setze zustand (k_neu, aktiv);
    setze start (k_neu, ende von k);
  end
end
if (kein k_neu gefunden)
  bearbeite instanziierung k nach;
end.
```

Die zweite Möglichkeit zur Aktivierung der Schwesterkanten ergibt sich bei Relativsätzen. Diese können jeweils nur direkt hinter der einleitenden Nominalphrase einer Nominalinstanziierung stehen und sind nicht als Rollenfüller in der Fallschablone vermerkt. Darum wird beim Auffinden eines richtig gebeugten Relativpronomens an dieser Stelle eine Kopie von *jeder* vorhandenen Schwesterkante ins Rennen geschickt, wobei die Nominalphrase, die dem Relativsatz vorangeht gleich als erstes mit eingefüllt werden muß. Dazu wird aus dieser Nominalphrase künstlich eine andere erzeugt, die den Kasus des Relativpronomens trägt, ansonsten aber übereinstimmt.

versuche relativsatz anzustossen:

```
for (alle wortinterpretationen w des naechsten wortes) do
  if (w ist ein relativpronomen und
```

```

        w ist in geschlecht und numerus wie vorherige nominalphrase)
        stosse relativsatz an;
    end;
end.

```

stosse relativsatz an:

```

for (alle schwesterkanten s) do
    neu = kopie von s;
    setze_start (s, ende von k);
    setze_laenge (s, 1); /* bis hinter das Pronomen */
    fuelle modifizierte nominalphrase von k in neu ein;
    /* Endknoten von neu ist jetzt hinter dem Relativpronomen */
    setze zustand (neu, aktiv)
end.

```

4.8.3.6 Erweitern einer Instanziierung

Erweitern einer Instanziierung bedeutet das Anfügen einer Kante, so daß die Instanziierungskante länger wird. Kanten, die für das Anfügen in Frage kommen, sind

1. Andere Instanziierungen. Dies sind entweder Relativsätze, die ohne Rollenzuordnung aufgenommen werden, oder Nominalinstanziierungen bzw. Nebensätze, die eine Rolle füllen.
2. Teile des Prädikats, also Verben, Hilfsverben, Modalverben oder Verbpräfixe.
3. Fragekanten. Auch diese füllen eine Rolle.
4. "unbekannt"-Kanten. Diese werden einfach an der betreffenden Stelle in die Instanziierung aufgenommen, damit die Instanziierung weitergehen kann.

fuelle fallschablone f in k ein:

```

if (f ist relativsatz)
    /* fuelle f direkt in k ein: */
    neu = kopie von k;
    ergaenze f bei subkanten von neu;
    verlaengere neu um f;
    setze zustand (neu, aktiv);
else
    for (alle offenen rollen r) do
        if (f hat grammatikalische form die bei r moeglich ist und
            (kopf von f ist erwarteter fueller von r oder
             unterkonzept davon oder erlaubterweise oberkonzept davon))
            erzeuge rollenkante rk mit rolle r und subkante f;
            setze start und laenge von rk wie bei f;
            setze zustand (rk, erfolgreich);
            neu = kopie von k;
            ergaenze rk bei subkanten von neu;
            verlaengere neu um rk;
        end;
    end;
end.

```

```

    streiche rolle r aus neu.offene_rollen;
    setze zustand (neu, aktiv);
  end;
end;
end.

```

Prädikatsteile führen zur Bestimmung der Satzeigenschaften Zeit, Person, Numerus, Modalfärbung, Präfix und vor allem Satzart:

fuege praedikatsteil p in k ein:

```

if (p ist hilfsverb)
  /* kann ich nicht */
else /* modalverb oder vollverb */
  if (p ist vollverb aber nicht das zur fallschablone gehoerende)
    return;
  if (k beginnt bei knoten 0)
    if (p hat imperativinterpretation und
        eingabesatz hat kein fragezeichen am schluss)
      neu = kopie von k;
      setze neu auf imperativinterpretation;
      trage p als modalverb oder vollverb in neu ein;
      setze zustand (neu, aktiv);
    end;
    if (p hat nicht nur imperativinterpretation und
        eingabesatz hat kein ausrufezeichen am schluss)
      neu = kopie von k;
      setze neu auf ja/nein-frage-interpretation;
      trage p als modalverb oder vollverb in neu ein;
      setze zustand (neu, aktiv);
    end;
  else /* p ist nicht ab knoten 0, d.h. nicht erstes Eingabewort */
    neu = kopie von k;
    if (k hat nicht schon w-frage-interpretation)
      setze neu auf deklarativsatz-interpretation;
    end;
    trage p als modalverb oder vollverb in neu ein;
    setze zustand (neu, aktiv);
  end;
end.

```

Eine Kante vom Typ “unbekannt” kennzeichnet ein Wort, das nicht im Wörterbuch aufgefunden werden konnte. Es ist klar, daß es dafür keine sinnvolle Einbaumöglichkeit geben kann. Aus diesem Grund werden solche Kanten einfach an die Instanziierung “drangepappt”:

fuelle unbekannte kante u in k ein:

```

neu = kopie von k;
fuege u zu neu.subkanten hinzu;
verlaengere neu um u;
setze zustand (neu, aktiv).

```

Eine Fragekante wird genauso als Füller einer Rolle einzutragen versucht, wie eine nominale Instanziierung. Dabei kann allerdings bei schwach spezifizierten Fragen nicht geprüft werden, ob die Kante zu dem erwarteten Konzept passt, da sie ja kein Konzept enthält:

fülle fragekante f in k ein:

```

if (f laesst sich als rolle r in k einfüllen)
  erzeuge rollenkante rk fuer rolle r im chart;
  setze start und laenge von rk wie bei f;
  setze f als subkante von rk;
  setze zustand (rk, erfolgreich);
  neu = kopie von k;
  ergaenze rk bei subkanten von neu;
  verlaengere neu um rk;
  streiche rolle r aus neu.offene_rollen;
  vermerke in neu dass es sich um eine frage handelt;
  setze zustand (neu, aktiv);
end.

```

f laesst sich als rolle r in k einfüllen:

```

/* nicht prozedurales, sondern deklaratives Programmstueck! */
if (es gibt eine Rolle r fuer die gilt:
    (k.offene_rollen enthaelt r und
     substantiv von f passt ggf als fueller zu r und
     form von f entspricht einer frageform von r))
  return (true);
else
  return (false);
end.

```

4.8.3.7 Abschluß und Auswertung einer Instanziierung

Wird eine Instanziierung aus einem der oben aufgeführten Gründe nicht mehr erweitert, so wird noch eine Abschlußbehandlung durchgeführt. Dabei wird die Güte dieser Instanziierung aus der augenblicklichen Güte neu berechnet, indem für jede Rolle, die nicht gefüllt worden ist, die Gütebewertung verringert wird, und zwar um jenen Betrag, den die Dringlichkeit dieser Rolle laut Fallschablone oberhalb der Standard-Dringlichkeit liegt (vorausgesetzt diese Differenz ist positiv). Die Standard-Dringlichkeit ist 20.

bearbeite instanziierung k nach:

```

for (alle rollen r in k.el.inst.offene_rollen) do
  if (r.dringlichkeit > std_dringlichkeit)
    k.guetewert = k.guetewert - (r.dringlichkeit - std_dringlichkeit);
  end;
end;
if (k.guetewert > 0 und
    (k ist nicht verbal oder k hat subjekt-praedikat kongruenz))
  setze zustand (k, erfolgreich)
else

```

```

    setze zustand (k, fehlgeschlagen);
end.

```

k hat subjekt-praedikat kongruenz:

```

for (alle rollenfueller r in subkanten von k) do
  if (r hat subjektmarkierung)
    if (r.fueller in person oder numerus nicht kompatibel mit k.verb)
      return (false);
    else
      return (true);
    end;
  end;
end;
return (true).

```

Alle erfolgreichen Instanziierungen sind jetzt mögliche Analyseergebnisse. Als Lösungen werden alle Instanziierungen angegeben, die am Eingabeanfang beginnen und maximale Länge haben.⁴ Die Ausgabe ist sortiert zuerst nach fallender Länge und dann nach fallendem Gütewert.

gib ergebnisse aus:

```

sortiere am knoten 0 die erfolgreichen instanziierungen;
for (alle erfolgreichen instanziierungen k) do
  if (k hat volle laenge oder kein k mit voller laenge gehabt)
    gib die instanziierung k aus;
  end;
end.

```

Dieses Ausgabeformat ist im Abschnitt 4.9 beschrieben.

4.8.4 Zeitkomplexität

Der Zeitaufwand für die Analyse eines Eingabesatzes hängt ab von der Anzahl der Rollenfüller und sonstigen Teile, in die die Eingabe zerlegt wird, der Anzahl von Rollen in den instanziierten Fallschablonen und der Häufigkeit von Mehrdeutigkeiten bei der Zuweisung der Teile in den Instanziierungen. Die Länge der Eingabe und die Größe der Wissensbasis gehen darüber nur mittelbar in die Laufzeit ein. Das theoretische Zeitverhalten dieses Zerteilungsalgorithmus läßt sich damit wie folgt beschreiben:

- Der optimale Fall liegt dann vor, wenn ein langer Eingabesatz mit $O(n)$ Worten, die n Rollenfüller bilden, nur so instanziiert werden kann, daß sich jeder Rollenfüller sofort nur in genau eine Rolle einfüllen läßt und die zugehörige Fallschablone auch genau n Rollen aufweist. In diesem Fall muß für n Füller die Verträglichkeit mit im Mittel $n/2$ Rollen geprüft werden. Eine solche Prüfung verlangt im Idealfall nur eine konstante Zahl von Operationen.

Dazu kommt der Aufwand für das Aufbauen des Chart. Dieser ist im optimalen Fall proportional zur Länge der Eingabe.

Daraus ergibt sich eine Zeitkomplexität im besten Fall von $O(n^2)$, wobei n die Zahl von Rollenfüllern in der Eingabe ist.

⁴Genauer: Falls es mindestens eine Kante gibt, die die gesamte Eingabe abdeckt, so werden nur Kanten dieser Art ausgegeben, ansonsten alle.

- Im mittleren Fall enthält bei einem Fallschablonensystem die Fallschablone vereinbarungsgemäß “starke” semantische Einschränkungen, so daß die Zahl von zu verfolgenden Instanziierungen nur um einen konstanten Faktor ansteigt. Allerdings erhöht sich der Aufwand für den Test, ob ein einzelner Rollenfüller zu einer Rolle passt: nachdem sich der Füller als von seiner grammatikalischen Form her passend für die Rolle erwiesen hat, muß seine semantische Verträglichkeit in der Konzepthierarchie geprüft werden. Der Aufwand hierfür ist im Normalfall (d.h. bei Mißerfolg) jeweils genauso groß, wie der zum Aufbauen einer Ober- und Unterkonzeptliste nötige.

Im mittleren Fall kann man annehmen, daß die Länge dieser Listen logarithmisch mit der Gesamtgröße der Konzepthierarchie, nennen wir sie k , zunimmt. Jeder der n Arbeitsschritte steigt deshalb in seinem Aufwand von n auf $n \cdot \log k$.

Beim Aufbauen des Charts werden für jedes Konzept die Ober- und Unterbegriffslisten aufgebaut und für die Fallschablonen die Rollen aufgesammelt. Die dabei verwendeten Operationen zur Mengenvereinigung sind einfach programmiert und haben deshalb einen Aufwand proportional zum Produkt der Kardinalitäten der beteiligten Mengen. Das Aufbauen des Chart dauert deshalb $O(n \cdot \log^2 k)$.

Es ergibt sich insgesamt ein Aufwand von $O(n^2 \cdot \log k + n \cdot \log^2 k)$.

- Im schlechtesten Fall ist jeder Begriff der Konzepthierarchie im Unterbegriffspfad jedes Konzeptes in der Eingabe enthalten, das Aufbauen des Chart dauert also $O(n \cdot k^2)$, und jeder Rollenfüller ist mit jeder Rolle verträglich, so daß n Instanziierungen der Restlänge $n - 1$ verfolgt werden, aus jeder von denen $n - 1$ Instanziierungen der Restlänge $n - 2$ entstehen und so weiter. Am Schluß liegen $n!$ Ergebnisse vor, die in $n!$ Arbeitsschritten entstanden sind. In diesem Fall wäre also die Speicherkomplexität $O(n!)$ und die Zeitkomplexität $O(n \cdot \log^2 k + n!)$.

In der Praxis sind alle diese Betrachtungen allerdings von geringer Bedeutung: einerseits ist der asymptotische Aufwand für sehr große Eingabelängen weitgehend uninteressant, weil es sich bei **SARA** um ein Dialogsystem handelt, bei dem stets nur mit kleinen oder sehr kleinen Eingabelängen gearbeitet wird, andererseits ist gerade wegen des Dialogbetriebs der konstante Faktor beim Zeitverhalten von großer Bedeutung. Versuche ergeben hier für **SARA** ein befriedigendes Bild (s. Abschnitt 8.1.3).

4.9 Die Ausgabe

Die Ausgabe, die **SARA** aus einem Eingabesatz produziert, kann zwei Formen annehmen:

Die erste Möglichkeit ist die Ausgabe der Definitions- und Kommentartexte aus der Wissensbasis. Falls diese Funktion seitens der Benutzerschnittstelle eingeschaltet ist, sucht der Zerteiler zu jeder erfolgreichen Instanziierung, die bei Knoten 0 beginnt den Text aus, der in der Konzepthierarchie als Definition bei dem Konzept angegeben war, das den Kopf der Instanziierung bildet, sowie den Kommentartext der zugehörigen Fallschablone (falls es eine solche gibt). Beide Texte können leer sein. Der Zerteiler achtet darauf, auch bei mehrdeutigen Ergebnissen jeden Text nur einmal auszugeben.

Die zweite Möglichkeit ist die Ausgabe der Instanziierungen selbst. Diese Ausgabe ist normalerweise zur Weiterverarbeitung mit einem Programm zur semantischen Analyse vorgesehen. Die Instanziierungen enthalten einige Informationen, die man wohl meist nicht nutzen wird, aber deren Angabe aufgrund des geringen Ausgabevolumens auch nicht schadet. Zur leichteren Lesbarkeit für Menschen werden die eingeschachtelten Teile

(mit Ausnahme der wortinfos) entsprechend ihrer Verschachtelungsstufe eingerückt. Der Liste der Instanziierungen geht eine Zeile voran, in der der Eingabesatz selbst steht, eingeschlossen in einfache Anführungszeichen und gefolgt von einem Pfeil aus zwei Minuszeichen und einem Größerzeichen.

Solche Ausgaben sehen zum Beispiel so aus:

Das letzte Beispiel sei hier verbal erläutert (man beachte die Verschachtelung der Klammern):

Es handelt sich um eine Instanziierung (erkennbar am :I) mit zwei gefüllten Rollen (:rolle in Zeile 2 und 5). Diese Instanziierung hat eine Länge von 4 Wörtern (L4) und eine Gütebewertung von 215 Punkten (G215). Ihr Kopfkonzep ist **sitzen** und es trat auf in der Form **sitze**. Die Instanziierung hat die Form eines Deklarativsatzes im Präsens, Numerus Singular, erste Person. Es blieben zwei Rollen der Fallschablone ungefüllt (und zwar **grund** und **art_und_weise**, was allerdings nur aus der Wissensbasis abgelesen werden kann.)

Die erste gefüllte Rolle (mit Länge 1 und Gütewert 2) ist die Rolle des Agenten. Sie wird gefüllt von einer nominalen Instanziierung der Länge 1 und Güte 2 zu dem Quasi-Konzept **PRONOMEN**, das in der Form **Ich** auftrat. Das Wortinfo zu dieser Wortform besagt: Geschlecht männlich, Fall Nominativ, Numerus Singular, erste Person, Wortart Personalpronomen, Wortname personalpronomen. Es blieben in dieser Instanziierung 0 Rollen ungefüllt (es gibt nämlich gar keine Fallschablone zu Pronomen).

Die zweite gefüllte Rolle (mit Länge 2 und Gütewert 3) ist die Rolle Ort. Sie wird gefüllt von einer nominalen Instanziierung der Länge 2 und Güte 3 zu dem Konzept **haus**, das in der Form **Haus** auftrat, eingeleitet von der Präposition **im**. Das Wortinfo zur Wortform **Haus** besagt: Geschlecht sächlich, Fall Nominativ oder Dativ oder Akkusativ, Numerus Singular, dritte Person, Wortart Substantiv, Wortname Haus. Es blieb in dieser Instanziierung eine Rolle ungefüllt (und zwar, wie man der Wissensbasis entnehmen kann, **ort**).

Die Form dieser Instanziierungen wird durch die folgende Grammatik beschrieben. (Einige Teile werden in verbaler Form nachgereicht.)

instanziierungen →

instanziierung instanziierungen

| ε

instanziierung →

'[:I laenge_und_guete kopfkonzepname ' wortform ']'
globalinfo offene_rollen rollen

globalinfo →

verbalinfo

| nominalinfo

verbalinfo →

satzart konjunktion modalverb praefix '(' zeitenliste ')' ','
numerusangabe personenangabe

satzart →

:deklarativ

```

    | :Satz5
    | :imperativ
    | :W-Frage
    | :J/N-Frage
    | :Nebensatz
    | :Relativsatz
konjunktion →
    | :Konjunktion ' wort '
    | ε
modalverb →
    | :modal ' wort '
    | ε
praefix →
    | :verbpraefix ' wort '
    | ε
nominalinfo →
    | praeposition vorspannworte kernsubstantiv
praeposition →
    | ' wort '
    | ε
vorspannworte →
    | vorspannworte wortinfo
    | ε
kernsubstantiv →
    | wortinfo
offene_rollen →
    | '(' '+' zahl_ungefuellter_rollen 'R' ')'
rollen →
    | rollen '[' rolle ']'
    | ε
rolle →
    | :rolle laenge_und_guete rollenname fueller
    | :w-frage laenge_und_guete rollenname fragewortliste
    | :W-frage laenge_und_guete rollenname fueller fragewortliste
    | :unbekannt laenge_und_guete wort
rollenname →
    | wort
fueller →
    | instanziierung
fragewortliste →
    | fragewortliste '[' wortform ']'
    | ε
wortform →
    | :wortform laenge_und_guete wortinfo
wortinfo →

```

⁵unbekannte Satzart

'[' geschlechtsangabe fallangabe komparationsangabe numerusangabe
 personenangabe zeitenliste wortart wort]'

laenge_und_guete →

'L' zahl 'G' zahl

rollenname →

wort

wortform →

wort

zahl_ungefuellter_rollen →

zahl

fallangabe →

z.B. Fn für 'Nominativ', Fgda für 'Genitiv, Dativ oder Akkusativ' oder ε

geschlechtsangabe →

z.B. Gms für 'männlich oder sächlich', Gw für 'weiblich' oder ε

komparationsangabe →

z.B. Kp für 'Positiv', Kk für 'Komparativ', Ks für 'Superlativ' oder ε

numerusangabe →

Ns für 'Singular', Np für 'Plural', Nsp für 'Singular oder Plural' oder ε

personenangabe →

z.B. P1 für 'erste Person', P13 für 'erste oder dritte Person' oder ε

wortart →

verb, hilfsverb, modalverb, substantiv, adjektiv, artikel, personalpronomen,
 reflexivpronomen, demonstrativpronomen, possessivpronomen, relativpronomen,
 interrogativpronomen, praeposition, konjunktion, lokaladverb, modaladverb,
 temporaladverb, abtrennbares_praefix, fragewort, zahl.

zeitenliste →

ε oder eine Teilmenge von

(praesens, praeteritum, konjunktivI, konjunktivII, imperativ, partizipI, partizipII,
 infinitiv, infinitiv_mit_zu) wobei die einzelnen Elemente durch Kommas getrennt sind.

Kapitel 5

Implementation

*Auch auf längere Sicht
wird der Teersumpf der Software-Entwicklung
eine klebrige Angelegenheit bleiben.*

Frederick Brooks, 1975

Dieses Kapitel geht auf die konkrete Realisierung von **SARA** ein. Es wird überblickshaft die Aufteilung des Programms in Module angegeben und grob deren Aufbau erläutert. Für Erläuterungen zu den Implementationsdetails verweise ich auf die Quelltexte des Programms, in denen sich dazu ausreichend Kommentare befinden.

5.1 Allgemeines

SARA wurde auf SUN-3 unter SUN UNIX 4.2 Release 3.5 (ein BSD Unix) implementiert. Dabei wurde der C++ Übersetzer von AT&T in Release 1.2 verwendet, sowie LEX und YACC und ein C-Präprozessor namens `cccp` von GNU.

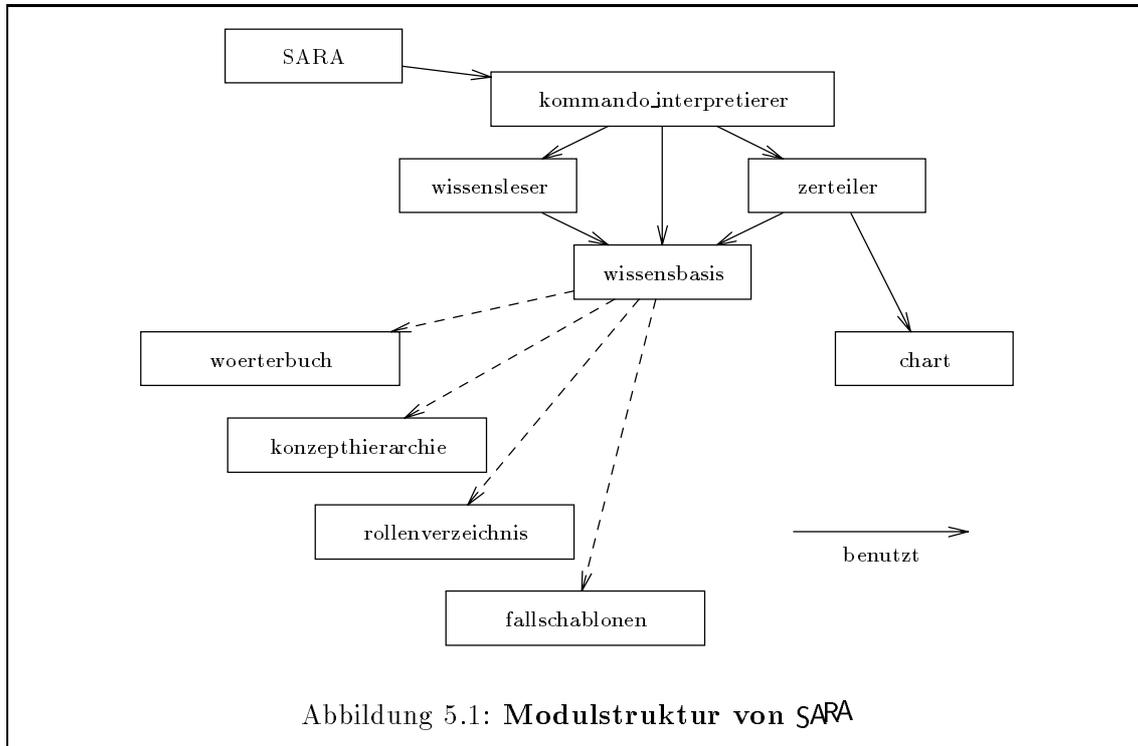
Es wurden von C++ lediglich die besseren Möglichkeiten zur Beschreibung von Datenstrukturen genutzt, die Fähigkeiten zum Überladen von Namen und Operatoren, generische Datentypen, Standardparameter und die erweiterte Typenprüfung, jedoch nicht die Möglichkeiten zum Objektorientierten Programmieren (Vererbung in einer Typhierarchie und virtuelle Funktionen). Nichtsdestoweniger stellt C++ auch in dieser Verwendungsweise einen großen Fortschritt gegenüber C dar.

Leider machte der Übersetzer erhebliche Schwierigkeiten, weil er alles andere als fehlerfrei ist. Dem schloß sich dann auch noch der normale C-Präprozessor an, so daß ich genötigt war den Präprozessor `cccp`, der im Quelltext vorlag, anzupassen. Dies ist eine potentielle Quelle von Schwierigkeiten für eine eventuelle Portierung. Ansonsten dürfte das Portieren von **SARA** keinerlei besondere Probleme aufwerfen.

5.2 Modularisierung (Top-Down)

Wie aus dem Kapitel über die Architektur hervorgeht, zerlegt sich **SARA** auf natürliche Weise in zwei Hauptteile: die Wissensbasis und den Zerteiler.

Drittens ist ein Kommando-Interpreter notwendig, der die Benutzerschnittstelle realisiert.



Die Wissensbasis ist ihrerseits in 4 Bereiche untergliedert, die einen mäßig starken Zusammenhang aufweisen: Wörterbuch, Konzepthierarchie, Rollenverzeichnis und Fallschablonenmenge. Diese Teile sind nicht als ganz eigenständige Module implementiert, da sich ihre Datenstrukturen intensiv gegenseitig benutzen, wurden in der Implementation jedoch weitestmöglich voneinander getrennt gehalten.

Die Wissensbasis wird von drei Stellen aus benutzt: von der Benutzerschnittstelle (zum Anzeigen von Daten aus der Wissensbasis), vom sogenannten Wissensleser¹ (zum Eintragen von Daten in die Wissensbasis) und natürlich vom Zerteiler (zum Abfragen der Wissensbasis). Aus diesem Grund sind die Operationen der vier Wissensbasisteile jeweils in die drei Gruppen Anzeigen, Erzeugen und Lesen unterteilt.

Der Zerteiler zerfällt in 2 Module: Die Implementation des abstrakten Datentyps chart und den eigentlichen Zerteiler, der chart verwendet.

Daraus ergibt sich eine Modulstruktur wie sie in Abbildung 5.1 angegeben ist.

Hinzu kommen noch eine Reihe von Hilfsmodulen, die in Abschnitt 5.3 beschrieben werden. Dazu zählen insbesondere die Module für Ein-/Ausgabe und die generischen Datentypen verkettete Liste und Hashtabelle.

5.2.1 Wissensbasis (gesamt)

Es gibt ein gemeinsames “Dach” der vier Wissensbasisteile: die Datenstrukturen. Diese benutzen sich gegenseitig und sind deshalb alle in einer Datei (daten.h) deklariert. Auch

¹oft auch als *Parser* bezeichnet.

die *Operationen* aller vier Teile werden zusammengefaßt; dies geschieht in der Datei `wissen.h`, die `daten.h` mit einschließt. Die beiden Dateien `daten.h` und `wissen.h` sind also allen Teilen der Wissensbasis gemeinsam und `wissen.h` stellt die Schnittstelle der Wissensbasis dar.

Diese vier Wissensbasisteile haben jeder eine nochmals unterteilte Struktur: der erste Teil enthält die Datentypen selbst und deren Grundoperationen, der zweite den eigentlichen Hauptteil (also die höheren Operationen).

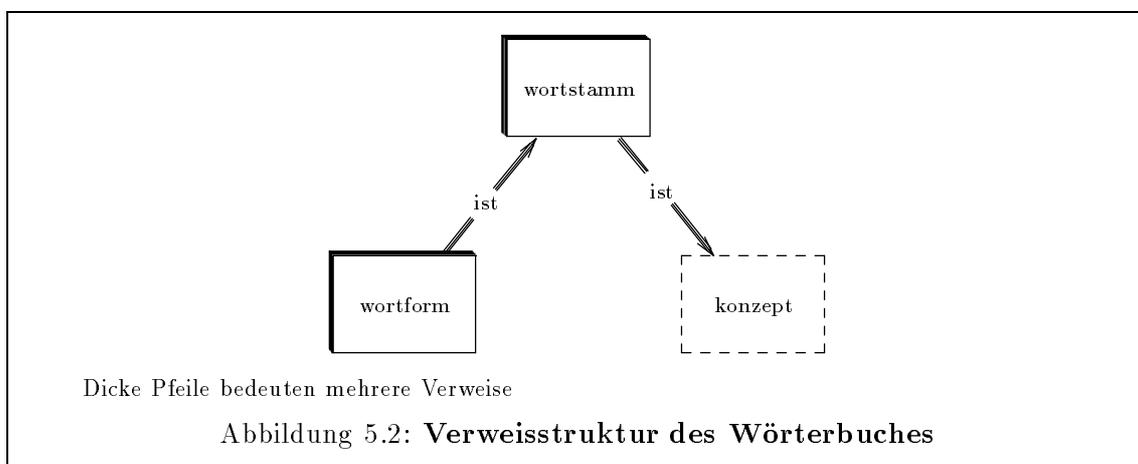
5.2.2 Wörterbuch

Das Wörterbuch ist in drei Teilen implementiert:

- `wortliste` (in den Dateien `daten.h` und `wortliste.C`), das lineare und einfach verschachtelte Listen von Zeichenketten realisiert.
- `wort` (`daten.h`, `wort.C`) das die zur Repräsentation des Wörterbuches notwendigen Datentypen enthält sowie deren Grundoperationen. `wortliste` und `wort` stellen zusammen den Grundteil des Wörterbuches dar.
- `woerterbuch` (`wissen.h`, `woerter.C`) das das Wörterbuch selbst realisiert.

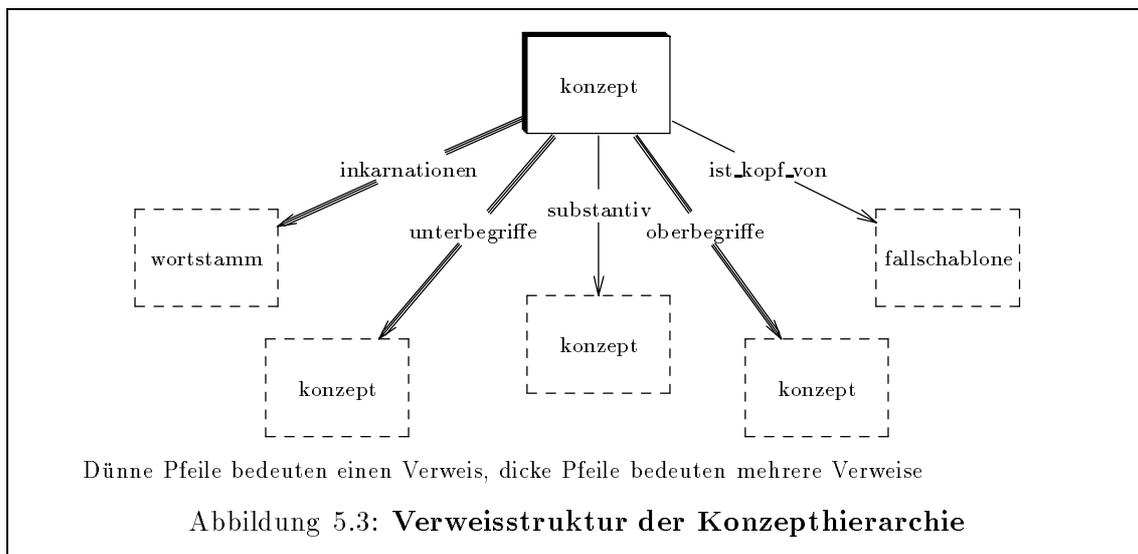
Das Wörterbuch enthält intern zwei Hashtabellen, die alle Information beherbergen. Die eine Hashtabelle enthält alle *Oberflächenformen*, also diejenigen Zeichenketten, die in der Benutzereingabe später als Wort erkannt werden können. Jeder solche Hascheintrag enthält eine Menge von Interpretationen dieser Oberflächenform. Zu einer Interpretation gehört Beugungsinformation und ein Verweis auf einen Wortstamm. Die zweite Hashtabelle enthält alle *Wortstämme* (eigentlich genauer *Wortnamen*). Diese enthalten jeweils eine Wortart und eine Menge von Verweisen auf Konzepte, die dieses Wort als eine Inkarnation enthalten.

Das Wörterbuch ermöglicht beim Zerteilen den Einstieg in die Wissensbasis: von den Wörterbucheinträgen aus werden durch Zeiger die Konzepte und Fallschablonen erreicht. Dadurch ist für jedes Eingabewort nur ein Zugriff in eine Hashtabelle notwendig.



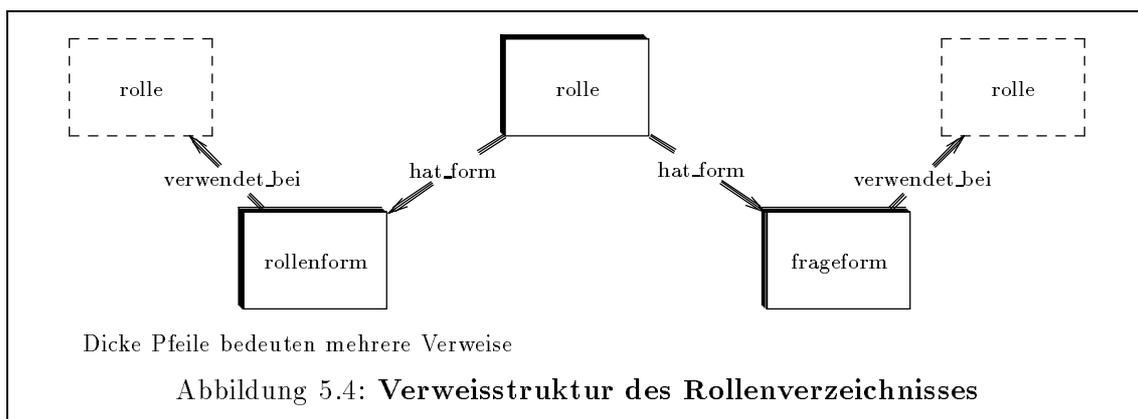
Somit ergibt sich für das Wörterbuch eine Verweisstruktur, wie sie in der Abbildung 5.2 dargestellt ist.

5.2.3 Konzepthierarchie



Die Konzepthierarchie, bestehend aus Datenstrukturen und Grundoperationen (daten.h, konzept.C) und dem Hauptteil (wissen.h, konzepte.C), ist um nur eine Hashtabelle herum aufgebaut. Diese enthält Konzepteinträge, die anhand ihres Namens und ihrer Wortart identifiziert werden. Ein Konzepteintrag enthält Verweise auf andere Konzepteinträge (für Ober- und Unterbegriffe), Verweise auf Wortstämme (für die Inkarnationen) sowie einen Verweis auf eine Fallschablone, in der dieses Konzept als Kopf auftaucht. Das ergibt für die Konzepthierarchie eine Verweisstruktur, wie sie in der Abbildung 5.3 dargestellt ist. Die transitive Hülle der Ober- und Unterkonzeptverweise wird dynamisch beim Ablaufen des Zerteilers berechnet. Es gibt keine Lesepezedur für die Konzepthierarchie, da die Einträge direkt über die Zeiger an den Wörterbucheinträgen erreicht werden.

5.2.4 Rollenverzeichnis

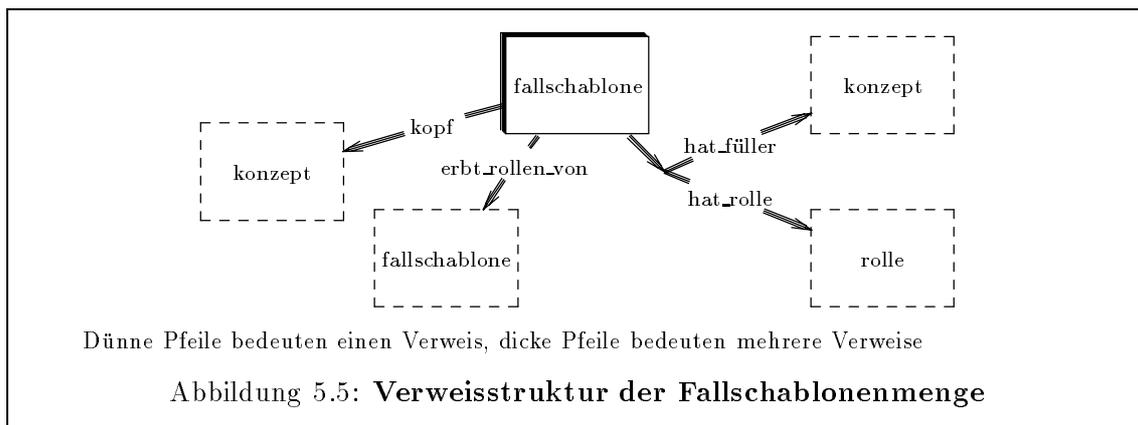


Das Rollenverzeichnis enthält drei Hashtabellen: eine für die Rollen selbst, eine für die grammatikalischen Erscheinungsformen von Rollen und eine für die Frageformen. Dies ist notwendig, um für beliebige Strategien des Zerteilers gerüstet zu sein: es muß sowohl möglich sein, dem Zerteiler solche Anfragen effizient zu beantworten, die zu einer Rolle die Angabe aller möglichen Darstellungsformen verlangen, als auch solche, die eine Darstellungsform angeben, und als Resultat eine Liste aller dazu passenden Rollen wünschen.

Beide Verfahren sollten auch bei den Frageformen möglich sein. Zwischen den Objekten der Datentypen `rolle` und `rollenform` sowie `rolle` und `frageform`, existieren also jeweils Zeiger in beiden Richtungen, so daß sich die in Abbildung 5.4 gezeigte Struktur ergibt.

Rollen werden anhand ihres Namens identifiziert, Rollenformen anhand ihres grammatikalischen Typs und ggf. ihrer einleitenden Präposition oder Konjunktion und Frageformen anhand ihres Typs und der Folge ihrer Fragewörter.

5.2.5 Fallschablonen



Die Fallschablonen werden ebenfalls in einer Hashtabelle gehalten und durch ihren Namen identifiziert (Dies ist der Name ihres Kopfkongzeptes). Sie haben Verweise auf das Konzept, das ihren Kopf bildet, auf Fallschablonen von denen ausdrücklich Rollen geerbt werden, auf die Rollen, die in ihnen direkt vorkommen, sowie auf die Konzepte, die als Füller bei diesen Rollen zugelassen sind. Soweit ergibt sich die in Abbildung 5.5 gezeigte Verweisstruktur. Dazu kommen nun noch implizite Verweise, die dadurch entstehen, daß die Konzepthierarchie verwendet wird, um implizit Rollen von denjenigen Fallschablonen zu vererben, die zu den Oberbegriffen des Kopfkongzeptes gehören, diese Verweise und ihre transitive Hülle werden dynamisch beim Ablaufen des Zerteilers berechnet.

5.2.6 Chart

Das Modul Chart realisiert zwei Datentypen, die für den Zerteiler von zentraler Bedeutung sind: `kante` und `chart`. Es wird aus Gründen der leichten Veränderlichkeit der Schnittstelle nicht ganz von der Implementation abstrahiert, so daß die Bezeichnung "Realisierung eines abstrakten Datentyps" nicht ganz auf dieses Modul zutrifft. Es wurde jedoch eine saubere Trennung zwischen den Elementaroperationen auf Charts und Kanten im Chart und den strategischen Operationen des Zerteilers erreicht. Das Modul Chart enthält nur Operationen zum Initialisieren und Löschen von Charts, zum Ausgeben von Charts, Kanten oder Teilen von Kanten, zum Erzeugen und Manipulieren von Kanten in einem Chart und zum Starten des Zerteilers.

Die eigentliche Zerteilungsfunktion wird dabei aus dem Chart-Modul heraus aufgerufen, obwohl die Benutzungshierarchie andersherum ist (der Zerteiler benutzt das Chart-Modul); die Ankopplung der Zerteilerfunktion erfolgt dabei über die Zuweisung eines Funktionszeigers.

5.2.7 Zerteiler

Der Zerteiler ist aus Gründen der einfacheren Handhabung in vier Dateien aufgetrennt. Dieses Modul verwendet ein Datenobjekt vom Typ `chart`, das in alle Funktionen, die darauf zugreifen müssen als Parameter übergeben wird. Diese Maßnahme soll zukünftige Erweiterungen erleichtern, bei denen es notwendig wird, mehr als ein Chart zu handhaben (insbesondere, um frühere Charts zur Bearbeitung von Rückwärtsreferenzen aufzuheben).

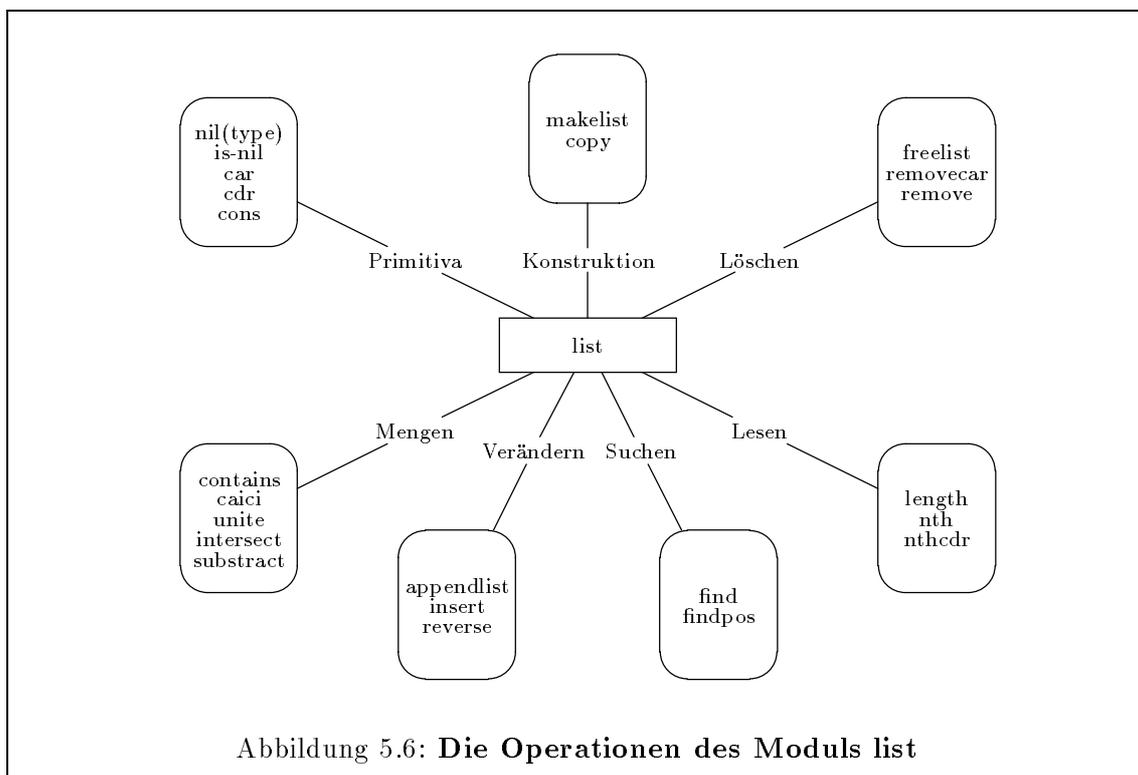
Für eine Beschreibung der internen Organisation dieses Moduls, siehe den Abschnitt 4.8 und die Programmquellen.

5.3 Sonstige Module (Bottom–Up)

einausgabe Dieses Modul ist unterteilt in “Ausgabe und Öffnen” (`ausgabe.h`, `ausgabe.C`) und “Eingabe” (`ausgabe.h`, `eingabe.h`, `eingabe.C`)

Die Ausgabe erfolgt mit Hilfe der Ausgabefunktionen der C Bibliothek. Das Modul realisiert Funktionen zur Ausgabe von Warnungen und Fehlern und zum Öffnen von Dateien mit Fehlermeldung und Ersatz bei Mißlingen. Ferner wird der in C++ gebräuchliche Ausgabeoperator `<<` für `FILE` realisiert.

Die Eingabe erfolgt mit Hilfe der Eingabefunktionen der C-Bibliothek. Das Modul realisiert einen abstrakten Datentyp `dateistapel`, der vom Wissensleser und von der Benutzerschnittstelle verwendet wird, um das verschachtelte Lesen von Eingabedateien zu verwalten. Es können in dem Stapel Dateien, Dateinamen und Zeilennummern abgelegt werden.



list implementiert den generischen und wiederverwendbaren abstrakten Datentyp einer einfach verketteten Liste mit einer Vielzahl von komfortablen Operationen. Dieses Modul bildet die Grundlage für fast alle Datenstrukturen des Zerteilers und der Wissensbasis. Die Auslegung der Operationen ist an den Listenfunktionen von LISP orientiert. Neben den Primitiva gibt es Funktionen zum Direktzugriff auf Elemente, zum Suchen und für Mengenoperationen, wie in Abbildung 5.6 dargestellt. Zur Bequemlichkeit (da es in C++ keine automatische Speicherbereinigung gibt) und da die Operationen möglichst effizient sein sollen, sind alle Funktionen, die eine Änderung bewirken, als destruktive Operationen implementiert. Das heißt sie erzeugen nicht eine Kopie des zu ändernden Operanden, sondern verändern unmittelbar dessen Struktur.

Das Modul ist sorgfältig im Quelltext dokumentiert. Es besteht lediglich aus einer Datei (`list.h`), da ein generischer Datentyp in C++ mit Hilfe der Makroexpansion implementiert wird.

hash baut auf `list` auf und implementiert den generischen und wiederverwendbaren abstrakten Datentyp einer Hashtabelle mit Kollisionsauflösung durch Verketteten. Es gibt außer dem eigentlichen Hashmodul (`hash.h`) auch noch eines, das eine Reihe von guten Hashfunktionen für bestimmte Anwendungssituationen bei der Verarbeitung von Zeichenketten realisiert (`hashfunction.h`, `hashfunction.C`). Beide sind im Quelltext ausführlich beschrieben.

Kapitel 6

Test

*Testen ist der Prozess,
ein Programm mit der Absicht auszuführen,
Fehler zu finden.
[... Dies] impliziert zum Beispiel,
daß Testen ein destruktiver,
ja geradezu ein sadistischer Prozeß ist.
Glenford Myers*

Dieses Kapitel beschreibt die Überlegungen, die dem Test von **SARA** zugrundeliegen. Leider sind im Bereich des Programmtestens viele englische Fachausdrücke nur sehr verkrampft ins Deutsche zu übertragen; es kommen deshalb in diesem Kapitel etwas häufiger Jargonbegriffe vor als sonst. Literatur zu diesem Kapitel sind [Myers79], aus dem auch der obige Spruch stammt, sowie [Fairley85].

6.1 Schwächen beim Test

Bevor ich mit der Beschreibung meiner Teststrategie beginne, möchte ich einige fundamentale Probleme nennen, mit denen die Implementation von **SARA** zu kämpfen hatte:

1. Zeitmangel. Es standen für Implementation und Test des gesamten Systems nur 9 Wochen zur Verfügung. Das ist jedoch bei weitem nicht ausreichend, um alle Module einer so genauen Kontrolle zu unterwerfen, wie es eigentlich wünschenswert wäre.
2. Personalmangel. Eine der Grundregeln des Programmtestens besagt, daß der Programmierer eines Moduls die denkbar schlechteste Person ist, um es zu testen. Gegen diese Maxime mußte ich notgedrungen ununterbrochen verstoßen, was vermutlich die Hauptschwäche im Test von **SARA** darstellt.
3. Neuartigkeit. Die ganze Konzeption des Systems war nicht durch ein vergleichbares früheres Projekt vertraut. Dadurch erschwerte sich eine geschickte Spezifikation enorm. Unregelmäßige oder ungeschickte Schnittstellen erschweren auch das Testen.

6.2 Allgemeines Vorgehen

Das Testen eines Programms oder Moduls ist ein dreistufiger Prozeß:

1. Fehlerfeststellung. Man benutzt Testfälle, die allgemein fehlerprovokant gestaltet sind, um möglichst einen Fehler hervorzurufen. Der Rest dieses Kapitels beschäftigt sich hauptsächlich mit diesem Problem. Ist bei keinem dieser Testfälle ein Fehler aufgetreten, so ist das Testen des Programms oder Moduls beendet; andernfalls fahre mit Stufe 2 fort.
2. Fehlerlokalisierung. Ist festgestellt worden, *daß* ein Fehler vorliegt, muß man nun ermitteln *wo* er sich befindet und *wie* er zustande kommt. Dazu gibt es eine Reihe von unterschiedlichen Ansätzen, die im Abschnitt 6.7 behandelt werden.
3. Fehlerbehebung. Hat man den Fehler nun genau beschrieben, so muß er beseitigt werden: es gibt entweder ein kleines fehlerhaftes Programmstück, das korrigiert werden kann (oft nur eine Anweisung oder ein Ausdruck) oder es muß ein gewisser größerer Teil des Programms oder Moduls umgestaltet werden. In beiden Fällen kann durch die Fehlerbehebung ein neuer Fehler entstehen; außerdem sind vielleicht bei Stufe 1 mehrere Fehler an den Fehlersymptomen beteiligt gewesen. Deshalb muß nun wieder bei Stufe 1 begonnen werden.

6.3 Big Bang Test \iff inkrementeller Test

Beim sogenannten *Big Bang Testen* oder auch *nichtinkrementellem Testen* werden alle Module zugleich getestet. Der Big Bang Test findet also statt, sobald das Programm fertig kodiert ist. Es werden Testfälle auf Anwendungsebene verwendet.

Beim *inkrementellen Testen* wird stets nur ein Modul getestet. Die getesteten Module werden nach und nach zum kompletten Programm zusammengebaut und zwar entweder beginnend mit den elementarsten Operationen (bottom-up) oder mit dem Hauptprogramm (top-down).

Für jedes Programm ernstzunehmender Größe ist das Big Bang Testen praktisch nicht mit Erfolg durchführbar. Es wird deshalb bei **SARA** inkrementell getestet.

6.4 Blackbox-Test \iff Whitebox-Test

Blackbox-Testen bedeutet, Testfälle anhand der Spezifikation zu entwerfen, ohne Kenntnis der inneren Struktur der Implementation.

Whitebox-Testen verwendet demgegenüber Wissen über die konkrete Implementation bei der Definition der Testfälle. Dort kann auch Wissen über die erwartete Reaktion in solchen Punkten verwendet werden, die von der Spezifikation nicht festgelegt sind (wo also Freiheitsgrade existieren), um zu prüfen, ob die Operation nicht nur korrekte Resultate liefert, sondern auch wirklich intern in der vorgesehenen Weise arbeitet. Obwohl sich das Whitebox-Testen als eine Erweiterung des Blackbox-Testens verstehen ließe, sieht man beide als disjunkt an: beim Whitebox-Test wird die Spezifikation zwar beachtet, jedoch nicht zur Definition von Testfällen genauer ausgewertet.

Ich habe im Allgemeinen ein gemischtes Verfahren verwendet, wobei vom Blackbox-Testen hauptsächlich die Einteilung der Parameter in *Äquivalenzklassen* herangezogen wird und vom Whitebox-Testen die *Entscheidungsabdeckung*, d.h. es wird gefordert, daß aus jeder Klasse von qualitativ unterschiedlichen Eingabedaten ein Beispiel getestet wird und daß jede Fallunterscheidung im Innern der Implementation mindestens einmal jedes mögliche Ergebnis gehabt hat. Insbesondere werden alle (vorgesehenen) Fehlerfälle getestet. Die Entscheidungsabdeckung ist keinesfalls mit der Pfadabdeckung zu verwechseln, bei der gefordert wird, daß jede mögliche *Kombination* von Ergebnissen bei allen Fallunterscheidungen einmal aufgetreten ist; diese Pfadabdeckung ist im allgemeinen wegen der kombinatorischen Explosion nicht einmal für einzelne Funktionen zu gewährleisten, geschweige denn für ein ganzes Programm.

6.5 Bottom-up Test \iff Top-down Test

Beim *Top-Down Testen* wird für jede Operation aus untergeordneten Modulen, die das zu testende Modul aufruft, ein sogenannter *Stummel* geschrieben, der die Leistungen dieser Operationen für die speziellen Testfälle simulieren kann. Man beginnt beim Testen mit dem Hauptmodul und fügt als nächstes jeweils eines der unmittelbar untergeordneten Module anstatt seines Stummels hinzu. Da beim Testen Ein-/Ausgabeoperationen von vitaler Bedeutung sind und diese zugleich meist sehr weit unten in der Modulhierarchie liegen, wird Top-Down Testen meist nicht in Reinkultur durchgeführt.

Bottom-Up Testen bedeutet mit den Modulen zu beginnen, die keine anderen mehr aufrufen. Man schreibt dafür einen Treiber, der seinerseits die Operationen des zu testenden Moduls aufruft. Es wird immer ein Modul hinzugefügt, zu dem alle untergeordneten Module bereits getestet sind.

Ich habe bei **SARA** beide Verfahren verwendet. So ist es beispielsweise sinnvoll, für die komplexen Operationen auf den höheren Ebenen den normalen Kommandointerpretierer, der die spätere Benutzerschnittstelle bildet, als Treiber zu verwenden, so daß dieser Top-Down getestet wird, während andererseits sich für die Bottom-Up Teile ein Bottom-Up Test schon aus Gründen des Aufwands aufdrängt und zudem nötig ist, um eine ausreichende Abdeckung zu gewährleisten.

6.6 Codeinspektionen und Walkthroughs

Eine *Codeinspektion* ist eine Veranstaltung mit etwa 3 oder 4 Personen, darunter der Programmierer und der Entwerfer des Programms, in der der Programmierer den übrigen Personen die Logik des zu untersuchenden Moduls oder Programms erklärt. Dabei ergeben sich Rückfragen dieser anderen Mitglieder, deren Diskussion vielfach zur Entdeckung von Fehlern führt; häufig werden auch spontan vom Programmierer beim Sprechen Fehler entdeckt.

Ein Walkthrough funktioniert ganz ähnlich, jedoch wird nicht irgendwie die Logik des Programms erklärt, sondern man spielt gemeinsam Computer, d.h. es werden einige einfache Testfälle Schritt für Schritt durchgespielt.

Der Nutzen beider Verfahren kommt vor allem aus der Kommunikation: durch den Austausch über die Schwierigkeiten beim Erklären, Verstehen oder Durchführen des

Programmablaufs ergeben sich Einsichten in die Fehleranfälligkeit von Programmteilen, mögliche Fehlverständnisse in den Schnittstellen und einige bisher unbeachtet gebliebene andere Eigenschaften des Programms. Leider ist dieser Nutzen weitgehend dahin, wenn nur eine Person zur Durchführung zur Verfügung steht. Dennoch zeigt die Praxis, daß es auch oft Erfolg hat, bei einer Ein-Mann-Codeinspektion die Programmlogik dem Tischbein oder einem Druckbleistift zu erklären.

Ich habe für manche Module eine Codeinspektion zu zweit zusammen mit meinem Betreuer Rolf Adams durchgeführt; manche der übrigen Module wurden einer Ein-Mann-Codeinspektion unterzogen. Ferner wurde eine Codeinspektion unterhalb der Modulebene für eine einzelne Prozedur durchgeführt, sobald ein Fehler einer Prozedur zugeordnet wurde — auch, wenn die Fehlerstelle offensichtlich zu sein schien. Zur Fehlerlokalisierung habe ich wo es gewinnbringend erschien auch Walkthroughs verwendet, die ich mit Unterstützung durch einen Debugger durchgeführt habe. Ansonsten wurden Walkthroughs nicht benutzt.

6.7 Fehlerlokalisierung

Zur genauen Beschreibung von Ort und Art eines entdeckten Fehlers gibt es eine Reihe von unterschiedlichen Vorgehensweisen:

Bei der *Hau-Ruck Methode* werden unstrukturiert Debugger und Ablaufverfolger eingesetzt, um “irgendwie” hinter den Fehler zu kommen, oder gar aufs geratewohl Änderungen am Programm gemacht, nach dem Motto “Mal sehen, was passiert, wenn ich hier statt einer 0 eine 1 hinschreibe”.

Die *Teekesselchen-Methode (induktives Vorgehen)* versucht durch geeignetes Zusammentragen der Beobachtungen aus den Testfällen allmählich eine Beschreibung von Fehlerort und Fehlerursache zu erhalten.

Bei der *Ausschlussmethode (deduktives Vorgehen)* trägt man die Beobachtungen anders zusammen und versucht von einer sehr allgemeinen Fehlerannahme ausgehend, den Fehler durch Ausschliessen von Ursachen und Orten einzukreisen.

Beim *Lokalisieren durch Testen* werden speziell gestaltete Testfälle verwendet, um einzelne Vermutungen über Ort oder Ursache des Fehlers zu überprüfen.

Das *Lokalisieren durch Ablaufverfolgung* kann erst auf schon recht eng eingekreiste Fehler angewendet werden. Es entspricht einem Walkthrough, wobei sich aber die Aufmerksamkeit auf wenige Objekte des untersuchten Programms konzentriert. Hierbei ist die Unterstützung eines symbolischen Debuggers hilfreich.

Beim *Lokalisieren durch Rückverfolgung* versucht man im Geiste das Programm ab der Stelle, an der man den Fehler festgestellt hat, rückwärts durchzuführen, bis die fehlerproduzierende Stelle erreicht ist. Dies ist jedoch nur bei sehr einfachen Programmstücken realisierbar.

Ich habe zur Fehlerlokalisierung in **SARA** meist mehrere dieser Verfahren verwendet. Dagegen ist nichts zu sagen, solange sie nicht unstrukturiert vermischt werden: Meist bietet es die besten Aussichten zunächst mit Lokalisieren durch Testen zu beginnen, wobei die Testfälle aus Erkenntnissen konstruiert werden, die man mit dem induktiven und dem deduktiven Vorgehen erlangt hat. Ist dann der Fehler genügend eng eingekreist, so kann man ihn mittels Ablaufverfolgung meist recht schnell exakt auffinden. Die Rückverfolgung dient zur Behandlung einfacher Fälle und Hau-Ruck Methoden stellen das letzte Mittel

bei totaler Ratlosigkeit dar, die zum Beispiel sehr häufig dann auftritt, wenn der Fehler nicht beim Programmierer liegt, sondern irgendwo vom Entwicklungssystem verursacht wird, was mir mehrfach begegnete.

6.8 Automatisches Rückfalltesten

Ein Hauptproblem beim Testen größerer Softwaresysteme ist die Empfindlichkeit gegen Änderungen: jede Korrektur an einer Stelle eines Moduls kann verändertes Verhalten auch an anderen Stellen nach sich ziehen. Je nach Einzelfall kann dieses unerheblich sein oder aber zu Fehlern führen. Meist wird jedoch in der Praxis nach einer Veränderung nur die kleinstmögliche von der Veränderung betroffene Einheit getestet und zwar mit der kleinstmöglichen ausreichend erscheinenden Zahl von Testfällen. Dies gilt sogar dann, wenn nur ein bereits vorhandener Satz von Testfällen ausgeführt werden muß; in diesem Fall scheitert das korrekte Testen daran, daß die anschließende Auswertung der Ausgaben nicht vollständig durchgeführt wird.

Diese Neigung führt dazu, daß oftmals Änderungen (auch Fehlerkorrekturen!) die Qualität eines Programms nicht verbessern, sondern verschlechtern. Dieser Effekt kann nur vermieden werden, indem man durch geeignete organisatorische Maßnahmen sicherstellt, daß alle für das geänderte Modul durchgeführten Tests nach jeder Änderung automatisch wiederholt werden — und zwar *einschließlich* der Kontrolle der Ausgaben.

Deshalb werden bei **SARA** die meisten Tests in der Form des *automatischen Rückfalltestens* durchgeführt. Dabei gibt es für jedes zu testende Modul einen Treiber, der ohne weitere Eingaben des Benutzers die vorgesehenen Testfälle ausführt und die Ergebnisse teilweise mit Hilfe von *Zusicherungen (assertions)* direkt überprüft und teilweise Ausgaben in eine Datei schreibt. Ferner gibt es eine Datei, welche die Soll-Ergebnisse enthält. Man kann dann mit Hilfe des von **make** automatisch bei der Neuübersetzung eines Moduls auch die zugehörigen Tests ablaufen lassen und mit **diff** die Ausgaben mit den Sollwerten vergleichen. Dies stellt sicher, daß jeder durch die Änderung bewirkte Fehler, der von den bisherigen Testfällen gefunden werden kann, auch tatsächlich gefunden wird. Man muß nicht mehr alle Ausgaben kontrollieren, sondern nur noch feststellen, ob irgendwelche Abweichungen angezeigt worden sind. Durch dieses Verfahren wird eine sehr häufige Quelle von Fehlern im endgültigen Programm ausgeschaltet.

6.9 Konkretes Vorgehen

Zu jedem zu testenden Modul **X** gibt es einen Testtreiber, bestehend aus den Dateien **Xtest.C**, **Xtest** (Quelltext und ausführbares Programm) und ggf. **Xtest.in** (Eingabedaten), mit dem zunächst im Handbetrieb das Modul so lange getestet und korrigiert wird, bis alle Resultate korrekt sind. Dieser Treiber stellt den nach den obigen Prinzipien entwickelten Test für das Modul dar.

Die korrekten Resultate werden dann in die Datei **Xtest.ok** geschrieben und es wird ein Aufruf **test Xtest** in das Makefile aufgenommen, der ab dann bei jeder Änderung von **X** im Rahmen des Wiederherstellens von **Xtest** automatisch den Test durchführt, die Ergebnisse in die Datei **Xtest.out** schreibt, diese mit **Xtest.ok** vergleicht und die Resultate des Vergleichs dem Benutzer zur Kenntnis bringt.

Diese Konstruktion stellt sicher, daß Rückfalltests auch tatsächlich durchgeführt werden.

Kapitel 7

Bedienung

An ENTER without an entered entry enters the end entry and ends the entry enter mode. So never enter ENTER without having entered an entry if you don't want to end the entering by entering the end entry and ending the entry enter mode.

7.1 Aufruf und Kommandoeingabe

SARA ist ein interaktives Programm. Es nimmt Kommandos und zu analysierende Eingabesätze normalerweise von der Tastatur entgegen und schreibt seine Ausgaben normalerweise auf den Bildschirm. Beides läßt sich jedoch umlenken.

Der Aufruf hat die Form

```
sara
oder
sara eingabe
```

Bei der zweiten Form wird die Eingabe auf die angegebene Datei umgelenkt und die dort stehenden Zeilen werden als Kommandos interpretiert und abgearbeitet. Nehmen wir an, **SARA** sei ohne Parameter aufgerufen worden.

Nach dem Aufruf meldet sich **SARA** mit **SARA:** und wartet auf Eingabe. Gibt man nun **?** ein und Return, so erfolgt die Kurzbeschreibung der verfügbaren Kommandos mit der Ausgabe

```
Erlaubte Kommandos sind:
? - Zeige diese Hilfeseite an
q - Quit. Beende die Sitzung
s - Rufe eine Shell auf
l - Lese Wissensbasis
w - Zeige Woerterbuch
k - Zeige Konzeptionhierarchie
r - Zeige Rollenverzeichnis
```

```

f - Zeige Fallschablonen
t - Teste die Wissensbasis auf Konsistenz
h - Hilfesystem: Zeige Kommentarfeld als Antwort (ein/aus)
i - Instanziierungen: Zeige volle Ausgabeform (ein/aus)
d - Frage nach Ausgabedateinamen bei l,w,k,r,f,t,h,i (ein/aus)
a - Standardausgabe umlenken
e - Standardeingabe umlenken (Kommandodatei)
Laengere Eingaben als ein Zeichen werden als Spracheingabe analysiert.

```

Ein Kommando besteht also immer nur aus einem einzigen Buchstaben; eventuell nötige Parameter werden im Dialog erfragt. Dabei gibt es meist einen Standardwert für jeden Parameter. Dieser wird benutzt, wenn bei der Anfrage eine leere Eingabe gemacht wird.

Als Kommando wird genau jede Eingabe aufgefaßt, die nur ein Zeichen lang ist, alle längeren Eingaben werden als Eingabesatz verstanden und dem Zerteiler zugeführt.

7.2 Beschreibung der Kommandos

In alphabetischer Reihenfolge:

a Standardausgabe umlenken. **SARA** fragt nach dem Namen der Datei in die die Standardausgabe umgelenkt werden soll; Standardwert ist stdout, also der Bildschirm. Falls diese Datei schon existiert, muß ein Schreibrecht darauf bestehen, falls sie nicht existiert, muß es möglich sein, sie zu eröffnen. Ist beides nicht der Fall, so wird statt dessen stdout verwendet.

Die mit diesem Kommando definierte Standardausgabe wird normalerweise bei allen Ausgaben außer dem Kommandodialog selbst verwendet, sie kann jedoch bei den einzelnen Kommandos nochmals temporär geändert werden (siehe **d**). An diese Datei wird immer angehängt, nicht überschrieben.

d Frage nach Ausgabedateinamen ein/aus. Bei den Kommandos **f**, **h**, **i**, **k**, **l**, **r**, **t** und **w** wird normalerweise nach einer Datei gefragt, in die die Ausgaben des Kommandos, anstatt zur Standardausgabe (die ihrerseits umgelenkt sein kann), gehen sollen. Diese Frage kann mit **d** ab- und auch wieder angeschaltet werden. Bei dieser Zweitumlenkung wird im Gegensatz zur einfachen Umlenkung die betroffene Datei jedesmal neu begonnen, also nicht angehängt.

e Standardeingabe umlenken. **SARA** erfragt den Namen einer Datei (Standardwert stdin, also die Tastatur), aus der ab sofort alle Kommandoeingaben entnommen werden. Dies wird normalerweise beibehalten, bis die Datei vollständig gelesen ist. Taucht in der Datei das Kommando **q** auf, wird **SARA** ordnungsgemäß verlassen. Bei **e** wird die dort angegebene Datei als Eingabe zwischengeschoben und nach ihrem Ende an der augenblicklichen Position weitergelesen; die Eingabeumlenkung kann also verschachtelt werden.

f Zeige Fallschablonen. **SARA** erfragt den Namen einer Fallschablone und eventuell auch einen Ausgabedateinamen (siehe **d**). Die angegebene Fallschablone wird gesucht und falls möglich ausgegeben. Eine leere Eingabe für den Namen führt zur Ausgabe aller Fallschablonen, die sich in der Wissensbasis befinden.

h Hilfesystem.¹ Zeige als Ausgabe nach der Analyse eines Eingabesatzes die Kom-

¹Natürlich ist diese Bezeichnung blöde. Aber es gibt nun mal nur 26 lateinische Buchstaben.

mentarfelder der instanziierten Fallschablonen und die Definitionen der zugehörigen Kopfkongzepte. Dieser Modus wird hiermit ein- und auch wieder ausgeschaltet.

i Instanzierungen. Zeige als Ausgabe nach der Analyse eines Eingabesatzes die Instanzierungen der erkannten Fallschablonen. Dieser Modus wird hiermit aus- und auch wieder eingeschaltet.

k Zeige Konzepthierarchie. Analog zu **f**.

l Lese Wissensbasis. Es wird der Reihe nach erfragt: (1) Der Name der Datei, die die Wissensbasis enthält (kein Standardwert; eine leere Eingabe führt zum Beenden des Kommandos) und (2) eventuell (siehe **d**) der Name einer Ausgabedatei für die Fehlermeldungen und Warnungen. Anschließend wird die betreffende Datei als Wissensbasis eingelesen. Dabei werden bei syntaktisch unzulässigen oder unsinnigen Eingaben entsprechende Meldungen ausgegeben. Es kann als Dateiname der Wissensbasis ein Minuszeichen eingegeben werden. Dies bewirkt, daß die Daten von der Standardeingabe gelesen werden, bis dort das "Dateiende" angezeigt wird. (Also im Normalfalle Lesen von der Tastatur, bis man Ctrl-D tippt.)

r Zeige Rollenverzeichnis. Analog zu **f**.

s Shell. Es wird eine C-Shell aufgerufen. Diese kann mit `exit` wieder verlassen werden, um nach `SARA` zurückzukehren.

t Test der Wissensbasis auf Konsistenz. Hiermit wird eine Kontrolle angestossen, ob für alle Objekte, auf die es in der Wissensbasis einen Verweis gibt, auch Einträge existieren und diese Einträge sinnvoll sind. Alle Objekte, für die das nicht der Fall ist, werden angezeigt.

Im Wörterbuch werden Einträge ohne Oberflächenformen entdeckt, die entstehen, wenn ein Konzept ein Wort als Synonym aufführt, ohne daß dieses Wort definiert ist.

Im Konzeptverzeichnis werden mehrfach aufgeführte Synonyme und Oberbegriffe gemeldet, sowie Oberbegriffspfade, die Kreise ergeben. Ferner fehlende Wortart oder Wortartwechsel zwischen Konzept und Synonym oder Oberkonzept.

Für das Rollenverzeichnis werden solche Rollen aufgelistet, die keine Rollenformen haben und solche, die Rollenformen oder Frageformen mehrfach besitzen.

Bei den Fallschablonen schließlich werden Fallschablonen gemeldet, die leer sind (das heißt sie wurden einmal per Vorausverweis benannt, danach jedoch nicht definiert.), die explizit zum Vererben benutzt werden, obwohl sie einen Kopf haben, die mehrmals von derselben Fallschablone erben oder die beim Vererben einen Kreis bilden.

Die meisten dieser Fehler sind nur Hinweise auf harmlose Ungereimtheiten in der Wissensbasis, die dazu führen können, daß der Zerteiler bestimmte Eingaben nicht verarbeiten kann. Die Fehler, die in der Konzepthierarchie oder bei den Fallschablonen jedoch kreisförmige Relationen bewirken, führen dazu, daß der Zerteiler in eine Endlosschleife geraten kann, wenn er diese Kreise bei seiner Analyse berührt.

q Quit. `SARA` wird beendet.

w Zeige Wörterbuch. Analog zu **f**. Im Wörterbuch haben allerdings gelegentlich mehrere Einträge denselben Namen; diese werden dann alle angezeigt.

7.3 Ausgabe des Chart

Wenn sowohl die Anzeige der Kommentarfelder als auch die Anzeige der Instanzierungen ausgeschaltet sind, so wird stattdessen das Chart selbst in seiner internen Form ausgegeben. Diese Ausgabe kann zum genaueren Verständnis der Reaktionen (oder Nichtreaktionen) des Zerteilers herangezogen werden.

Diese Ausgabe kann bei eingeschalteter Kommentar- und/oder Instanzierungsausgabe auch zusätzlich angefordert werden, indem man als erstes Zeichen des Eingabesatzes ein Doppelkreuz # eingibt. Dieses wird vom Zerteiler bei der Analyse ignoriert.

Treffen beide Bedingungen aufeinander, d.h. weder Kommentar- noch Instanzierungsausgabe ist eingeschaltet *und* ein Doppelkreuz erscheint am Beginn der Eingabe, dann wird die Ausgabe des Chart wieder unterdrückt. Es erfolgen dann also überhaupt keine Ausgaben, was für Leistungsmessungen des Zerteilers nützlich sein kann.

7.4 Ausgaben im EDGE-Format

EDGE [Newbery88] ist ein Grapheneditor, der insbesondere das automatische Layout von beliebigen Graphen beherrscht. Da bei SARA Datenmengen von beträchtlicher Größe verwendet werden, die die Struktur eines Graphen haben, wurde eine Möglichkeit vorgesehen, diese Datenmengen in einem für EDGE lesbaren Format auszugeben, um eine leichtere Bearbeitung zu ermöglichen.

In allen Fällen wird die Anweisung `include: sarainclude.grl` mit in die Ausgabe aufgenommen.

7.4.1 Konzepthierarchie

Die Konzepthierarchie ist ein zyklensfreier gerichteter Graph, dessen Knoten die Konzepte sind. Die Kanten haben die Bedeutung "ist Oberkonzept von". Die Konzepthierarchie wird immer dann im EDGE-Format ausgegeben, wenn beim Kommando `[k]` kein Name angegeben wird (also alle Konzepte ausgegeben werden sollen). Die EDGE-Ausgabe erfolgt ohne weitere Bedienung oder Meldung und landet immer in der Datei `sara_konzepte.grl`

Die Knoten tragen als Titel den Konzeptnamen, die Kanten bleiben unbenannt. Jeder Knoten hat als `typename:` die Wortart seines Konzepts. Für diese Typnamen kann in `sarainclude.grl` eine Layoutvorschrift angegeben werden.

7.4.2 Chart

Das Chart ist ein gerichteter Graph mit genau einer Quelle (Eingabeanfang) und einer Senke (Eingabeende), dessen übrige Knoten die Wortzwischenpositionen sind. Die Kanten tragen vielfältige Information unterschiedlicher Bedeutung, z.B. Wort, Rolle, Instanzierung. Das Chart wird immer dann im EDGE-Format ausgegeben, wenn es auch in der internen Form ausgegeben wird (s. Abschnitt 7.3). Diese EDGE-Ausgabe landet immer in der Datei `sara_chart.grl`

Die Knoten tragen als Titel ihre Nummer, die Kanten eine je nach Kantentyp unterschiedliche Kurzbeschreibung ihres Inhalts. Jede Kante hat als `typename`: den Kantentyp. Für diese Typnamen kann in `sarainclude.gr1` eine Layoutvorschrift angegeben werden.

Leider läßt sich die Information einer Kante nicht gut in sehr kurzer Form zusammenfassen. Insbesondere wird deshalb nicht dargestellt, welche Unterkanten einer längeren Kante zugehören. Für Instanzierungskanten kann auch der eigene Inhalt bei weitem nicht komplett im verfügbaren Raum dargestellt werden. Diese Einschränkungen machen die Graphendarstellung des Chart etwas unbefriedigend. Sie ist jedoch dennoch eine nützliche Orientierungshilfe, zu der die schriftliche Form der Chartausgabe als Ergänzung herangezogen werden kann.

Kapitel 8

Auswertung und Ausblick

*Was wir zu tun lernen müssen
lernen wir, indem wir es tun.*

Aristoteles

Bei Abschluß dieser Arbeit war bereits eine Studienarbeit in vollem Gange, die **SARA** verwendet und zum Ziel hat, eine Wissensbasis zu erstellen für ein Hilfesystem für **L^AT_EX**. Deshalb konnten bereits einige Erfahrungen mit **SARA** gesammelt werden. Weitere Experimente wurden mit der im Anhang abgedruckten Beispiel-Wissensbasis gemacht.

8.1 Beobachtungen

Selbst in der kurzen Zeit, die **SARA** bislang erst verwendet wird, ließen sich bereits eine Reihe von Punkten ausmachen, die als auffallend positiv oder negativ zu bewerten sind:

8.1.1 Stärken

1. Die für das Wörterbuch verwendete Form der Eintragungen ist eingängig und kann nach kurzer Eingewöhnungszeit mit einer sehr geringen Fehlerquote verwendet werden. Dies gilt vor allem für den kritischen Punkt der Beschreibung von Substantiven.
2. Die Konzepthierarchie ist ein gutes Beschreibungsmittel für semantische Kategorien, in dem Sinne, daß es die Größe der Fallschablonenmenge in erträglichen Grenzen hält.
3. Die Möglichkeit, das Rollenverzeichnis selbst zu bestimmen ermöglicht eine der Anwendung angepasste Bezeichnung der Rollen und manchmal eine Beseitigung von Mehrdeutigkeiten durch Weglassen bestimmter Rollenformen, was der Qualität der Ausgaben des Zerteilers sehr zuträglich ist.
4. Die Vererbung von Rollen in der Fallschablonenmenge entlang der Unterbegriffsverweise scheint ebenfalls eine adäquate Methode zu sein. Sie haelt die Größe der einzelnen Fallschablonen gering und erleichtert die Änderung von Fallschablonengruppen.
5. Die Geschwindigkeit des eigentlichen Zerteilers ist hoch (s. Abschnitt 8.1.3).

8.1.2 Schwächen

1. Die unkontrollierte Zuordnung von beliebigen Verbpräfixen zu beliebigen Verben führt manchmal zu unsinnigen zusätzlichen Ausgaben bei der Analyse. Dies gilt insbesondere, sobald das Präfix *ein-* definiert ist, das mit dem unbestimmten Artikel kollidiert.
2. Die Wortartbindung der Konzepte ist eine unnötige Einschränkung. Dadurch lassen sich Rollen, die wahlweise durch verschiedene Klassen grammatikalischer Konstruktionen beschrieben werden können, nur mittels mehrerer Konzepte in Fallschablonen einbauen, auch wenn die Konzepte sich semantisch gar nicht unterscheiden. Man bräuchte stattdessen vollpolymorphe Konzepte, d.h. solche mit beliebig unterschiedlichen Erscheinungsformen.
3. Die rein manuelle Handhabung unregelmäßiger Verben ist doch etwas umständlich.
4. Das Einlesen einer Wissensbasis dauert recht lange¹.
5. Die Fehlermeldungen des Parsers auf syntaktische Fehler beim Einlesen der Wissensbasis sind dürftig.

8.1.3 Messungen

Leider erwies sich die Verwendung des Profiling-Utilities `gprof` als nicht ohne Weiteres durchführbar, weil die dabei verwendete Bibliotheksroutine `_exit` mit einer gleichnamigen von C++ kollidiert. Deshalb kann hier leider keine detaillierte Auswertung des Zeitverhaltens erfolgen, die auch feststellt, *wo* im Innern des Zerteilers die meiste Zeit verbraucht wird.

Es werden deshalb nur einige simple Messungen anhand der Beispiel-Wissensbasis angegeben. Diese erfolgten auf einem Rechner SUN 3/60M-8. Alle Messungen wurden mehrfach unmittelbar hintereinander ausgeführt, so daß die angegebenen Zeiten im Wesentlichen keine Plattenzugriffe einschließen. Der Rechner war ansonsten unbelastet, die Zeitangaben sind Echtzeit. Man beachte, daß die Beispiel-Wissensbasis die Standard-Wissensbasis mit einschliesst.

Die ausführbare Datei hat eine Größe von 237 kB.

Das Laden und wieder verlassen des Programms dauert 0.7s.

Es benoetigt im leeren Zustand (laut `ps`) 416 kB Speicher; dieser Wert erhöht sich durch das Laden der Beispiel-Wissensbasis um 136 kb.

Dieses Laden dauert beim ersten Mal 17s, beim zweiten Mal (d.h. wenn alle Einträge überschrieben werden) 15s.

Der Konsistenztest dauert dann 1s (einschliesslich Ausgabe der Ergebnisse auf den Bildschirm).

Der Zerteiler selbst wurde mit mehreren Beispielen gemessen. Bei der Beurteilung dieser Ergebnisse ist zu beachten, daß für die Dauer der Zerteilung neben der Länge der Eingabe nicht an sich die Größe der Wissensbasis ausschlaggebend ist, weil durch die Art der Implementierung mit Hashtabellen jeder Zugriff auf die Wissensbasis unabhängig von ihrer

¹ Allerdings kann bei Änderungen inkrementell nachgebessert werden, d.h. es muß nur selten die ganze Wissensbasis eingelesen werden.

Größe in fast konstanter Zeit abgewickelt werden kann.

Entscheidend ist vielmehr die Stärke der Mehrdeutigkeit im Eingabesatz gemessen an der Struktur der Wissensbasis, weil diese Mehrdeutigkeit dramatisch die Zahl von Operationen beeinflusst, die der Zerteiler durchführen muss.

In der Beispiel-Wissensbasis sind die semantischen Einschränkungen relativ schwach, so daß die Beispiele als ein eher ungünstiger Fall angesehen werden müssen, dies gilt vor allem für die Beispiele mit mehr als einem Verb.

Der Eingabesatz **Der Mann liebt seine Oma** erzeugte ein Chart mit 36 Kanten und zog 1 Ausgabe nach sich. Die Analyse dauerte weniger als 0.2s.

Der Eingabesatz **Die Oma die einen Baum pflanzt** erzeugte ein Chart mit 46 Kanten und zog 1 Ausgabe nach sich. Die Analyse dauerte circa 0.2s.

Der Eingabesatz **Ich gehe in den Garten hinter dem Haus weil meine Oma einen Baum pflanzt.** erzeugte ein Chart mit 110 Kanten und zog 1 Ausgabe nach sich. Die Analyse dauerte circa 0.5s.

Der Eingabesatz **Die Oma die im Garten hinter dem Haus einen Baum pflanzt steht im Garten hinter dem Haus in der kleinen Stadt in dem Tal hinter dem Berg weil sie im Garten einen Baum pflanzt** erzeugte ein Chart mit 374 Kanten und zog 3 Ausgaben nach sich. Die Analyse dauerte circa 1.6s.

8.2 Ausbau und Ausbaufähigkeit

Es gibt eine ganze Reihe von möglichen Änderungen und Erweiterungen von **SARA**. Die wichtigsten davon, die die syntaktischen Fähigkeiten betreffen, sollen hier aufgezählt und ihr Aufwand eingeschätzt werden.

1. Die größte Schwäche in den syntaktischen Fähigkeiten von **SARA** ist zweifellos die Unfähigkeit zur Behandlung der Konjunktionen **und** und **oder**, und damit verbunden von Aufzählungen der Art **A, B und C**. Diese sind zwar im Datentyp **kante** und den Ausgabefunktionen des Zerteilers bereits vorgesehen, werden aber im Zerteilungsalgorithmus bisher in keiner Weise berücksichtigt. Ich vermute, daß die Erweiterung des Algorithmus um diese Fähigkeit recht weitreichender Änderungen bedarf und deshalb verhältnismäßig aufwendig ist.
2. Ein zweiter wichtiger Punkt wäre die Behandlung von Verneinungen. Zur Zeit kann man lediglich, das Wort **nicht** zum Modaladverb erklären, so daß es als Rollenfüller auftreten kann oder als Modifizierer in einer Nominalphrase. Eine vollständige Behandlung sollte aber automatisch stattfinden (also insbesondere auf Satzebene ohne eine Rolle vorzusehen), d.h. den richtigen Bezug auffinden und "versteckte" Verneinungen (wie zum Beispiel in dem Wort **kein**) in explizite Umwandeln. Dazu sind aber sowohl erhebliche Änderungen im Zerteiler, als auch eine Erweiterung der in der Wissensbasis dargestellten Information notwendig.
3. Die Beschreibungsmöglichkeiten für Rollen, die Formen in mehr als einer der Klassen nominal, verbal, adverbial zulassen, sind etwas unzufriedenstellend. Zwar kann man mit dem Substantivquerverweis bei Verbkonzepten die häufigsten dieser Fälle abdecken, jedoch nicht die adverbialen Formen. Die saubere Lösung wäre die, sich

von der Wortartbindung der Konzepte vollständig zu lösen, so daß ein Konzept grundsätzlich als polymorph anzusehen ist. Der Zerteiler würde dann die Wortartinformation nur noch dem Wörterbuch entnehmen. Diese Vorgehensweise macht keinerlei Schwierigkeiten und ist auch nachträglich relativ einfach in den Zerteiler einzubauen; in der Konzepthierarchie muß nötigenfalls Wortartinformation in der Synonymliste die Unterscheidung bei gleichlautenden Wortnamen ermöglichen. Daß dieses Prinzip nicht bereits realisiert ist, liegt lediglich daran, daß seine Vorteile von mir zu spät erkannt wurden.

4. Die Analyse von Relativsätzen kann momentan nur solche Relativsätze erkennen, deren Relativpronomen sich direkt an das Bezugswort anschließt. Das schließt erstens eine Verwendung von Relativsätzen mit Präposition aus und verbietet zweitens das Vorziehen eines Verbs vor den Relativsatz. Es wäre wünschenswert, beide Einschränkungen zu beseitigen. Zumindest bei der Zweiten ist das allerdings nicht so einfach, weil das Verb ggf. als Endpunkt des Nebensatzes angesehen wird.
5. Zur Erhöhung der Universalität des Zerteilers sollte gelegentlich ein Verbautomat nachgerüstet werden, der aus der Folge von Haupt-, Hilfs- und Modalverben die Angaben Zeit, Person und Numerus für einen Satz berechnet. In diesem Zusammenhang muß beim Zerteiler ermöglicht werden im Nebensatz auch nach dem Vollverb (das heute stets das Ende des Nebensatzes markiert) noch ein Hilfsverb und/oder Modalverb zu analysieren.
6. Leicht nachträglich einzufügen ist die Behandlung von Quotierungen (also Eingeborenen, die in Anführungszeichen eingeschlossen sind). Objekte dieser Art werden schon jetzt beim Aufbauen des Chart erkannt und stehen dem Zerteiler als Kanten zur Verfügung. Hier gilt es lediglich erst einmal genau zu beschliessen, wie und als was solche Objekte eigentlich verwendet werden, wie man sie demnach im Zerteiler beurteilen sollte und wo sie dementsprechend in welcher Weise eingebaut werden dürfen. Die technischen Änderungen dafür sind aber gering.
7. Verbesserungswürdig ist sicherlich die Behandlung von Zahlen. Momentan sind Zahlen nur als Modifizierer eines Substantivs erlaubt (ähnlich der Verwendung von Adjektiven) und es gibt nur ganze Zahlen. Die Verwendung von Zahlen ohne zugeordnetes Substantiv bedarf wie die Quotierungen nur einer Festlegung des Rahmens, in dem sie geschehen soll, die technische Seite ist aber einfach. Eine weitere Verbesserung wäre die Erlaubnis von Dezimal- oder Gleitkommazahlen und die Behandlung von angehängten Maßeinheiten.

Anhang A

Standard-Wissensbasis sara.std

```

/*****
Projekt : Fallschablonenzerteiler fuer Deutsch
Datei   : Standard-Wissensbasis
Autor   : Lutz Prechelt, Karlsruhe
Stand   : 01.10.89
RCS     : $Id: sara.std,v 1.4 89/10/01 22:35:39 prechelt Exp $
*****/

/* Dies ist die Standard-Wissensbasis fuer SARA.
   Sie enthaelt das Rollenverzeichnis sowie die grundlegenden
   Woerterbucheintraege, als da waeren:
   (f = fehlt, u = unvollstaendig, w = weitgehend vollstaendig, v = vollstaendig)
   Hilfsverben,           v
   Modalverben,          v
   Artikel,              v
   Demonstrativpronomen, w
   Possessivpronomen,   v
   Interrogativpronomen, v
   Personalpronomen,    v
   Reflexivpronomen,    v
   Relativpronomen,     w
   Konjunktionen,       w
   Modaladverben.       u

   Die Pronomen sind sehr stark mehrdeutig. Es ist daher sinnvoll, alle
   Formen, die fuer eine Anwendung nicht benoetigt werden, nicht mitzuladen.

   Die Dudenverweise beziehen sich auf die Randnummern in
   Duden, Band 4: Die Grammatik, 1984
*/

/***** Hilfsverben *****/
;gemaess (Duden 219)
{ sein      :hilfsverb
  :praesens (bin, bist, ist, sind, (seid,seit), sind)
  :praeteritum (war, warst, war, waren, wart, waren)
  :konjunktiv_I (sei, (seist,seiest), sei, seien, seiest, seien)
  :konjunktiv_II (waere, (waerst, waerest), waere,
                 waeren, (waert, waeret), waeren)
  :partizip (seiend, gewesen)
  :imperativ (sei, (seid, seit))
}

{ haben     :hilfsverb
  :praesens (habe, hast, habt, haben, habt, haben)
  :praeteritum (hatte, hattest, hatte, hatten, hattest, hatten)
  :konjunktiv_I (habe, habest, habe, haben, habet, haben)
  :konjunktiv_II (haette, haettest, haettet, haetten, haettet, haetten)
  :partizip (habend, gehabt)
  :imperativ (habe, habt)
}

{ werden   :hilfsverb
  :praesens (werde, wirst, wird, werden, werdet, werden)
  :praeteritum ((wurde, ward), wurdest, (wurde, ward), ;??
               wurden, wurdet, wurden)
  :konjunktiv_I (werde, werdest, werde, werden, werdet, werden)
  :konjunktiv_II (wuerde, wuerdest, wuerde, wuerden, wuerdet, wuerden)
  ; das Hilfsverb werden hat ein anderes 2. Partizp
  ; als das Vollverb werden, naemlich 'worden' statt 'geworden'.
  ; Diese Form wird zur Bildung des Passiv benutzt.
  :partizip (werdend, (worden, geworden))
  :imperativ (werde, werdet)
}

/***** Modalverben *****/
```

```

;hauptsaechlich gemaess (Duden 216)
{
  duerfen :modalverb
  :praesens (darf, darfst, darf, duerfen, duerft, duerfen)
  :praeteritum (durfte, durftest, durfte, durften, durftet, durften)
  :konjunktiv_I (duerfe, duerfest, duerfe, duerfen, duerfet, duerfen)
  :konjunktiv_II (duerfte, duerftest, duerfte, duerften, duerftet, duerften)
  :partizip (duerfend, gedurft)
  :imperativ ( )
}

{
  koennen :modalverb
  :praesens (kann, kannst, kann, koennen, koennt, koennen)
  :praeteritum (konnte, konntest, konnte, konnten, konntet, konnten)
  :konjunktiv_I (koenne, koennest, koenne, koennen, koennet, koennen)
  :konjunktiv_II (koennte, koenntest, koennte, koennten, koenntet, koennten)
  :partizip (koennend, gekonnt)
  :imperativ ( )
}

{
  moegen :modalverb
  :praesens (mag, magst, mag, moegen, moegt, moegen)
  :praeteritum (mochte, mochtest, mochte, mochten, mochtet, mochten)
  :konjunktiv_I (moege, moegest, moege, moegen, moeget, moegen)
  :konjunktiv_II ( )
  ;eigentlich moechte, moechtest, moechte, moechten, moechtet, moechten
  ;siehe jedoch bei 'wollen'
  :partizip (moegend, gemocht)
  :imperativ ( )
}

{
  muessen :modalverb
  :praesens (muss, musst, muss, muessen, muesst, muessen)
  :praeteritum (musste, musstest, musste, mussten, musstet, mussten)
  :konjunktiv_I (muesse, muessest, muesse, muessen, muesset, muessen)
  :konjunktiv_II (muesste, muesstest, muesste, muessten, muesstet, muessten)
  :partizip (muessend, gemusst)
  :imperativ ( )
}

{
  sollen :modalverb
  :praesens (soll, sollst, soll, sollen, sollt, sollen)
  :praeteritum (sollte, solltest, sollte, sollten, solltet, sollten)
  :konjunktiv_I (solle, sollest, solle, sollen, sollet, sollen)
  :konjunktiv_II (sollte, solltest, sollte, sollten, solltet, sollten)
  :partizip (sollend, gesollt)
  :imperativ ( )
}

{
  wollen :modalverb
  ;Die KonjunktivII-Formen von moegen werden auch als Praesensformen
  ;von wollen benutzt !
  :praesens ((will, moechte), (willst, moechttest), (will, moechte),
  (wollen, moechten), (wollt, moechtst), (wollen, moechten))
  :praeteritum (wollte, wolltest, wollte, wollten, wolltet, wollten)
  :konjunktiv_I (wolle, wollest, wolle, wollen, wollet, wollen)
  :konjunktiv_II (wollte, wolltest, wollte, wollten, wolltet, wollten)
  :partizip (wollend, gewollt)
  :imperativ ( )
}

/***** Artikel *****/

;gemaess (Duden 352)
{
  bestimmter_artikel :artikel
  :maennlich (der, des, dem, den)
  :weiblich (die, der, der, die)
  :saechlich (das, des, dem, das)
  :plural (die, der, den, die)
}

;gemaess (Duden 352)
{
  unbestimmter_artikel :artikel
  :maennlich (ein, eines, einem, einen)
  :weiblich (eine, einer, einer, eine)
  :saechlich (ein, eines, einem, ein)
  :plural ( )
}

;gemaess (Duden 551)
{
  demonstrativartikel :artikel
  :maennlich (dieser, (dieses, diesen), diesem, diesen)
  :weiblich (diese, dieser, dieser, diese)
  :saechlich ((dies, dieses), dieses, diesem, (dies, dieses))
  :plural (diese, dieser, diesen, diese)
}

;gemaess (Duden 551)
{
  demonstrativartikel :artikel
  :maennlich (jener, (jenes, jenen), jenem, jenen)
  :weiblich (jene, jener, jener, jene)
  :saechlich (jenes, jenes, jenem, jenes)
  :plural (jene, jener, jenen, jene)
}

;gemaess (Duden 554)

```

```

{ demonstrativartikel :artikel
:maennlich (derjenige, desjenigen, demjenigen, denjenigen)
:weiblich (diejenige, derjenigen, derjenigen, diejenige)
:saechlich (dasjenige, desjenigen, demjenigen, dasjenige)
:plural (diejenigen, derjenigen, denjenigen, diejenigen)
}

;gemaess (Duden 555)
{ demonstrativartikel :artikel
:maennlich (derselbe, desselben, demselben, denselben)
:weiblich (dieselbe, derselben, derselben, dieselbe)
:saechlich (dasselbe, desselben, demselben, dasselbe)
:plural (dieselben, derselben, denselben, dieselben)
}

/* Man beachte, dass bei den Possessivpronomen in der Verwendung als
Artikel das Geschlecht und der Numerus zweimal auftauchen:
Einmal fuer das Substantiv, mit dem sie stehen und einmal fuer
die Person, auf die sie sich beziehen.
Die Bezugsperson wird hier nicht identifiziert.
*/
{ possessivartikel :artikel
:maennlich ((mein, dein, sein, ihr, unser, euer, ihr),
(meines, deines, seines, ihres, unseres, eures, eueres, ihres),
(meinem, deinem, seinem, ihrem, unserem, eurem, euerem, ihrem),
(mein, dein, sein, ihr, unser, euer, ihr))
:weiblich ((meine, deine, seine, ihre, unsere, euere, eure, ihre),
(meiner, deiner, seiner, ihrer, unserer, eurer, euerer, ihrer),
(meiner, deiner, seiner, ihrer, unserer, eurer, euerer, ihrer),
(meine, deine, seine, ihre, unsere, euere, eure, ihre))
:saechlich ((mein, dein, sein, ihr, unser, euer, ihr),
(meines, deines, seines, ihres, unseres, eures, eueres, ihres),
(meinem, deinem, seinem, ihrem, unserem, eurem, euerem, ihrem),
(mein, dein, sein, ihr, unser, euer, ihr))
:plural ((meine, deine, seine, ihre, unsere, euere, eure, ihre),
(meiner, deiner, seiner, ihrer, unserer, eurer, euerer, ihrer),
(meinen, deinen, seinen, ihren, unseren, euren, eueren, ihren),
(meine, deine, seine, ihre, unsere, euere, eure, ihre))
}

/***** Demonstrativpronomen *****/

/* Demonstrativpronomen in Substantiv-Verwendung, siehe auch 'Artikel' */
/* gemaess (Duden 551,554,555) */

/*{ demonstrativpronomen :demonstrativpronomen
:geschlecht m
:singular (dieser, dieses, diesem, diesen)
:plural (diese, dieser, diesen, diese)
}

{ demonstrativpronomen :demonstrativpronomen
:geschlecht w
:singular (diese, dieser, dieser, diese)
:plural (diese, dieser, diesen, diese)
}

{ demonstrativpronomen :demonstrativpronomen
:geschlecht s
:singular ((dies, dieses), dieses, diesem, (dies, dieses))
:plural (diese, dieser, diesen, diese)
}*/

{ demonstrativpronomen :demonstrativpronomen
:geschlecht m
:singular (derjenige, desjenigen, demjenigen, denjenigen)
:plural (diejenigen, derjenigen, denjenigen, diejenigen)
}

{ demonstrativpronomen :demonstrativpronomen
:geschlecht w
:singular (diejenige, derjenigen, derjenigen, diejenige)
:plural (diejenigen, derjenigen, denjenigen, diejenigen)
}

{ demonstrativpronomen :demonstrativpronomen
:geschlecht s
:singular (dasjenige, desjenigen, demjenigen, dasjenige)
:plural (diejenigen, derjenigen, denjenigen, diejenigen)
}

{ demonstrativpronomen :demonstrativpronomen
:geschlecht m
:singular (derselbe, desselben, demselben, denselben)
:plural (dieselben, derselben, denselben, dieselben)
}

{ demonstrativpronomen :demonstrativpronomen
:geschlecht w
:singular (dieselbe, derselben, derselben, dieselbe)
:plural (dieselben, derselben, denselben, dieselben)
}

{ demonstrativpronomen :demonstrativpronomen
:geschlecht s

```

```

:singular (dasselbe, desselben, demselben, dasselbe)
:plural (dieselben, derselben, denselben, dieselben)
}

/***** Possessivpronomen *****/
/* Possessivpronomen in Substantiv-Verwendung, siehe auch 'Artikel' */
/* gemaess (Duden 545ff) */
{ possessivpronomen :possessivpronomen
:geschlecht (m,w,s)
:person erste
:singular ((meins, meines), meines, meinem, (meins, meines))
:plural (meine, meiner, meinen, meine)
}
{ possessivpronomen :possessivpronomen
:geschlecht (m,w,s)
:person zweite
:singular ((deins, deines), deines, deinem, (deins, deines))
:plural (deine, deiner, deinen, deine)
}
{ possessivpronomen :possessivpronomen
:geschlecht (m,s)
:person dritte
:singular ((seins, seines), seines, seinem, (seins, seines))
:plural (seine, seiner, seinen, seine)
}
{ possessivpronomen :possessivpronomen
:geschlecht (w)
:person dritte
:singular ((ihrs, ihres), ihres, ihrem, (ihrs, ihres))
:plural (ihre, ihrer, ihren, ihre)
}

/***** Interrogativpronomen *****/
/* gemaess (Duden 558ff) */
{ interrogativpronomen :interrogativpronomen
:maskulinum (welcher, wessen, welchem, welchen)
:femininum (welche, wessen, welcher, welche)
:neutrum (welches, wessen, welchem, welches)
:plural (welche, wessen, welchen, welche)
}

/***** Personalpronomen *****/
{ personalpronomen :personalpronomen
:person erste
:singular (ich, meiner, mir, mich)
:plural (wir, unser, uns, uns)
}
{ personalpronomen :personalpronomen
:person zweite
:singular ((du, sie), (deiner, ihrer), (dir, ihnen), (dich, sie))
:plural ((ihr, sie), (euer, ihrer), (euch, ihnen), (euch, sie))
}
{ personalpronomen :personalpronomen
:geschlecht maennlich
:person dritte
:singular (er, seiner, ihm, ihn)
:plural (sie, ihrer, ihnen, sie)
}
{ personalpronomen :personalpronomen
:geschlecht weiblich
:person dritte
:singular (sie, ihrer, ihr, sie)
:plural (sie, ihrer, ihnen, sie)
}
{ personalpronomen :personalpronomen
:geschlecht saechlich
:person dritte
:singular (es, seiner, ihm, es)
:plural (sie, ihrer, ihnen, sie)
}

/***** Reflexivpronomen *****/
/* gemaess (Duden 541) */
{ reflexivpronomen :reflexivpronomen
:geschlecht (m, w, s)
:person erste
:singular (-, meiner, mir, mich)
:plural (-, unser, uns, uns)
}

```

```

{ reflexivpronomen :reflexivpronomen
  :geschlecht (m, w, s)
  :person zweite
  :singular (-, deiner, dir, dich)
  :plural (-, euer, euch, euch)
}

{ reflexivpronomen :reflexivpronomen
  :geschlecht (m, s)
  :person dritte
  :singular (-, seiner, sich, sich)
  :plural (-, ihrer, sich, sich)
}

{ reflexivpronomen :reflexivpronomen
  :geschlecht (w)
  :person dritte
  :singular (-, ihrer, sich, sich)
  :plural (-, ihrer, sich, sich)
}

/***** Relativpronomen *****/
/* gemaess (Duden 559,561) */
{ Relativpronomen :relativpronomen
  :maskulinum (der, dessen, dem, den)
  :femininum (die, deren, der, die)
  :neutrum (das, dessen, dem, das)
  :plural (die, (deren, derer), denen, die)
}

{ Relativpronomen :relativpronomen
  :maskulinum (welcher, dessen, welchem, welchen)
  :femininum (welche, deren, welcher, welche)
  :neutrum (welches, dessen, welchem, welches)
  :plural (welche, deren, welchen, welche)
}

/* Ferner wer, was und formelhaft was fuer ein */
/***** Konjunktionen *****/
{ und :konjunktion }
{ oder :konjunktion }
{ sowie :konjunktion }
{ aber :konjunktion }
{ sondern :konjunktion }

{ waehrend :konjunktion }
{ als :konjunktion }
{ nachdem :konjunktion }
{ seitdem :konjunktion }
{ bis :konjunktion }
{ ehe :konjunktion }
{ insofern :konjunktion }
{ insoweit :konjunktion }
{ weil :konjunktion }
{ da :konjunktion }
{ dass :konjunktion }
{ ob :konjunktion }
{ wenn :konjunktion }
{ obwohl :konjunktion } ;etc.

/***** Modaladverben *****/
/* Nur solche zur Gradabschattung bei Adjektiven */
{ beinah :modaladverb }
{ einigermassen :modaladverb }
{ etwas :modaladverb }
{ fast :modaladverb }
{ halbwegs :modaladverb }
{ recht :modaladverb }
{ sehr :modaladverb }
{ wirklich :modaladverb }
{ ziemlich :modaladverb }

/***** ROLLEN *****/
;----- Personen oder wie Personen beteiligte Objekte
<< agent :subjekt ;wird durch einen Imperativ erledigt
  :nur_verbale
  :nominativ ;'die Mutter' geht ...
  :nur_nominal
  :genitiv ;der Tanz 'meines Vaters'
  :frage wer
  :frage was
  :fragenominativ ()
>>

<< koagent
  :dativ mit ;die Mutter geht 'mit dem Kind' einkaufen
  :frage (mit wem)
  :fragedativ (mit)

```

```

>> :fragedativ (zusammen mit)
>>
<< patient
  :dativ                ;er schlug 'dem Gegner' den Ball um die Ohren
  :frage wem
  :fragedativ ()
>>
<< nutzniesser
  :dativ                ;ich schenke 'dem Geburtstagskind' ...
  :akkusativ fuer      ;ich bastele 'fuer das Geburtstagskind' ...
  :frage wem
  :frage (fuer wen)
  :fragedativ ()
  :frageakkusativ (fuer)
>>
<< eigentuemer
  :genitiv              ;die Hosen 'meines Vaters' sind ...
  :dativ von           ;die Hosen 'von meiner Mutter' sind ...
  :frage wessen
  :fragegenitiv ()
>>
;----- Allgemeine Rollen
<< thema ;"affiziertes Objekt"
  :akkusativ          ;ich sehe 'eine schoene Landschaft'.
  :frage wen
  :frage was
  :frageakkusativ ()
>>
<< genitiv ;!@#%~&*(@$#%$~%~&&&*, aber was soll's...
  :genitiv
  :frage wessen
  :fragegenitiv ()
>>
<< definition
  :nominativ          ;Der Junge ist 'ein Flegel'.
  :frage was
  :frage (von welcher Art)
>>
;----- Rollen von Sachen
<< attribut
  :dativ mit          ;Der Spinat 'mit dem Blub'
  :frage (mit was)
  :frage (mit wem) ;?
  :frage womit
  :fragedativ (mit)
>>
<< material
  :dativ aus          ;Diese Wand ist 'aus Beton'.
  :frage (aus was)
  :frage woraus
  :fragedativ (aus)
>>
<< werkzeug
  :genitiv mittels    ;Ich gehe 'mittels meiner Fuesse'
  :dativ mittels      ;du gehst 'mittels deinen Fuessen'
  :dativ mit          ;sie geht 'mit ihren Fuessen'
  :akkusativ durch    ;ich schaffe es 'durch meine guten Fuesse'.
  :frage (mit was)
  :frage womit
  :frage (durch was)
  :frage wodurch
  :fragedativ (mit)
  :frageakkusativ (durch)
>>
<< resultat ;"effiziertes Objekt"
  :akkusativ          ;ich backe '3 Torten'
  :frage was
  :frageakkusativ ()
>>
;----- Rollen des Raums
<< ort
  :dativ in           ;ich bin 'in einem Zimmer'
  :dativ im           ;ich bin 'im Haus'
  :dativ auf          ; etc.
  :dativ vor
  :dativ hinter
  :dativ hinterm
  :dativ neben
  :dativ unter
  :dativ unterm
  :dativ bei
  :dativ beim

```

```

:dativ an
:dativ am
:genitiv rechts ;** "rechts von" koennen wir leider nicht. **
:genitiv links ;es steht 'links des Bahndammes'
:nur_nominal
:genitiv ;die Tuer 'des Zimmers'
:frage worin
:frage worauf
:frage wovor
:frage wohinter
:frage woneben
:frage worunter
:frage wo
:frage woran
:frage (rechts wovon)
:frage (links wovon)
:frage (rechts wessen)
:frage (links wessen)
:fragegenitiv (rechts)
:fragegenitiv (links)
:fragedativ (rechts von)
:fragedativ (links von)
:fragedativ (in)
:fragedativ (auf)
:fragedativ (vor)
:fragedativ (neben)
:fragedativ (hinter)
:fragedativ (unter)
:fragedativ (bei)
:fragedativ (an)
>>

<< startpunkt
:dativ ab ;sie reisen 'ab Koeln'
:dativ aus ;sie holt ihr Fahrrad 'aus dem Keller'
:dativ von ;ich falle 'von der Treppe'
:dativ vom ;ich falle 'vom Fahrrad'
:frage (woraus)
:frage (woher)
:frage (von wo)
:frage (von wo aus)
:frage (von wo ab)
:frage (von wo her)
:fragedativ (von)
:fragedativ (ab)
>>

<< zielpunkt
:akkusativ in ;ich gehe 'in das Badezimmer'
:akkusativ ins ;ich komme 'ins Dunkle'
:akkusativ auf ; etc.
:akkusativ aufs
:akkusativ vor
:akkusativ vors
:akkusativ hinter
:akkusativ hinters
:akkusativ neben
:akkusativ nebens
:akkusativ unter
:akkusativ unters
:akkusativ an
:akkusativ ans
:dativ zu
:dativ zum
:frage wohin
:frage (zu wem)
:frage (zu was)
:frageakkusativ (in)
:frageakkusativ (auf)
:frageakkusativ (vor)
:frageakkusativ (hinter)
:frageakkusativ (neben)
:frageakkusativ (unter)
:frageakkusativ (an)
:fragedativ (zu)
>>

<< richtung ; wie zielpunkt !?
:akkusativ in
:akkusativ ins
:akkusativ auf
:akkusativ aufs
:akkusativ vor
:akkusativ vors
:akkusativ hinter
:akkusativ hinters
:akkusativ neben
:akkusativ nebens
:akkusativ unter
:akkusativ unters
:akkusativ an
:akkusativ ans
:dativ zu
:dativ zum
:frage wohin
:frage (zu wem)

```

```

:frage (zu was)
:frageakkusativ (in)
:frageakkusativ (auf)
:frageakkusativ (vor)
:frageakkusativ (hinten)
:frageakkusativ (neben)
:frageakkusativ (unter)
:frageakkusativ (an)
:fragedativ (zu)
>>

<< bahn
:genitiv entlang ;gehen sie 'entlang weisser Linien'!
:dativ entlang ;gehen sie 'entlang den weissen Linien'!
:akkusativ ueber ;gehen sie 'ueber die Strasse'
:akkusativ durch ;gehen sie 'durch den Tunnel'
:frage worueber
:frage (wo entlang)
:frage (wo durch)
:frage (entlang wessen)
:frageakkusativ (durch)
:fragedativ (ueber)
:fragegenitiv (entlang)
:fragedativ (entlang)
>>

<< erstreckung
:genitiv entlang ;es fuehrt 'entlang des Rheins'
:dativ entlang ;es fuehrt 'entlang dem Rhein'
:frage wohin
:frage wo
:frage (wo entlang)
:frage (entlang wessen)
:fragegenitiv (entlang)
:fragedativ (entlang)
>>

;----- Rollen der Zeit

<< zeitpunkt
:temporaladverbial ;'gestern' ging es noch
:frage wann
:fragedativ an
:fragedativ in
>>

<< beginn
:dativ ab ;'ab 2 Uhr' bin ich wieder da
:dativ seit ;'seit 1 Uhr' mache ich Mittagspause
:frage (ab wann)
:frage (seit wann)
>>

/*
<< ende
:frage (bis wann)
>>
*/

<< dauer
:akkusativ ;sie dauert '60 Minuten'
:frage (wie lange)
>>

;----- Rollen der Art und Weise

<< art_und_weise
:nominativ wie ;er geht 'wie seine Oma'
:dativ mit ;er macht es 'mit Begeisterung'
:modaladverbial ;er macht es 'begeistert'
:nebensatz indem ;er macht es, 'indem er lacht'
:frage wie
>>

<< grad
:modaladverbial ;er stottert 'sehr'
:frage wie
:frage (wie sehr)
:frage (wie stark)
>>

;----- Rollen zur Begrueundung

<< grund
:genitiv wegen ;ich warte 'wegen des Regens'
:dativ wegen ;ich warte 'wegen dem Regen'
:akkusativ durch ;ich schaffe es 'durch meine Geduld'
:nebensatz da ;ich warte, 'da die Sonne scheint'
:nebensatz weil ;ich warte, 'weil es schneit'
:frage wieso
:frage weshalb
:frage warum
:fragegenitiv (wegen)
:fragedativ (wegen)
>>

```

```
<< zweck
    :genitiv zwecks      ;ich trinke 'zwecks meiner Belustigung'
    :akkusativ fuer     ;ich bete 'fuer mein Seelenheil'
    :nebensatz um       ;** zu? **
    :nebensatz damit    ;ich renne, damit ich die Strassenbahn erreiche
:frage wieso
:frage weshalb
:frage warum
:frage wofuer
:frage wozu
:fragegenitiv (wegen)
:fragedativ (wegen)
:frageakkusativ (fuer)
>>
```

Anhang B

Beispiel–Wissensbasis

Literatur

*Füge eine Kleinigkeit zur anderen
und das Ergebnis wird ein großer Haufen sein.*

Ovid

- [Barr81] Avron Barr, Edward A. Feigenbaum: *The Handbook of Artificial Intelligence, Volume 1*. William Kaufman Inc, Los Altos, California, 1981
- [Bodsb89] Nils Bodsberg: *A Caseframe Parser for English*. Diplomarbeit, Universität Karlsruhe, 1989
- [Brooks75] Frederick P. Brooks: *Der Mythos des Mann-Monats*. Addison-Wesley Deutschland, 1987
- [Bünting71] Karl-Dieter Bünting: *Einführung in die Linguistik*. 2. Auflage, Athenäum Verlag, Frankfurt, 1971
- [Dreyfus86] Hubert L. and Stuart E. Dreyfus: *Mind Over Machine*. Basil Blackwell Ltd, Oxford, 1986
- [Duden(4)] Die Dudenredaktion: *Der Duden Band 4 — Grammatik*. Bibliographisches Institut Mannheim, 1984
- [Duden(5)] Die Dudenredaktion: *Der Duden Band 5 — Fremdwörterbuch*. Bibliographisches Institut Mannheim, 1974
- [Fairley85] Richard E. Fairley: *Software Engineering Concepts*. McGraw-Hill, 1985
- [Fillmore68] Charles J. Fillmore: *The Case for Case*. in Emmon Bach, Robert T. Harms (ed): *Universals in Linguistic Theory*, Holt, Rinehart and Winston Inc, London, Reprint 1972
- [FinNeu86] Wolfgang Finkler, Günter Neumann: *Morphix — Ein hochportabler Lemmatisierungsmodul für das Deutsche*. Universität des Saarlandes, Saarbrücken, KI-Labor am Lehrstuhl für Informatik IV, Memo Nr. 8, Juli 1986
- [Görz88] Günther Görz: *Strukturanalyse natürlicher Sprache*. Addison-Wesley Deutschland, 1988
- [GooWai84] Gerhard Goos, William M. Waite: *Compiler Construction*. Springer New York, 1984

- [HayCar81] P.J. Hayes, J.G. Carbonell: *Robust Parsing Using Multiple Construction-Specific Strategies*. in L. Bolc (ed.): *Natural Language Parsing Systems*, Springer New York, 1985, oder: *Proceedings of 7th International Joint Conference on Artificial Intelligence*, 1981
- [HayCar87] P.J. Hayes, J.G. Carbonell: *Natural Language Understanding*. in Stuart C. Shapiro (ed.): *Encyclopedia of Artificial Intelligence*, Wiley Interscience Publication, John Wiley and Sons, New York, 1987
- [KerRit83] Brian W. Kernighan, Dennis M. Ritchie: *Programmieren in C*. Carl Hanser, München, Wien, 1983
- [Knuth84] Donald E. Knuth: *The T_EXbook*. Addison-Wesley, 1986
- [Kopka88] Helmut Kopka: *L^AT_EX — Eine Einführung*. Addison-Wesley Deutschland, 1988
- [Lindsay63] Robert K. Lindsay: *A Program for Parsing Sentences and Making Inferences about Kinship Relations*. Symposium on Simulation Models, Methodology and Applications to the Behavioral Sciences (ed: A.C. Hoggatt, F.E. Balderston), South-Western Publishing, Cincinnati, 1963
- [Myers79] Glenford J. Myers: *Methodisches Testen von Programmen*. Oldenbourg Verlag, 1982
- [Newbery88] Frances J. Newbery: *EDGE: An Extensible Directed Graph Editor*. Interner Bericht 8/88, Fakultät für Informatik, Universität Karlsruhe, 1988
- [SchaRies81] Roger C. Schank, Christopher K. Riesbeck (ed.): *Inside Computer Understanding*. Lawrence Erlbaum Associates, Inc., Hillsdale, New Jersey, 1981
- [Schnei82] Wolf Schneider: *Deutsch für Profis*. Stern-Buch bei Gruner und Jahr, 2. Auflage 1982
- [Steele84] Guy L. Steele Jr.: *Common Lisp — The Language*. Digital Press, 1984
- [Strou87] Bjarne Stroustrup: *Die C++ Programmiersprache*. Addison-Wesley Deutschland, 1987
- [TiAdHo88] Walter F. Tichy, Rolf Adams, Lars Holter: *NLH/E — A Natural Language Help System*. Proceedings of the International Conference on Software Engineering, 1989
- [Uszkoreit87] Hans Uszkoreit: *Wort Order and Constituent Structure in German*. Center for the Study of Language and Information, CSLI/Stanford, Lecture Notes 8, Ventura Hall, California, 1987
- [Weizenb66] Joseph Weizenbaum: *ELIZA — A Computer Program for the Study of Natural Language Communication Between Man and Machine*. CACM 9, 1966
- [Winograd83] Terry Winograd: *Language as a Cognitive Process, Volume 1: Syntax*. Addison-Wesley, Reading, Massachusetts, 1983

- [Winston81] Patrick Henry Winston: *Artificial Intelligence*. Addison-Wesley, Reading, Massachusettes, corrected reprint 1984
- [Woods73] William A. Woods: *Progress in Natural Language Understanding: An Application to Lunar Geology*. 1973 National Computer Conference, AFIPS Conference Proceedings 42, AFIPS Press, Montvale, New Jersey, 1973
- [Zorach80] Cecile Zorach: *English Grammar for Students of German*. The Olivia and Hill Press Inc, Ann Arbor, Michigan, 7th printing 1987