# $\mathcal{PARLEX}$ und der Semantische Inspektor

## Werkzeuge für die Repräsentation und Analyse natürlicher Sprache

Dissertation
zur
Erlangung der Doktorwürde
der Philosophischen Fakultäten
der Universität Augsburg

vorgelegt von
Rolf Widdig

Berichterstatter

1. Prof. Dr. Hans-Jürgen Heringer

2. Prof. Dr. Bruno Strecker

*Tag der mündlichen Prüfung 16. Mai 2000*

*Copyright by rolf.widdig@oocon.de*

# Inhaltsverzeichnis

# Abbildungsverzeichnis

# Tabellenverzeichnis

# Überblick

Diese Arbeit ist während meiner Zeit am Institut für Informatik und Gesellschaft der Albert-Ludwigs-Universität Freiburg entstanden. Der erste Teil, der auch den Hauptteil der Arbeit ausmacht, beschreibt das natürlichsprachliche System $\mathcal{PARLEX}$. Hierbei steht die Entwicklung der lexikalischen Wissensbasis im Vordergrund.
Während dieser Zeit entstand in Zusammenarbeit mit Prof. Heringer auch das Programm des Semantischen Inspektors. Liegt im ersten Teil meiner Arbeit der Schwerpunkt auf der Beschreibung und Repräsentation natürlichsprachlicher Phänomene, so kann der Semantische Inspektor als ein Werkzeug der *quantitativen Linguistik* angesehen werden. Die quantitative Linguistik unterscheidet sich nach Köhler und Rieger [KR93] zwar nicht grundsätzlich in ihren Zielen von anderen Richtungen der Linguistik, durch den Einsatz von mathematischen Modellen wohl aber durch ihre Methoden.
Ich möchte mich an dieser Stelle bei Prof. Britta Schinzel bedanken, die viel zu meiner informatischen Ausbildung beigetragen hat und die mir die Möglichkeit zur Erstellung dieser Arbeit gab. Bei Prof. Hans Jürgen Heringer für die Betreuung von Seiten der Linguistik.

# Teil I

# Das „natürlichsprachliche System“ $\mathcal{PARLEX}$ und seine lexikalische Wissensbasis

# Kapitel 1

# Einleitung

Natürlichsprachliche Systeme, im folgenden mit NLP[1]-Systemen bezeichnet, müssen Kenntnisse über den Sprachgebrauch haben. Dabei ist festzustellen, daß NLP-Systeme, aber auch „native speaker“, nie den vollen Kenntnisstand des Sprachgebrauchs erreichen können. Dies gilt für das lexikalische Wissen in einer Dimension und Unausweichlichkeit, wie es bzgl. einer im Prinzip mit endlichen Beschreibungsmitteln explizit erfaßbaren Grammatik nicht der Fall ist. Lange Zeit wurde jedoch dem Lexikon nur geringe Bedeutung zugeordnet. Z.B. sah Chomsky das Lexikon als einen Behälter, in dem die Irregularitäten gesammelt werden, die Grammatik als das Beschreibungsmittel für die Regularitäten einer natürlichen Sprache[2].

Doch mittlerweile wurde erkannt, daß der praktische Einsatz von natürlichsprachlichen Systemen in starkem Maße von der Verfügbarkeit von lexikalischem Wissen abhängt und erhebliche Anstrengungen auf die Organisation einer lexikalischen Wissensbank gelegt werden müssen. So sieht Nirenburg in dem Fehlen von „ansehnlichen“ Lexika, die insbesondere Einträge über Wortbedeutungen enthalten, den Grund, daß NLP-Systeme nur in einem „Demo-Mode“ operieren [Nir94]:

*„Current natural language processing systems typically operate in a*

[1] Natural Language Processing

[2] Diese strikte Trennung zwischen Struktur (Grammatik) und Lexik wurde von Halliday ironisch als „Backstein-Mörtel-Theorie“ bezeichnet (vgl. [SR90]).

> *„demo“ mode - they sometimes feature sizeable grammars but seldom sizeable lexicons containing information about meaning.“*

Zernik bezeichnet das Lexikon als den „Flaschenhals“ der Verarbeitung natürlicher Sprache [Zer91]:

> *„The lexicon has emerged as the major natural language processing bottleneck. Currently language processing is hampered by gaps in lexicons.“*

Im wissenschaftlichen Bereich weisen neuere linguistische Theorien (z.B. LFG[3], GPSG[4] und FUG[5]) dem Lexikon eine zentrale Rolle in der Sprachverarbeitung zu, indem sie einen Großteil ihrer Beschreibungen in das Lexikon verlagern [Lud93]. Ein Großteil der Forschungen im Bereich der Lexikalischen Datenbanken sind Ressourcen orientiert, in der Weise, daß existierende konventionelle Lexika als Basis genommen werden und untersucht wird, wieviel Information aus den Lexika für NLP-Systeme wiederverwendet werden kann. Beispiele für diese Forschungsrichtung finden sich in [BB89] und für den deutschsprachigen Raum in [Lud93].

In diesem Teil der Arbeit wird eine lexikalische Wissensbasis für das natürlichsprachliche System $\mathcal{PARLEX}$ beschrieben. Das folgende Kapitel gibt einen Überblick zur maschinellen Verarbeitung natürlicher Sprache. Anschließend beschreibe ich das Gesamtsystem $\mathcal{PARLEX}$ und erläutere die theoretischen Grundlagen des Parsers. Bevor ich die lexikalische Wissensbasis von $\mathcal{PARLEX}$ vorstelle, behandle ich in Kapitel 4 allgemein Lexika und lexikalische Wissensbasen.

[3] Lexical Functional Grammar [KB88]
[4] Generalized Phrase Structure Grammar [GKPS85]
[5] Functional Unification Grammar [Kay84]

# Kapitel 2

# Maschinelle Verarbeitung natürlicher Sprache

Die maschinelle Verarbeitung natürlicher Sprache, im Unterschied zu formalen oder Kalkülsprachen wie in der Mathematik, der Logik oder diversen Programmiersprachen, ist eines der zentralen Forschungsgebiete der künstlichen Intelligenz. Es gibt zwei grundlegende Motivationen für die Erstellung von Systemen zur Verarbeitung natürlicher Sprache [GW95]:

1. Die *theoretische Herangehensweise* legt linguistische oder psycholinguistische Fragestellungen zugrunde und möchte ein System als Testbett zur Validierung einer bestimmten Theorie benutzen.

2. Aus der *Ingenieursperspektive* kann man fragen, wie NLP-Systeme für bestimmte Anwendungsgebiete zu konstruieren sind.

Während die theoretische Perspektive vorwiegend in der Linguistik und den Kognitionswissenschaften eingenommen wird, liegt im Bereich der Künstlichen Intelligenz der Schwerpunkt meist auf der technologischen Perspektive [Bur96].

> Warum ist aber die Entwicklung von Sprachverarbeitungssystemen so schwierig?

Die Antwort findet sich bei der Betrachtung der Unterschiede von natürlichen und formalen Sprachen. Formale Sprachen sind durch die eindeutige Festlegung

von Syntax und Semantik gekennzeichnet. Das heißt, der Sprachumfang formaler Sprachen ist genau definiert und in der Regel nicht sehr groß und die Relationen zwischen den Sprachelementen sind eindeutig festgelegt.
Im Gegensatz dazu zeichnen sich natürliche Sprachen durch ihren umfangreichen Wortschatz und ihre vielfältigen generellen Ausdrucksmittel aus. Sie sind dynamisch erweiterbar und ihre Regeln verändern sich für den einzelnen Sprecher zwar unbemerkt, aber kontinuierlich. Sie sind durch Mehrdeutigkeiten auf verschiedenen sprachlichen Ebenen gekennzeichnet. Ein weiteres charakteristisches Merkmal ist die hohe Fehlertoleranz bei der Verwendung natürlicher Sprache. Betrachtet man nämlich die menschliche Kommunikation, so kann man feststellen, daß sich Menschen verstehen, obwohl sie die (grammatischen) Regeln ihrer natürlichen Sprache verletzen. Sie deuten Äußerungen auf der Grundlage von Erfahrungen, die sie mit Abweichungen von der Grammatik gemacht haben und aufgrund des Kontextes in dem die Äußerung gefallen ist. Diese Eigenschaften natürlicher Sprache erklären viele der Probleme bei der Entwicklung von Systemen zur maschinellen Sprachverarbeitung.
Ziel dieser Arbeit ist es, lexikalisches Wissen für ein sprachverarbeitendes System zur Verfügung zu stellen, das die Beschreibung und Analyse natürlicher Sprache[1] bzgl. grammatischer Strukturen erlaubt. Bei der Betrachtung von Systemen zur Verarbeitung natürlicher Sprache ergibt sich allgemeiner die Frage:

> Wie können sprachliche Äußerungen in eine Form überführt werden,
> die ihre Bedeutung widerspiegeln?

Bei der Entwicklung solcher Systeme ist aufgrund der Komplexität eine weitgehende Modularisierung notwendig. Naheliegend wäre eine Gliederung der Module entsprechend den Abstraktionsebenen Syntax, Semantik und Pragmatik. Obwohl unser Ansatz grundsätzlich auf der syntaktischen Ebene operiert, wobei jedoch semantische Informationen zur Auflösung von syntaktischen Mehrdeutigkeiten verwendet werden, möchte ich hier alle drei Ebenen kurz beschreiben.
Bei der syntaktischen Analyse wird die grammatische Struktur einer natürlichsprachlichen Äußerung ermittelt. Eine Eingabesequenz wird bei der Analyse in

[1] Wir betrachten nur geschrieben Sprache, die in elektronischer Form vorliegt.

eine (hierarchische) Konstituentenstruktur überführt. Auf der Ebene der Syntax kann man zwischen prozeduralen und deklarativen Teilen unterscheiden. Durch die Definition einer Grammatik wird festgelegt, welche syntaktischen Strukturen zu einer Sprache bzw. einem Sprachausschnitt gehören und in welche Strukturen sie durch den Verarbeitungsprozeß überführt werden. Die Grammatik wird von der prozeduralen Komponente, dem Parser, zur Analyse verwendet. Wie wir später sehen werden, wurde für unser System ein sogenannter Automatengenerator entwickelt, der die grammatischen Regeln in erkennende Automaten übersetzt. Bei der semantischen Analyse soll die Bedeutung einer sprachlichen Äußerung ermittelt werden. Um eine Abgrenzung bzgl. der pragmatischen Analyse zu erhalten, könnte man von wörtlicher oder textueller Bedeutung sprechen. Es wird also eine innersprachliche Bedeutungsebene angenommen, die unabhängig von der Äußerungssituation und dem außersprachlichen Wissen wäre. Wenn man das Kompositionalitätsprinzip zugrundelegt, kann die syntaktische und semantische Analyse parallel durchgeführt werden. Das Kompositionalitätsprinzip oder Frege-Prinzip[2] besagt:

> Die Bedeutung eines komplexen Ausdrucks folgt aus der Bedeutung der Teile und der Art ihrer Zusammensetzung.

Ohne ein Kompositionalitätsprinzip in irgendeiner Weise zu berücksichtigen, kann man nicht ernsthaft Semantik betreiben [Pin95]. Die Frage der Striktheit ist jedoch unklar. Es finden sich verschiedene Varianten des Kompositionalitätsprinzips, wobei es sich häufig um Verschärfungen handelt (siehe auch [Par84]). Solche Verschärfungen gewinnt man insbesondere durch Eingrenzung zugelassener Arten syntaktischer Zusammensetzungen und durch Eingrenzung des Bereichs der zugelassenen Funktionen, die die Bedeutungen der Teilausdrücke und der Arten ihrer Zusammensetzung auf die Bedeutung der komplexeren Ausdrücke abbilden. Formal kann das Frege-Prinzip so gedeutet werden, daß es zu jeder syntaktischen Operation eine simultane semantische Operation geben muß, welche die Aus-

[2] Obgleich eine Formulierung dieses Prinzips bei Gottlob Frege nicht vorkommt, wird es oft als Fregesches Prinzip bezeichnet, da die Fregesche Semantik als eine der entscheidenden Wegbereiter der kompositionell-semantischen Ansätze gilt.

wirkung der syntaktischen Komposition auf die wörtliche Bedeutung des entstehenden Ausdrucks charakterisiert. Ein Beispiel für einen Ansatz, der auf diesem Prinzip basiert, ist die Montague-Semantik [Tho79, Geb78, Mon73]. Montague hat das Fregesche Kompositionalitätsprinzip als einen Homomorphismus[3] zwischen Semantik und Syntax formalisiert.

Eine genaue Abgrenzung der pragmatischen Analyse erweist sich als schwierig. Während sich die Komponenten der Syntax und Semantik mit den strukturellen Eigenschaften sprachlicher Ausdrücke beschäftigt, wird bei der pragmatischen Analyse untersucht, wie sich diese strukturellen Eigenschaften bei der Verwendung der Ausdrücke in Bezug auf einen Äußerungskontext auswirken. Zu den Aufgaben der pragmatischen Analyse zählt man üblicherweise:

- die Auflösung von satzinternen oder satzübergreifenden Beziehungen mit Hilfe von Weltwissen;

- die Analyse von Diskursen und Dialogen sowie das Erkennen von Intentionen der beteiligten Personen.

Offensichtlich ist es kaum möglich, eine klare Trennung zwischen den Ebenen der syntaktischen, semantischen und pragmatischen Analyse zu definieren. Außerdem muß aufgrund der auf jeder Ebene auftretenden Ambiguitäten von komplexen Wechselwirkungen zwischen den einzelnen Ebenen ausgegangen werden.

Um dieser unscharfen Trennung und den komplexen Wechselwirkungen gerecht zu werden, wird in Abbildung 2.1 zwischen morpho-syntaktischer, syntaktisch-semantischer und semantisch-pragmatischer Analyse unterschieden.

[3] Montague verwendete den Begriff des Homomorphismus um die Strukturähnlichkeit der semantischen Ebene zur syntaktischen Ebene auszudrücken. Dabei definierte er die beiden Ebenen als zwei (uninterpretierte) Sprachen.

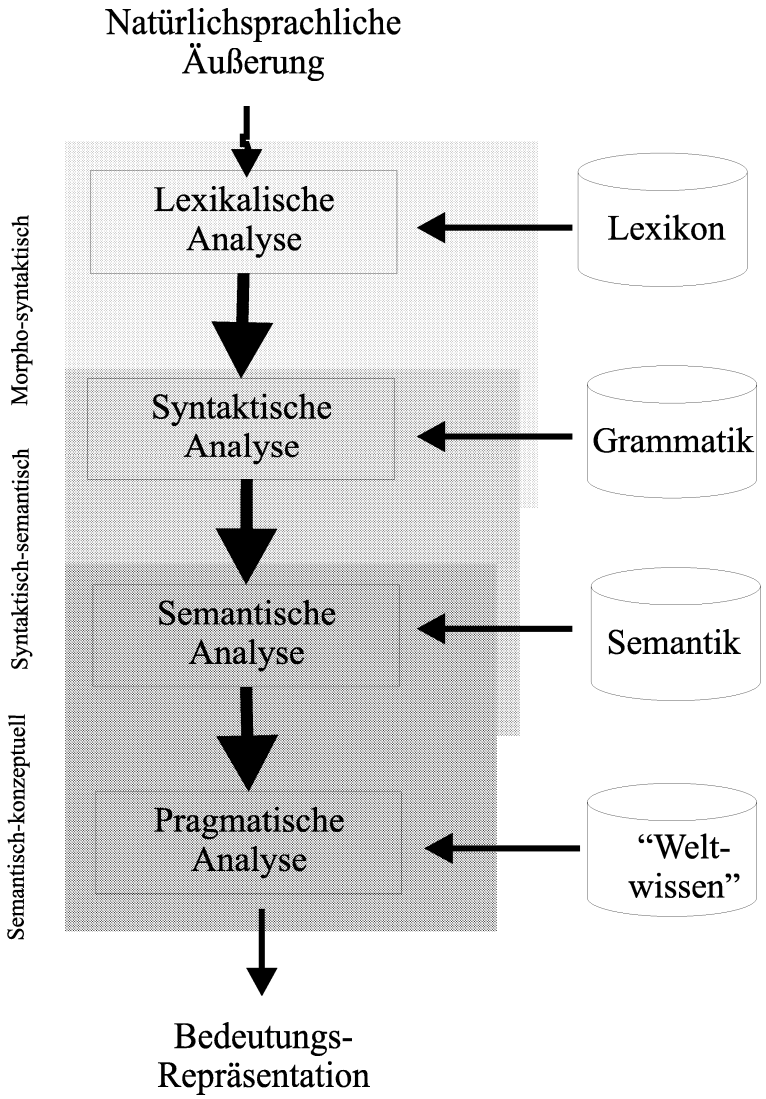


Abbildung 2.1: Verarbeitungsmodell zur Analyse natürlicher Sprache

Vor der syntaktischen Analyse einer sprachlichen Äußerung sind noch eine Reihe von Aufgaben durchzuführen, die wir in der Ebene der lexikalischen Analyse[4] zusammenfassen. Der zu analysierende Text muß zuerst in diskrete Einheiten wie mögliche lexikalische Einheiten und Interpunktionszeichen zerlegt werden.

[4]Die lexikalische Analyse ist in dem System der lexikalischen Wissensbasis in der Lemmatisierungskomponente integriert. Eine ausführliche Besprechung der lexikalischen Analyse findet sich in 5.5.

Nach dem Segmentierungsschritt kann die morphologische Analyse erfolgen, die den einzelnen Token lexikalische Einheiten mit ihren morphosyntaktischen und semantischen Informationen zuordnet. Die morphologische Analyse greift auf ein mehr oder weniger umfangreiches Lexikon zurück (siehe auch Kapitel 4 und 5).

# Kapitel 3

# $\mathcal{PARLEX}$ - Ein LEXikalischer PARser

$\mathcal{PARLEX}$ ist ein System, welches die Beschreibung und Analyse natürlicher Sprache bez. grammatischer Strukturen erlaubt. Die Analyse wird durch einen inkrementell und verteilt arbeitenden natürlichsprachlichen Parser unterstützt. Inkrementell[1] heißt, daß der Parser zu jedem Zeitpunkt der Analyse des natürlichsprachlichen Fragments eine vorläufige grammatische Struktur mit Hypothesen für dessen Fortsetzung erzeugt. Er arbeitet verteilt, da das grammatische Wissen auf die verschiedenen lexikalischen Einträge verteilt ist.

Da der Schwerpunkt auf der Erkennung partieller Strukturen liegt, beruht die Konzeption des lexikalischen Parsers nicht auf einer Grammatik im Sinne Chomsky's, sondern auf einer Erweiterung der monadischen Logik zweiter Stufe. Büchi zeigte (siehe [Bü62], [Tho90]), daß mit dem monadischen Logikkalkül $\mathcal{L}_{M2}$ genau die $\omega$-regulären Sprachen (Mengen von Wörtern unendlicher Länge über einen Alphabet $\Sigma$) modelliert werden können. Schränkt man das Modell auf Wörter endlicher Länge ein, so erhält man die Äquivalenz zu den regulären Sprachen. Für die Beschreibung und Analyse natürlicher Sprache reichen die regulären Spra-

[1] Die Konzeption des lexikalischen Parsers basiert also auf der Annahme, daß menschliche Sprachen inkrementell analysierbar sind. Die Intuition, daß das Verstehen und die Interpretation von gesprochener bzw. geschriebener Sprache inkrementell verarbeitende Prozesse sind, wird auch von psycholinguistischen Experimenten unterstützt (siehe [MWT80, Ste83]).

chen jedoch nicht aus, so daß das Büchi-Kalkül um Zählerrelationen erweitert wurde. Durch die Erweiterung erhalten wir eine Sprachklasse, die echt kontextfreie und echt kontextsensitive Mitglieder hat. Die Sprachklasse, die somit quer zur Chomsky-Hierarchie liegt, umfaßt z.B. die Klammersprachen (eine echte Untermenge der kontextfreien Sprachen) und $\{a^n b^n c^n | a, b, c \in \Sigma\}$ (eine echte Untermenge der kontextsensitiven Sprachen). Die Erweiterung um Zählerprädikate erlaubt es z.B., geschachtelte Nebensätze zu erfassen, die in der deutschen Sprache häufig Verwendung finden.

Die generativen Grammatiken im Sinne Chomsky's folgen dem Prinzip der Konstituenz, welche die Linearität einer Sprache reflektiert: Eine Sprache wird als Konkatenation von Zeichen aufgefaßt. Die unterschiedlichen Grammatikformalismen, welche für die Analyse natürlicher Sprache entwickelt wurden, erlauben im allgemeinen entweder die Formalisierung aller kontextsensitiven Sprachen (z.B. ATN's [Wod70]) oder eine Obermenge der kontextfreien Sprachen (wie z.B. TAG's [Jos87]). Unser verwendeter Logikformalismus ist nicht nur in der Lage, Konstituenzphänomene zu beschreiben, sondern auch Dependenzphänomene. Er unterstützt daher den Grammatikenwurf für natürliche Sprachen, welche auf einer freien Wortstellung beruhen. Dies leisten im Bereich der Konstituentenstrukturgrammatiken nur sehr mächtige Grammatikformalismen (vgl. auch [KK85]).

Die Bildung des Beschreibungsmodells für grammatische Strukturen natürlicher Sprache beruht auf einer Abstraktion, welche weiter geht als sie durch andere Grammatikformalismen vorgenommen wird. Die Modellbildung wird im allgemeinen auf der Ebene der sprachlichen Größen oder sprachlichen Einheiten vorgenommen. Die Modellbildung mit der erweiterten monadischen Logik zweiter Stufe setzt auf der Ebene der Textpositionen als Grundbereich auf. Durch diese Abstraktion ist es möglich, daß das System bei unbekannten Wortformen Hypothesen über die nicht bekannte sprachliche Einheit machen kann.

Auch [Sik97] sieht einen Wechsel bei den Formalismen, in denen natürliche Sprache beschrieben wird:

> *„Secondly, the formalisms in which natural language grammars are described have changed over the last decade. This has some conse-*

*quences for parsing natural language grammars. Logic has gained an important role in the interface between grammarians and computers.“*

Einerseits existieren logische Programmiersprachen bzw. Paradigmen wie PROLOG und CLP[2], die es erlauben, Programme als Menge logischer Formeln zu schreiben. Andererseits kann auch eine Grammatik als Menge logischer Formeln beschrieben werden. Einen Satz oder einen natürlichsprachlichen Ausdruck kann man dann als Hypothese ansehen, die „korrekt“ ist, falls eine Formel mit der folgenden Interpretation gültig ist: „dies ist ein Satz und die Struktur ist so-und-so“. Damit hätte man nach [Sik97] eine Art „automatische Programmierung“, da nur noch die Grammatik in der Logik spezifiziert werden muß und ein solcher Beweis mittels eines PROLOG- oder CLP-Interpreters geführt werden könnte. Die Notwendigkeit der Konstruktion eines Parsers würde somit entfallen, aber er stellt fest [Sik97]:

*„There is a catch, however. Such specifications in logic can (under certain restrictions) be interpreted directly by machines, but that does not necessarily mean that a machine will do so in an efficient manner. From a computational point of view it is more appropriate to see such a grammar as an executable specification, not as the most suitable implementation of a parser. Computer Science, therefore, can make valuable contributions to the construction of efficient parsers for these grammar formalisms.“*

Wir glauben mit der Konzeption des lexikalischen Parsers einen solchen Beitrag leisten zu können, denn zu dem erweiterten Büchi-Kalkül $\mathcal{L}_{ZR2}$ existiert ein äquivalentes Automatenmodell - der Endliche Automat mit Zählerrelation. Der konstruktive Äquivalenzbeweis bildet die Grundlage für die Konstruktion eines *Automatengenerators*, der Hauptbestandteil der Parserkomponente ist (siehe [Sch95]). Der Automatengenerator generiert zu grammatischen Strukturen, notiert in Formel des erweiterten Büchi-Kalküls[3], erkennende Endlichen Automaten

[2] Constraint Logic Programming

[3] Die grammatischen Strukturen werden in dem System $\mathcal{PARLEX}$ in $\mathcal{L}_{ZR}$, also in Formeln 1. Stufe notiert. Die Überführung der Formeln in eine Normalform machen eine Quantifizierung 2. Stufe notwendig.

mit Zählerrelation, welche als Parserkomponente dienen.

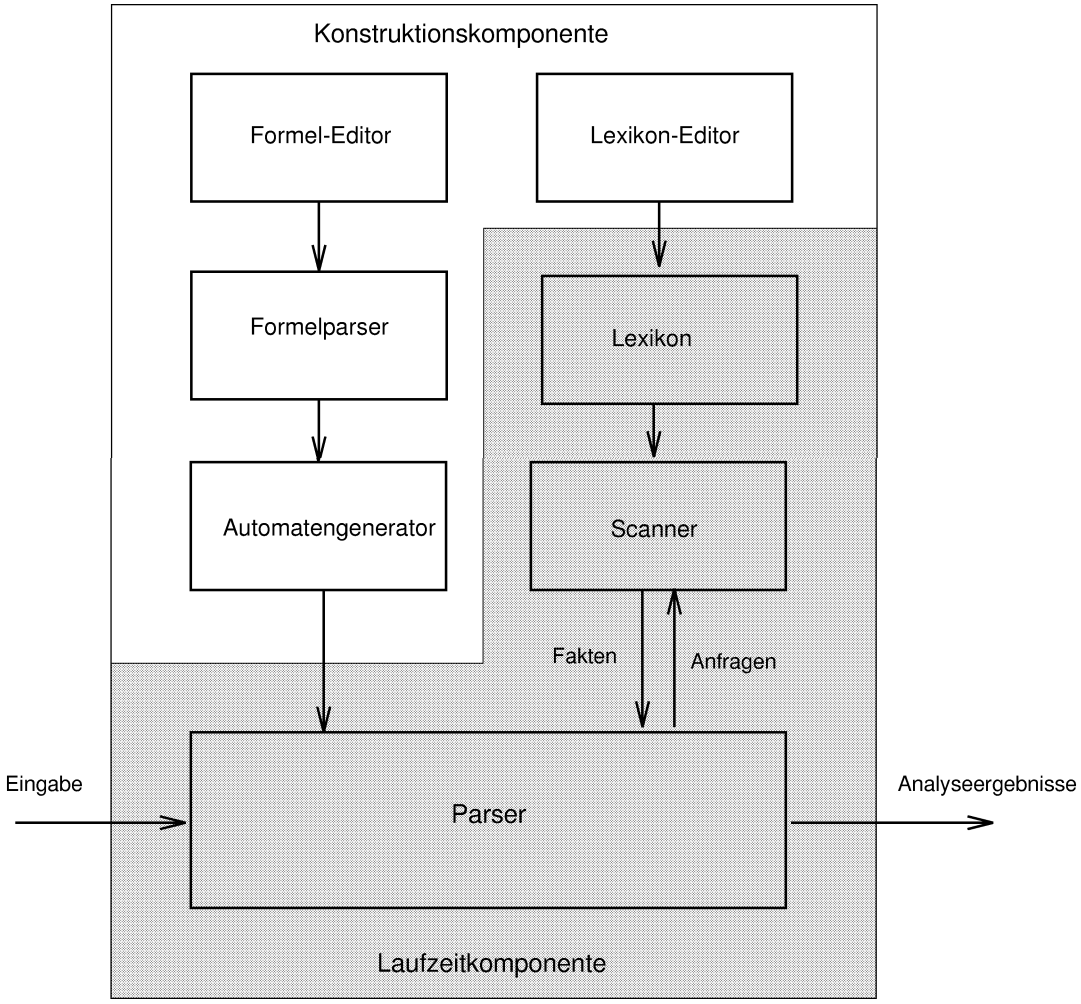


Abbildung 3.1: Architektur des Parsergenerators

Jeder Wortform eines natürlichsprachlichen Fragments wird während der Laufzeit seine kanonische Position im Fragment zugewiesen. Die lexikalische Wissensbasis enthält die morpho-syntaktischen und semantischen Informationen zu Wörtern einer natürlichen Sprache. Der Parser fordert diese Fakten zu jeder Position der Eingabe an. Die Interpretation der Automaten beruht auf der Auswertung dieser Faktenmenge.
Durch den Automatengenerator ist das System $\mathcal{PARLEX}$ in der Lage, nicht nur Modifikationen und Erweiterungen in der lexikalischen Wissensbasis zu bearbei-

ten, sondern auch im Bereich der Grammatikenwicklung sind jederzeit Änderungen und Erweiterungen durch Recompilierung zugelassen. Als Grundlage für die logischen Formeln wird auch das gespeicherte lexikalische Wissen in Form von Valenzmustern genutzt.
Im folgenden Abschnitt wird der Beschreibungsformalismus genauer betrachtet.

## 3.1 Beschreibung natürlicher Sprache mit einer erweiterten monadischen Logik zweiter Stufe $\mathcal{L}_{ZR2}$

Büchi zeigte über das äquivalente Automatenmodell, daß die $\omega$-regulären Sprachen durch geschlossene Formeln der monadischen Logik zweiter Stufe eines Nachfolgers (S1S) definierbar sind. S1S ist die Erweiterung der Logik erster Stufe, welche die Quantifizierung über monadische, also einstellige Prädikate erlaubt. Dabei sind einstellige Prädikate Erkennungsrelationen für die Zeichen eines Alphabets.

**Definition 3.1 (S1S)**
*Die Formeln der monadischen Logik zweiter Stufe eines Nachfolgers*[4] *(S1S) sind aufgebaut mit Hilfe von:*

- *n einstelligen Prädikatssymbolen* $P_1, P_2, \ldots, P_n$,
- *einem einstelligen Funktionssymbol succ,*
- *einem Konstantensymbol min,*
- *einem 2-stelligen Prädikatssymbol* $\dot{<}$,
- *den Booleschen Operatoren* $\vee, \wedge, \neg$,
- *den Quantoren erster Stufe* $\exists x, \forall x$, *wobei x für ein Element des Interpretationsbereichs steht.*

[4] Second order Logic of 1 Successor

- *den Quantoren zweiter Stufe* $\exists X, \forall X$*, wobei* $X$ *für eine Teilmenge des Interpretationsbereichs steht.*

*Als Interpretationsbereich betrachten wir die Natürlichen Zahlen* $\omega$*, wir interpretieren:*

- *min als 1,*
- *succ als natürliche Nachfolgerfunktion* $m \to m + 1$*,*
- $\dot{<}$ *als die übliche Ordnung* $<$ *auf* $\omega$

Der Beweis von Büchi zeigt die Äquivalenz der monadischen Logik zweiter Stufe zum endlichen Automatenmodell. Dabei entspricht das Automatenmodell dem Modell für Wörter endlicher Länge, lediglich die Akzeptanzbedingung ist modifiziert:

**Definition 3.2 (Büchi-Automat)**
*Ein Büchi-Automat ist ein endlicher Automat* $\mathcal{A} = (Q, \Sigma, I, \Delta, F)$*, mit:*

- $Q = \{q_0, \ldots, q_n\}$ *ist eine nicht-leere, endliche Menge von Zuständen,*
- $\Sigma$ *ist ein endliches Alphabet,*
- $I \subseteq Q$ *ist eine Menge ausgezeichneter Anfangszustände,*
- $\Delta \subseteq Q \times \Sigma \times Q$ *ist die Übergangsrelation,*
- $F \subseteq Q$ *ist die Menge der Endzustände.*
- *Die Akzeptanzbedingung für* $\mathcal{A}$ *lautet: Ein unendlicher Pfad heißt erfolgreich, wenn*
  1. $q_0 \in I$ *und*
  2. *es unendlich viele i mit* $q_i \in F$ *gibt.*

Um die von Büchi-Automaten akzeptierten Sprachen genauer charakterisieren zu können, führen wir noch folgende Schreibweise ein; sei $\mathcal{A}$ ein Büchi-Automat mit $p, q \in Q$, dann ist

$$L_{p,q} := \{\omega \in \Sigma^* | \omega \text{ ist die Beschriftung eines endlichen Pfades von q nach p}\}$$

Diese Sprachen sind offenbar regulär.

**Satz 3.1**
*Es gilt* $L_\omega(\mathcal{A}) = \bigcup_{i \in I, j \in F} L_{i,j} \cdot L_{j,j}^\omega$.[5]
*Wir bezeichnen die von Büchi-Automaten akzeptierten Sprachen daher auch als $\omega$-reguläre Sprachen.*

**Beispiel 3.1**
*Sei $\Sigma = \{a, b\}$. Der folgende nicht-deterministische Büchi-Automat $\mathcal{A}$ akzeptiert die Sprache $(a \cup b)^* b^\omega$.*

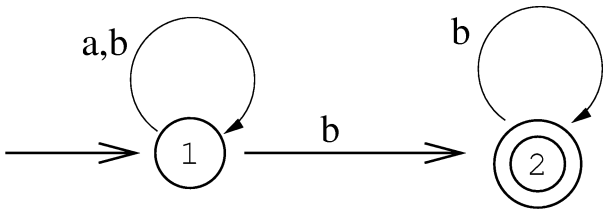


Abbildung 3.2: Beispiel eines nicht-deterministischen Büchi-Automaten

$$L_{1,2} = (a \cup b)^* \cdot b \cdot L_{2,2} \qquad L_{2,2} = b^*$$

$$\begin{aligned} L_\omega(\mathcal{A}) &= L_{1,2} \cdot L_{2,2}^\omega \\ &= (a \cup b)^* \cdot b \cdot b^* \cdot (b^*)^\omega \\ &= (a \cup b)^* \cdot b^\omega \end{aligned}$$

[5]Wir bezeichnen mit $L^\omega$ die unendliche Iteration: Sei $L \subseteq \Sigma^*$ eine Menge endlicher Wörter, dann ist $L^\omega := \{\alpha \in \Sigma^\omega | \alpha = \omega_0, \omega_1, \ldots \text{mit } \omega_i \in L \backslash \{\epsilon\}\}$

An diesem einfachen Beispiel läßt sich zeigen, daß im Gegensatz zu Automaten auf endlicher Wortlänge **nicht** gilt, daß jeder nicht-deterministische Automat durch *Potenzmengenkonstruktion* in einen endlichen deterministischen Automaten überführt werden kann. Denn, angenommen $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ wäre ein deterministischer[6] endlicher Automat mit $L_\omega(\mathcal{A}) = L$. Da $ab^\omega \in L$ gibt es ein $k_1 > 0$ und $f_1 \in F$ mit $q_0 \xrightarrow{ab^{k_1}}_{\mathcal{A}} f_1$. Nun ist aber auch $ab^{k_1}ab^\omega \in L$ und $\mathcal{A}$ deterministisch. Also gibt es ein $k_2 > 0$ und $f_2 \in F$ mit $q_0 \xrightarrow{ab^{k_1}}_{\mathcal{A}} f_1 \xrightarrow{ab^{k_2}}_{\mathcal{A}} f_2$. Iteriert man diesen Prozeß, so erhält man, daß $\alpha = ab^{k_1}ab^{k_2}ab^{k_3}\cdots \in L_\omega(\mathcal{A})$ ist. Da $\alpha \notin L$ gilt, ist dies ein Widerspruch zu $L_\omega(\mathcal{A}) = L$.

**Satz 3.2**
*Eine formale Sprache L ist $\omega$-regulär genau dann, wenn L durch eine geschlossene S1S-Formel definierbar ist.*

Die Beweisrichtung „Für jede Formel gibt es einen endlichen Automaten“ wird konstruktiv über den Aufbau der Formeln in zweistufiger Normalform geführt. Außerdem muß noch gezeigt werden, daß jede beliebige Formel aus S1S auf die Normalform reduzierbar ist.

**Definition 3.3 (Zweistufige Normalform)**
*Formeln, die aus den folgenden Symbolen und atomaren Formeln mit Hilfe Boolescher Operationen und Quantifizierung zweiter Stufe aufgebaut sind, heißen Formeln zweiter Normalform $S1S_0$:*

1. *Es dürfen nur noch Variablen $X_i$ zweiter Stufe vorkommen.*

2. *Atomare Formeln haben die Gestalt:*

   - *$X_i \dot{\subseteq} X_j$ als Abkürzung für $\forall x(X_i(x) \Rightarrow X_j(x))$,*
   - *$Succ(X_i) \doteq X_j$ mit Semantik $X_j^I = \{n_j\}, X_i^I = \{n_i\}$ und $n_i + 1 = n_j$. Dabei sind $X_i$ und $X_j$ Variablen zweiter Stufe oder Prädikatssymbole $P_1, \ldots, P_n$.*

[6] D.h. $\delta$ ist eine Funktion: $\delta : Q \times \Sigma \to Q$.

Betrachten wir nun, wie die Formeln aus S1S auf die Normalform reduziert werden können.

Man kann auf die Symbole $\dot{<}$ und min verzichten, da diese durch die anderen Symbole darstellbar sind:

1. x = min ist äquivalent zu $\neg\exists y(y\dot{<}x)$.
2. $x\dot{<}y$ ist äquivalent zu $\forall X(X(succ(x)) \wedge \forall z(X(z) \Rightarrow X(succ(z))) \Rightarrow X(y))$

Geschachtelte Anwendungen der Nachfolgerfunktion können wie folgt aufgelöst werden:

$$y\dot{=}\underbrace{succ(succ(\ldots succ(x)\ldots))}_{m-mal}$$

ist äquivalent zu

$$\exists y_1 \ldots \exists y_{m-1}(y_1\dot{=}succ(x) \wedge y_2\dot{=}succ(y_1) \wedge \cdots \wedge y_{m-2}\dot{=}succ(y_{m-1} \wedge succ(y_{m-1})\dot{=}y)$$

Wir haben damit nur noch atomare Formeln der Form $x\dot{=}y$, $succ(x)\dot{=}y$, $P_i(x)$ und $X(x)$ zu betrachten. Wir führen noch folgende Abkürzungen ein:

- $X\dot{\equiv}Y$ für $X\dot{\subseteq}Y \wedge Y\dot{\subseteq}X$,
- $X \not\equiv Y$ für $\neg(X \doteq Y)$,
- $Sing(X)$ für $\exists Y(Y\dot{\subseteq}X \wedge Y \not\equiv X \wedge \forall Z(Z\dot{\subseteq}X \Rightarrow (Z\dot{\equiv}X \vee Z\dot{\equiv}Y)))$, d.h. X hat genau eine echte Teilmenge, also ist X eine Einermenge.

Die Quantifizierung über Positionsvariablen ist nun möglich:

- $\exists x\varphi(x)$ steht für $\exists X(Sing(X) \vee \varphi(X))$
- $\forall x\varphi(x)$ wird ersetzt durch $\forall X(Sing(X) \Rightarrow \varphi(x))$

Damit wurde gezeigt, daß jede Formel aus S1S auf eine Formel aus $S1S_0$ reduzierbar ist. Die Konstruktion des zu einer Formel $\varphi$ äquivalenten Automaten erfolgt induktiv über den Aufbau der Formeln. Zunächst konstruieren wir Automaten für

die atomaren Formeln. Da wir auch freie Variablen in einer Formel zulassen wollen, gibt die „Kantenbeschriftung“ der Automaten zusätzlich zu der Bedingung, welches Zeichen erkannt wird, Bedingungen für die Belegung der freien Variablen an. Sei also $\varphi$ eine Formel mit freien Variablen $X_1, \ldots, X_n$. So steht $X_i$ für die Bedingung, daß die aktuelle Position Element von der Interpretation von $X_i$[7] ist; $\sim X_i$ steht für die Bedingung, daß die aktuelle Postion nicht Element der Interpretation von $X_i$ sein darf ($*X_i$ steht für $X_i$ oder $\sim X_i$). Betrachten wir nun die „atomaren“ Automaten:

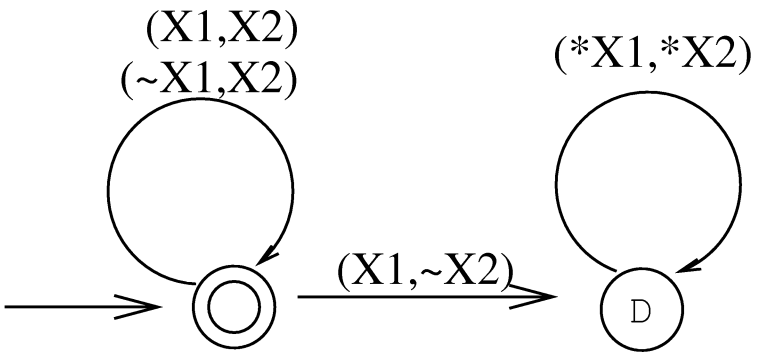


Abbildung 3.3: $A(X_1, X_2)$: $X_1 \subseteq X_2$

Der Automat erwartet Positionen, welche, falls sie Element von der Menge $X_1^I$ sind, auch Element der Menge $X_2^I$ sind.
Wir fügen der Kantenbeschriftung noch eine weitere Stelle für die Erkennungsrelationen $P_i$ hinzu: $P_i$ steht für die Bedingung, daß $P_i$ auf die aktuelle Position zutrifft, $\sim P_i$ für die Bedingung, daß $P_i$ nicht zutrifft und * steht für $P_1$ oder ... oder $P_n$.
Der folgende Automat erwartet Positionen, welche, falls sie Element von der Menge $X_1^I$ sind, auch Element der Menge $P^I$ sind.

[7] Im folgenden mit $X_i^I$ bezeichnet.

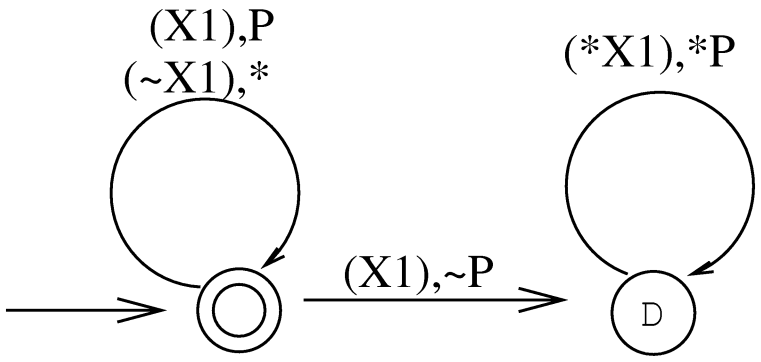


Abbildung 3.4: $A(X_1)$: $X_1 \subseteq P$

Der Automat für die Succ-Funktion erwartet ein Element der Menge $X_1^I$. Sobald das Element erkannt wird, erwartet er ein Element der Menge $X_2^I$. Die Elemente müssen an zwei aufeinanderfolgenden Positionen erkannt werden und sie müssen verschieden sein. Der Automat geht ebenfalls nicht in einen Endzustand, falls es sich bei $X_1$ oder $X_2$ nicht um einelementige Mengen handelt.

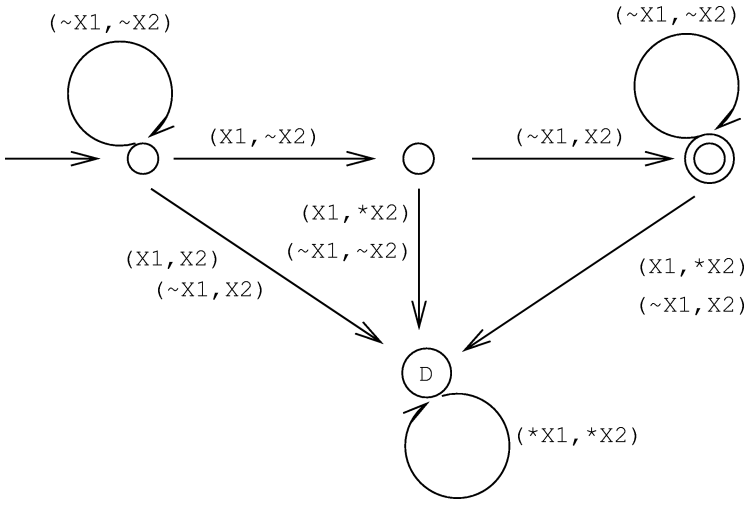


Abbildung 3.5: $A(X_1, X_2)$: $Succ(X_1) = X_2$

Beim Induktionsschritt genügt es, $\vee$, $\exists X$ und $\neg$ zu betrachten:

1. „$\vee$“ entspricht der Vereinigung, wobei man beachten muß, daß bei einer Formel $\phi = \phi_1 \vee \phi_2$ die in $\phi_1$ und $\phi_2$ vorkommenden freien Variablen oder

Prädikatssymbole unterschiedlich sein können. Man erweitert $\phi_1$ und $\phi_2$ um die fehlenden freien Variablen oder Prädikate, indem man einfach wahre Formeln anfügt, und erhält so $\bar{\phi}_1$ und $\bar{\phi}_2$. Die disjunkte Vereinigung der beiden erweiterten Automaten akzeptiert dann die Sprache $L_\omega(\bar{\phi}_1 \vee \bar{\phi}_2)$.

2. Es sei $\phi(X_1, \ldots, X_n) = \exists Y(\varphi(Y, X_1, \ldots, X_n))$.
   Ist $\mathcal{A}$ ein Automat für $L_\omega(\varphi(Y, X_1, \ldots, X_n))$, so erhält man durch Streichung der Transitionsbedingungen, welche die Belegung der Variablen Y charakterisieren, einen Automaten für $\phi(X_1, \ldots, X_n)$.

3. Da die $\omega$-regulären Sprachen unter Komplement abgeschlossen sind, ist die Negation ebenfalls gezeigt.

Die umgekehrte Beweisrichtung zeigen wir, indem wir die Existenz eines „erfolgreichen“ Pfades durch eine S1S-Formel beschreiben. Sei $A = (Q, \Sigma, I, \Delta, F)$ ein Büchi-Automat mit $L = L_\omega(A)$. Es sei weiterhin $Q = \{q_0, \ldots, q_m\}$ und ohne Einschränkung, die Menge der Anfangszustände $I = \{q_0, \ldots, q_k\}$ für ein $k \leq m$. Wir verwenden Variablen $X_0, \ldots, X_m$, wobei $X_i(x)$ als „der x-te Zustand im Pfad ist $q_i$“ gelesen werden soll[8].

$$
\begin{aligned}
&\exists X_0 \ldots \exists X_m (\forall x (\bigwedge_{0 \leq i < j \leq m} \neg(X_i(x) \wedge X_j(x)))) \wedge \\
&(X_o(min) \vee \cdots \vee X_m(min) \wedge \\
&(\forall x (\bigvee_{(q_i, a, q_j) \in \Delta} X_i(x) \wedge Q_a(x) \wedge X_j(succ(x)))) \wedge \\
&(\bigvee_{q_i \in F} \forall x \exists y (x \dot{<} y \wedge X_i(y)))
\end{aligned}
$$

Wir betrachten also disjunkte Mengen und beginnen mit einem Anfangszustand. An der Position $x$ steht ein bezüglich $\Delta$ erlaubter Folgezustand. Einer der Endzustände wird unendlich oft angenommen. Nach Konstruktion erfüllt $\alpha \in \Sigma^\omega$ diese Formel genau dann, wenn $\alpha$ die Beschriftung eines „erfolgreichen“ Pfades

[8] $Q_a$ steht für eine Erkennungsrelation.

ist.

Der skizzierte Beweis zeigt, daß man zu jeder S1S-Formel $\phi$ effektiv einen Büchi-Automaten konstruieren kann mit $L_\omega(A) = L_\omega(\phi)$. Damit folgt, daß man von einer geschlossenen S1S-Formel $\phi$ entscheiden kann, ob sie in der kanonischen Interpretation ($\omega$,<,succ,min) gültig ist oder nicht.

### 3.1.1 Die erweiterte monadische Logik zweiter Stufe $\mathcal{L}_{ZR2}$

Die Formeln sollen der grammatischen Beschreibung natürlicher Sprache dienen, d.h. sie sollen bez. Sätzen oder Texten interpretiert werden. Hierzu wird jedem Wort in einem natürlichsprachlichen Text seine Position zugeordnet. Das Alphabet der zu beschreibenden Sprache umfaßt die Wörter der Sprache. Die Anpassung des Konzeptes erfolgt in mehreren Schritten:

Zuerst wird das Modell auf Wörter endlicher Länge eingeschränkt, $max$ wird für die maximale Position in endlichen Wortmodellen als Positionskonstante eingeführt. Sei $\Sigma = \{a_1, \ldots, a_n\}$ ein endliches Alphabet und $\Sigma^+$ die Menge der Wörter über $\Sigma$ mit endlicher Länge. Die Länge eines Wortes $\omega$ wird mit $|\omega|$ bezeichnet. Bezüglich der Symbolmenge $\{P_1, \ldots, P_n, \dot{\subseteq}, Succ\}$ wird das Wortmodell $\mathcal{M}_\omega := (\omega, P_1^\omega, \ldots, P_n^\omega, \dot{\subseteq}^\omega, Succ^\omega)$ definiert durch die Menge der Positionen des Wortes $\omega$ und die Mengen $P_1^\omega, \ldots, P_n^\omega$. Dabei ist $P_i^\omega$ die Menge der Wortpositionen $\omega_k$ für die $\omega_k = a_i$ gilt. „$\dot{\subseteq}$“ wird auf die Teilmengenbeziehung bez. der Potenzmenge der Menge der Wortpostionen abgebildet. $Succ$ wird auf die Funktion abgebildet, die zu einer einelementigen Teilmenge von $\mathcal{M}_\omega$ die Nachfolgerfunktion bestimmt: Ist die Wortpostion $\omega_i$ und $i < |\omega|$, das einzige Element der gegebenen Menge, so ist die nachfolgende Position $\omega_{i+1}$ Element der Nachfolgermenge. Bei der Wortgrenze, d.h. die Wortposition ist $\omega_i$ mit $i = |\omega|$, wird $\omega_i$ als Element der Nachfolgermenge gesetzt. $\omega_i$ ist in diesem Fall also die maximale Position des Wortes $\omega$, dieses Element wird mit $max$ bezeichnet. Um die Äquivalenz, die zwischen deklarativer und operationaler Semantik für Wörter unendlicher Länge gegeben ist, auch für endliche Wortmodelle zu erhalten, muß für die Funktion $Succ$ der Automat geändert werden.

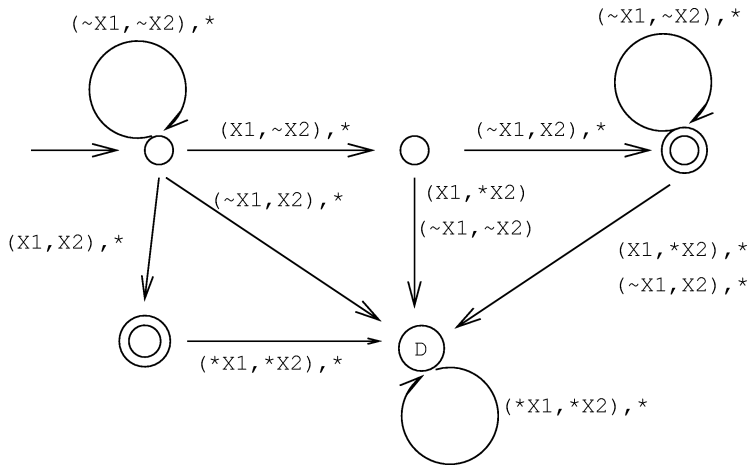


Abbildung 3.6: Modifizierter *Succ*-Automat für endliche Wortmodelle

Der angegebene Automat erreicht für den Fall, daß eine Position Element von $X_1$ und $X_2$ ist, einen Endzustand. Falls dies jedoch nicht die maximale Position ist, verläßt er diesen Endzustand wieder.
Durch die Einschränkung auf endliche Wortmodelle erhält man die Klasse der regulären Sprachen.
Die Wörter einer natürlichen Sprache sind in Wortklassen eingeteilt. Um grammatische Strukturen durch Formeln beschreiben zu können, wird für jede Wortklasse und für jede morpho-syntaktische und semantische Eigenschaft eine neue Erkennungsrelation eingeführt. Diese *Lexikalischen Erkennungsrelationen* werden mit $L_1, \ldots, L_n$ bezeichnet. Im Gegensatz zu den natürlichen Sprachen hatten wir bei den formalen Sprachen durch die Erkennungsrelationen über dem Träger eines Wortmodells eine echte Partition gegeben. Durch die Einführung der lexikalischen Erkennungsrelationen können nun auf einer Position mehrere Erkennungsrelationen zutreffen. Im Sinne der deklarativen Semantik wird $L_i$ auf das lexikalische Prädikat $L_i^\omega$ abgebildet. $L_i^\omega$ trifft auf die k-te Position zu, falls $\omega_k$ ein $a_j$ ist, mit $a_j \in \Sigma_i \subset \Sigma$. Entsprechend werden im äquivalenten Automatenmodell Transitionen zugelassen, die bezüglich der Erkennungsrelationen mehrere Bedingungen enthalten.
Außerdem wird das Kalkül um Zählerrelationen $Z_{i,j}$ und $Z_{i=j}$ erweitert. Diese erlauben z.B. den Vergleich der Auftrittshäufigkeiten zweier Wortformen. Darüber

hinaus ist auch der Vergleich der Auftrittshäufigkeit von Repräsentanten verschiedener Wortklassen zugelassen. Durch diese Erweiterungen ist die Formalisierung von Nebensatzstrukturen und die Beschreibung von diskontinuierlichen Phänomenen möglich. Die Erweiterung um Zählerrelationen führt aus der Klasse der regulären Sprachen heraus, es sind nun echt kontextfreie und kontextsensitive Mitglieder enthalten.
Semantisch werden die Symbole der Zählerrelationen auf die Zählerprädikate ${Z_{i,j}}^{\omega}$ und ${Z_{i=j}}^{\omega}$ abgebildet. Die Indizes $i$, $j$ entsprechen den Indizes einer Erkennungsrelation oder eines lexikalischen Prädikats. ${Z_{i,j}}^{\omega}$ trifft auf die k-te Position zu, falls die Anzahl der Positionen in $\omega_1, \ldots, \omega_k$ auf die $P_i^{\omega}$ bzw. $L_i^{\omega}$ zutrifft, kleiner ist als die Anzahl der Positionen in $\omega_1, \ldots, \omega_k$ auf die $P_j^{\omega}$ bzw. $L_j^{\omega}$ zutrifft. Bei ${Z_{i=j}}^{\omega}$ muß entsprechend die Anzahl gleich sein.

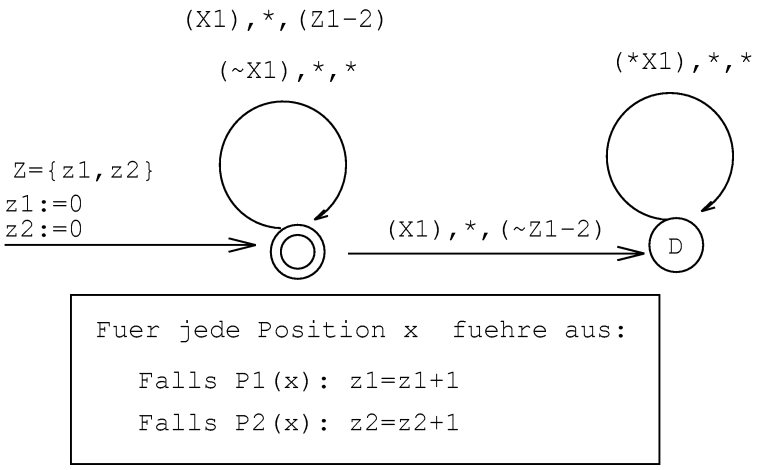


Abbildung 3.7: $A(X_1)$: $X_1 \subseteq Z_{1,2}$

Im äquivalenten Automatenmodell lassen wir nun Transitionen zu, die bez. der Zählerrelationen Bedingungen enthalten. Das Automatenmodell muß noch durch Zähler erweitert werden, damit die Bedingungen geprüft werden können.
Die Kantenbeschriftung (siehe Abbildung 3.7) nennt an der dritten Stelle Bedingungen für die Zählerrelationen. Jeder Index des Zählerprädikats referiert auf einen Zähler für eine Erkennungsrelation bzw. für ein lexikalisches Prädikat.
In der folgenden Definition sind noch einmal die Erweiterungen zusammengefaßt.

#### 3.1.1.1 Die Logik

**Definition 3.4** ($\mathcal{L}_{ZR2}$)
*Die erweiterte monadische Logik zweiter Stufe ist aufgebaut aus:*

- *den Mengenvariablen $X_1, \ldots, X_n$ und den Individuenvariablen $x_1, \ldots, x_n$;*
- *den Zeichen für Negation $\neg$, Konjunktion $\wedge$, Disjunktion $\vee$, Implikation $\rightarrow$ und Äquijunktion $\leftrightarrow$;*
- *den Zeichen für Generalisierung $\forall$ und Partikularisierung $\exists$;*
- *dem Zeichen für Identität $=$;*
- *den zweistelligen Relationssymbolen $\subseteq, \in, <$;*
- *den einstelligen Relationssymbolen $P_1, \ldots, P_n$; $L_{n+1}, \ldots, L_{n+m}$, $Z_{i,j}$ und $Z_{i=j}$;*
- *den einstelligen Funktionssymbolen $succ$ und $prec$ und*
- *den Konstanten $min$ und $max$.*

*$\Sigma$ sei ein endliches Alphabet $\{a_1, \ldots, a_n\}$ und $\omega$ ein Wort der Sprache $L \subset \Sigma^+$. Die Länge von $\omega$ sei $|\omega|$ und $\omega_i$ sei das i-te Zeichen in $\omega$. Das Wortmodell $\mathcal{M}_\omega$ wird bezüglich der Symbolmenge wie folgt definiert:*

1. *Der Träger des Grundbereiches wird mit $M_\omega$ bezeichnet. $M_\omega = \{1, \ldots, |\omega|\}$ ist die Menge der Positionen in $\omega$.*
2. *Die Symbole werden folgendermaßen interpretiert:*
   - *$P_i$ wird auf die Erkennungsrelation $P_i^\omega$ bez. $a_i \in \Sigma$ abgebildet: $P_i^\omega$ trifft auf die k-te Position zu, falls $\omega_k = a_i$ ist.*
   - *$L_i$ wird auf das lexikalische Prädikat $L_i^\omega$ abgebildet: $L_i^\omega$ trifft auf die k-te Position zu, falls $\omega_k$ ein $a_j$ ist, mit $a_j \in \Sigma_i$ und $\Sigma_i$ eine Teilmenge von $\Sigma$ ist.*

- *$Z_{i,j}$ wird auf das Zählerprädikat $Z^{\omega}_{i,j}$ abgebildet. Die Indizes i,j entsprechen den Indizes einer Erkennungsrelation oder eines lexikalischen Prädikats. $Z^{\omega}_{i,j}$ trifft auf die k-te Position zu, falls die Anzahl der Positionen in $\omega_1, \ldots, \omega_k$ auf die $P^{\omega}_i$ bzw. $L^{\omega}_i$ zutrifft, kleiner ist als die Anzahl der Positionen, auf die $P^{\omega}_j$ bzw. $L^{\omega}_j$ zutrifft. $Z_{i=j}$ wird auf das Zählerprädikat $Z^{\omega}_{i=j}$ abgebildet und entsprechend muß die Anzahl der Positionen gleich sein.*

- *$min$ wird auf die kleinste Position $min^{\omega}$ in $M_{\omega}$ abgebildet.*

- *$max$ entsprechend auf die größte Position: $max^{\omega} := |\omega|$.*

- *succ wird auf die einstellige Nachfolgerfunktion $succ^{\omega}$ abgebildet:*

$$succ^{\omega} = \begin{cases} k & \text{für } k = max^{\omega} \\ k+1 & \text{sonst} \end{cases}$$

- *prec wird auf die einstellige Vorgängerfunktion $prec^{\omega}$ abgebildet:*

$$prec^{\omega} = \begin{cases} k & \text{für } k = min^{\omega} \\ k-1 & \text{sonst} \end{cases}$$

- *$<$ wird auf die natürliche Ordnungsrelation, eingeschränkt auf $M_{\omega}$, abgebildet.*

- *$\in$ wird auf die Elementbeziehung bez. der Positionsmenge $M_{\omega}$ abgebildet.*

- *$\subset$ wird auf die Teilmengenbeziehung bez. der Potenzmenge der Positionsmengen abgebildet.*

*Die Belegung $\beta$ in einem Wortmodell bildet die Menge der Variablen $x_1, \ldots, x_n$ in die Menge der Positionen $M_{\omega}$ ab. Die Menge der Variablen $X_1, \ldots, X_n$ wird in die Potenzmenge von $M_{\omega}$ abgebildet. Die Interpretation der Variablen ist durch ihre Belegung $\beta(x)$ gegeben.*

#### 3.1.1.2 Das Automatenmodell

**Definition 3.5**

*Ein deterministischer Automat über den Parametern $X_1, \ldots, X_m$ wird mit $A(X_1, \ldots, X_m)$ bezeichnet. Er ist für Wortmodelle $\mathcal{M}_w$ bezüglich einer Symbolmenge*

$$\{P_1, \ldots, P_n, L_{n+1} \ldots L_{n+m}, <, min, max, succ, prec\} \cup$$
$$\{Z_{i,j} \mid i \neq j \ , \ 1 \leq i, j \leq n+m\} \ \cup \ \{Z_{=i,j} \mid 1 \leq i < j \leq n+m\}$$

*durch ein Tupel*

$$A = (Q, Z, \delta, q_0, F)$$

*definiert, mit:*

- *$Q$ ist nicht leere, endliche Zustandsmenge*
- *$q_0 \in Q$ ist Anfangszustand*
- *$Z = (z_1, \ldots, z_{n+m})$ ist Tupel der Zähler, wobei $i = 1, \ldots, n+m$ und $z_i \in I\!N$*

  *$Z_0$ ist ausgezeichnetes Tupel von Zählern: $Z_0 = (0, \ldots, 0)$*
- *$F \subset Q$ ist nicht leere Menge der Endzustände*
- *Die Übergangsfunktion $\delta$ bildet einen Zustand, einen Tupelzähler und eine Position in einen Zustand und einen Tupelzähler ab.*

  $\delta : Q \times Z \times \{1, \ldots, |w|\} \rightarrow Q \times Z$

  $\delta(q, Z, x) := \delta_w(q, \delta_z(Z, x))$, *wobei:*

  - $\delta_z : Z \times \{1, \ldots, |w|\} \rightarrow Z \times \{1, \ldots, |w|\}$

    $\delta_z(Z, x) = (Z', x)$ *mit* $Z' = (z'_1, \ldots, z'_{n+m})$, *für alle* $1 \leq i \leq n+m$ *gilt:*

    - $z'_i = z_i + 1$, *falls $i$ ist Index der Erkennungsrelation $P_i^w$ und $P_i^w$ trifft auf $\beta(x)$ zu*
    - $z'_i = z_i + 1$, *falls $i$ ist Index des lexikalischen Prädikats $L_i^w$ und $L_i^w$ trifft auf $\beta(x)$ zu*
    - $z'_i = z_i$ *sonst*
  - $\delta_w : Q \times Z \times \{1, \ldots, |w|\} \rightarrow Q \times Z$

$\delta_w(q, Z, x) = (q', Z)$ *falls* $\beta(x)$ *die Transitionsbedingung erfüllt.*

$\delta$ *wird fortgesetzt zu:* $\delta : Q \times Z \times \{M_w\} \to Q \times Z$

*durch:*

– $\delta(q_0, Z_0, M_w) = \delta_w(\ldots(\delta_w(q_0, \delta_z(Z_0, 1)), 2, \ldots, |w|))$

*Gegeben ist ein endlicher deterministischer Automat* $A(X_1, \ldots, X_m)$ *und eine Belegung der Parameter* $X_1, \ldots, X_m$.

- *Die Folge von Transitionen, die durch* $\delta(q_0, Z_0, M_w)$ *gegeben ist, ist eine Interpretation des Automaten.*
- *Ist mit* $\delta(q_0, Z_0, M_w)$ *ein Endzustand gegeben, so akzeptiert der* $Büchi_{ZR2}$*-Automat das Wort* $w$.

### 3.1.2 Die Grammatikentwicklung mit $\mathcal{L}_{ZR2}$

Eine grammatische Struktur L einer natürlichen Sprache wird als formale Sprache über einem durch die lexikalische Wissensbasis gegebenen Alphabet aufgefaßt, indem eine $\mathcal{L}_{ZR2}$-Charakterisierung $\phi$ angegeben wird. Es werden Bezeichner für ein- oder mehrstellige $\mathcal{L}_{ZR2}$-Prädikate und das Symbol $:\Leftrightarrow$ für die Definition eines $\mathcal{L}_{ZR2}$-definierbaren Prädikats benötigt. Die Stelligkeit des Prädikatssymbols ist dabei gleich der Anzahl der freien Variablen. Diese sogenannte *Wissensformel* definiert eine grammatische Struktur L. Die Analyse eines natürlichsprachlichen Fragments $\omega$ bezüglich der grammatischen Struktur L beruht auf der These $L = L(\phi)$. Die Analyse kann also durch Prüfen der Modellbeziehung mit einem zu $\phi$ äquivalenten Automaten geschehen.

**Beispiel 3.2 (Wissensformel - Nebensatz)**
*Es seien* $NSeinl$ *und* $finVerb$ *zwei lexikalische Prädikate.* $Z_{1=2}$ *ist das entsprechende Zählerprädikat für* $NSeinl$ *und* $finVerb$*, dann ist die grammatische Struktur „Nebensatz“ definiert durch:*
$Nebensatz(x_1, x_2) :\Leftrightarrow x_1 < x_2 \wedge NSEinl(x_1) \wedge finVerb(x_2) \wedge Z_{1=2}(x_2)$

Eine *Erwartungsformel* hat die Form $\alpha \Rightarrow \beta$. Beim Eintreffen der Vorbedingung $\alpha$ wird die Nachbedingung $\beta$ mit einem zu $\beta$ äquivalenten Automaten überprüft.

**Beispiel 3.3 (Erwartungsformel - Nebensatzeinleitung)**
$NSEinl(x_1) \Rightarrow \exists x_2 \in [succ(x_1), max] \wedge Nebensatz(x_1, x_2)$[9]

Abhängig von der Art der Vorbedingungen, sprechen wir von *lexikalischen Erwartungen*, *grammatischen Erwartungen* oder *globalen Erwartungen* (Vorbedingung TRUE). Bei den lexikalischen Erwartungen ist die Vorbedingung eine atomare Formel, die über ein lexikalisches Prädikat oder eine Erkennungsrelation und über eine Variable gebildet wird. Bei grammatischen Erwartungen ist die Vorbedingung durch ein Relationssymbol für ein benutzerdefiniertes Prädikat und der Stelligkeit entsprechend vieler Variablen gekennzeichnet.
Die lexikalischen Erwartungen können zur Formalisierung von Argumentstrukturen im Sinne der Valenztheorie[10] verwendet werden. Betrachtet man die Klasse der Verben, so lassen sich im Deutschen etwa 20 verschiedene Valenzmuster unterscheiden und damit die Verben subklassifizieren. Da in unserem Ansatz die Valenzbeschreibung nicht auf die syntaktische Spezifikation beschränkt ist, sondern auch semantische Merkmale und Rollen zur Beschreibung einbezogen werden, kann die Zahl der Valenzbeschreibungen beliebig wachsen.

**Beispiel 3.4 (Beschreibung von Argumentstrukturen)**
*Wir betrachten eine Teilmenge der Klasse der einstelligen Verben:*
$V_1$ :={*arbeit,atm,brenn,frier,kling,...*}
*und eine Teilmenge der zweistelligen Verben mit präpositionaler Ergänzung:*
$V_{präp}$:={*brenn auf,denk an, fahr nach/zu, geh nach/zu,...*}.

- *Betrachten wir die Argumentstrukturen als rein syntaktische Phänomene:*

  1. *die Klasse $V_1$ der Verben kann durch folgende Erwartungsformel be-*

[9] Die Intervallschreibweise ist lediglich eine Abkürzung, es wird $\mathcal{L}_{ZR2}$ nicht verlassen. Denn:
$(\exists x_1 \in [x_1, x_2]\varphi(x_1)) \leftrightarrow (\exists x_1((x_2 \leq x_1 \leq x_3) \wedge \varphi(x_1)))$ und
$(\forall x_1 \in [x_1, x_2]\varphi(x_1)) \leftrightarrow (\forall x_1((x_2 \leq x_1 \leq x_3) \rightarrow \varphi(x_1)))$

[10] Eine Beschreibung der verwendeten Valenztheorie findet die Leserin in 5.1.2

*schrieben werden:*

$$\begin{aligned} L_{V_1}(x) \Rightarrow & \exists x_1 \exists x_2 \in [min, max](ERG_{NOM}(x_1, x_2) \wedge \\ & \neg\exists x_3 \neg\exists x_4 \in [min, max](x_1 \neq x_3 \wedge x_2 \neq x_4 \wedge \\ & ERG_{NOM}(x_3, x_4))) \end{aligned}$$

*$ERG_{NOM}$ ist eine grammatische Struktur „nominale Ergänzung im Nominativ“, deren Konstituenz im weiteren Grammatikentwurf definiert werden müßte, z.B. Charakterisierung als „Nominalphrase“.*

2. *Betrachten wir nun die zweite Subklasse der Verben:*

$$\begin{aligned} L_{V_{präp}}(x) \Rightarrow & (\exists x_1 \exists x_2 \in [min, max](ERG_{NOM}(x_1, x_2) \wedge \\ & \neg\exists x_3 \neg\exists x_4 \in [min, max](x_1 \neq x_3 \wedge x_2 \neq x_4 \wedge \\ & ERG_{NOM}(x_3, x_4))) \wedge \\ & (\exists x_5 \in [succ(x_2), max](Praep(x_5) \wedge \\ & \exists x_6 \exists x_7 \in [succ(x_5), max](ERG_{AKK}(x_6, x_7) \vee \\ & ERG_{dat}(x_6, x_7) \vee ERG_{gen}(x_6, x_7))))) \end{aligned}$$

- *Auf der semantischen Ebene führen wir nun weitere Restriktionen in Form von semantischen Prädikaten ein, diese Subklassifizierung kann in Einzelfällen bis zur Beschreibung einzelner Verben, also einelementigen Teilmengen, führen.*

  1. *Betrachten wir das einstellige Verb „brennen“, so läßt sich feststellen, daß es sinnvoll ist, den semantischen Bereich auf Konkreta einzuschränken*[11]*. Wir führen also ein semantisches Prädikat $ERG_{konkret}$ ein und erhalten für das Verb:*

$$\begin{aligned} L_{brenn}(x) \Rightarrow & \exists x_1 \exists x_2 \in [min, max] \\ & ERG_{konkret}(x_1, x_2) \wedge (ERG_{NOM}(x_1, x_2) \wedge \\ & \neg\exists x_3 \neg\exists x_4 \in [min, max](x_1 \neq x_3 \wedge x_2 \neq x_4 \wedge \\ & ERG_{NOM}(x_3, x_4))) \end{aligned}$$

[11] Einen metaphorischen Gebrauch schließen wir an dieser Stelle aus.

*Das semantische Prädikat* $ERG_{konkret}$ *ist wie folgt definiert:* $ERG_{konkret}(x_1, x_2) \Leftrightarrow \exists x \in [x_1, x_2] konkret(x)$.

2. *Betrachten wir nun das zweistellige Verb „brennen auf“: die nominale Ergänzung im Nominativ kann semantisch auf das Merkmal „etwas Menschliches“ (hum) beschränkt werden, das zweite Argument der präpositionalen Ergänzung schränken wir auf etwas Abstraktes ein. Es ergibt sich folgende Erwartungsformel:*

$$
\begin{aligned}
L_{brenn\_auf} \Rightarrow & (\exists x_1 \exists x_2 \in [min, max] \\
& (ERG_{hum}(x_1, x_2) \wedge ERG_{NOM}(x_1, x_2) \wedge \\
& \neg \exists x_3 \neg \exists x_4 \in [min, max](x_1 \neq x_3 \wedge x_2 \neq x_4 \wedge \\
& ERG_{NOM}(x_3, x_4))) \wedge \\
& (\exists x_5 \in [x_2, max](L_{auf} \wedge \\
& \exists x_6 \exists x_7 \in [x_5, max] \\
& (ERG_{abstrakt}(x_6, x_7) \wedge ERG_{AKK}(x_6, x_7)))))
\end{aligned}
$$

*Das semantische Prädikat* $ERG_{abstrakt}$ *ist dabei analog zu* $ERG_{konkret}$ *definiert:* $ERG_{abstrakt}(x_1, x_2) \Leftrightarrow \exists x \in [x_1, x_2] abstrakt(x)$.

# Kapitel 4

# Lexikalische Wissensbasen - Übersicht und Anforderungen

Da die Lexikographie auf eine lange Tradition zurückschauen kann und daher ein riesiges Potential an lexikalischen Ressourcen vorzuweisen hat, mittlerweile auch in digitalisierter Form, sollen in diesem Kapitel nicht nur NLP-Lexika bzw. lexikalische Wissensbasen für natürlichsprachliche Anwendungen behandelt werden, sondern auch traditionelle Lexika in die Betrachtungen mit einbezogen werden. Betrachtet man traditionelle Lexika, so ist festzustellen, daß diese als Nachschlagewerke für menschliche Benutzer konzipiert sind und ein gewisses Maß an Grundwissen voraussetzen. Daher müssen bestimmte Informationen nicht oder nur implizit kodiert werden, da sie vom Benutzer inferiert werden können. Auch bez. der Repräsentation der Information wird ein gewisses Maß an Grundwissen vorausgesetzt. Generell kann man behaupten, daß bei der Erstellung von traditionellen Lexika ein sinnvolles Gleichgewicht zwischen dem Platzbedarf, dem Zeitaufwand, den ein Benutzer benötigt um eine gesuchte Information zu erhalten, und dem vorgesehenen Informationsgehalt angestrebt wird. Traditionelle Lexika weisen in bezug auf maschinelle Sprachverarbeitung folgende Mängel auf [MM95]:

- Unvollständige Kodierung der Information,
- Inkonsistente Kodierung,

- Unvollständigkeit bzgl. der Stichwörter.

Im Gegensatz zu traditionellen Lexika dienen lexikalische Wissensbasen (LKB[1]) in erster Linie als Datenbasis für sprachverarbeitende Systeme. Unabhängig von der intendierten Anwendung (wie z.B. Lemmatisierer oder Rechtschreibprüfung) sollte mittels der LKB jedes Element eines Textes identifizierbar sein. Da der gesamte Sprachumfang einer natürlichen Sprache weder von einem „native Speaker“ noch von einem (digitalen) Lexikon erfaßt werden kann und jede natürliche Sprache einem ständigen Wandel unterliegt, ist die Veränderbarkeit und Erweiterbarkeit einer lexikalischen Wissensbasis von zentraler Bedeutung. Aus informatischer Sicht stellen sich weiterhin die interessanten Fragen des automatischen Wissenserwerbs und wie mit lexikalischen Lücken umzugehen ist.
Eine weitere zentrale Forderung an eine lexikalische Wissensbasis besteht in der Existenz von definierten Schnittstellen, die einen Import und einen Export von lexikalischem Wissen erlauben. Hier sind insbesondere die Bemühungen der TEI (Text Encoding Initiative) und die Sprache SGML (Standard Generalized Markup Language) zu nennen (siehe [CHu95a, CHu95b, CHu95c]).

## 4.1 Speicherstrategien für Lexika

Man kann vorhandene Systeme unter folgenden Gesichtspunkten untersuchen:

> Existiert ein separates Lexikon?

Die meisten NLP-Systeme werden über ein separates Lexikon verfügen. Ein anderer Weg ist, die lexikalischen Informationen als zusätzliche Regeln zu formulieren. In diesem Fall besteht kein Bedarf an einem separaten Lexikon, da lexikalische Regeln als ein weiterer Typ von (Phrasenstruktur-) Regeln aufgefaßt werden.

> Falls ein separates Lexikon existiert, werden die Informationen aus einem permanenten oder virtuellen Lexikon abgerufen?

[1] Lexical Knowledge Base

Church [Chu80] prägte den Begriff des *virtuellen Lexikons*, in dem nur die Informationen permanent gespeichert sind, die für die Generierung der (morphologischen) Variationen eines Eintrags notwendig sind. Die Wortformen werden dann bei Bedarf generiert.
Die Einträge traditioneller Lexika umfassen die Grundformen (z.B. Duden) oder bei zweisprachigen Wörterbüchern zum Teil auch die Vollformen. NLP-Lexika sind entweder als Vollformen-, Stammformen-, Morphem- oder Grundformenlexika organisiert.

### 4.1.1 Grundformenlexikon

Beim Grundformenlexikon wird per Konvention eine Form des Paradigmas als Grundform ausgezeichnet; z.B. ist in der Regel bei nominalen Einträgen der Nominativ Singular ausgezeichnet, bei verbalen Einträgen der Infinitiv. Bei Verwendung von Grundformenlexika muß der Benutzer in der Lage sein, ein Wort auf seine entsprechende Grundform zu reduzieren. Bei Sonderfällen oder mehrsprachigen Lexika werden zusätzlich repräsentative Flexionsformen mit Verweis auf ihre Grundformen ausgezeichnet.
Die Konzeption eines NLP-Lexikon als Grundformenlexikon setzt analog eine Morphologiekomponente voraus, die jede im Text vorkommende Wortform auf ihre Grundform reduzieren kann. Die Effizienz und Korrektheit der natürlichsprachlichen Anwendung hängt somit wesentlich von der Effizienz und Qualität dieser Morphologiekomponente ab. Als Verfahren bietet sich beispielsweise die 2-Ebenen-Morphologie [Kos83][2] an. Ausgehend von einer Grundform, werden die Kombinationsmuster und Morphologieregeln (Umlautung etc.) abstrakt definiert und durch Verwendung von endlichen Zustandsautomaten (*finite state Automaton*) erhält man eine effiziente Generierungs- und Analysemöglichkeit. Im allgemeinen werden die Regeln, die eine Veränderung der Orthographie beschreiben in der Automatenkomponente kodiert, während Stämme und Affixe in der Lexikonkomponente gespeichert sind. Das Modell geht von zwei unterschiedlichen Repräsentationen einer Wortform aus: zum einen repräsentiert die Oberflächenform

[2] In der Literatur auch als KIMMO-System beschrieben

die Wortform so, wie sie in einem Text erscheint; zum anderen besteht das Wort auf der lexikalischen Ebene aus einer Folge von Stamm und Affixen, ohne Umlautung etc. Die Automatenkomponente besteht aus mehreren finite state Automaten (FSA) mit zwei Köpfen, die parallel die Korrespondenz von Oberflächen- und Lexikonform inspizieren (Abb. 4.1 aus [BBR88]).

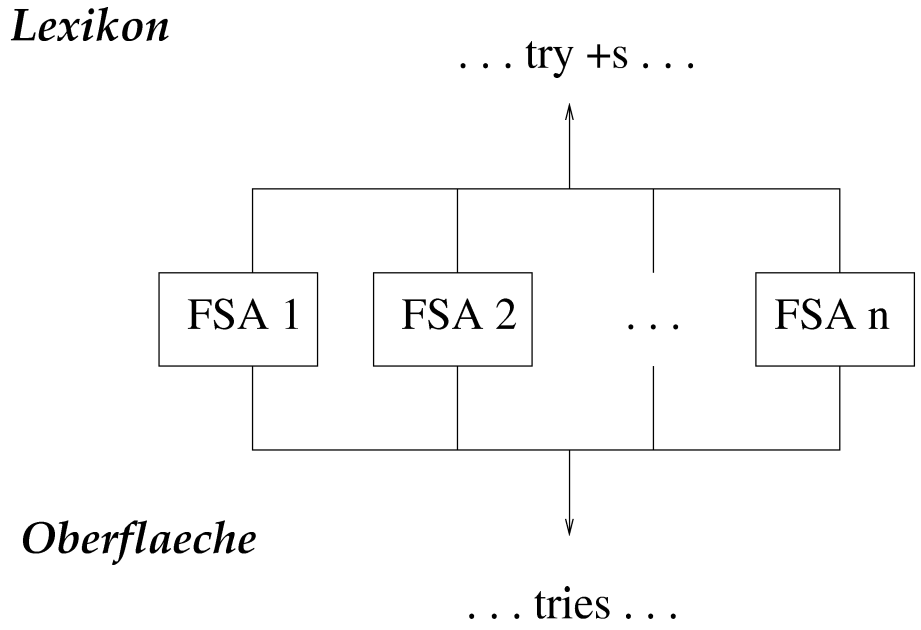


Abbildung 4.1: Automatenkomponente zur Inspektion der Lexikon-Oberflächen Korrespondenz

Jede Regel, die eine Veränderung der Orthographie beschreibt, wird als ein Automat implementiert. Man kann sagen, daß jeder Automat der Korrespondenz zwischen Lexikon- und Oberflächenform gewisse Beschränkungen auferlegt. Zum Beispiel lautet eine einfache Regel für den Wechsel von „y“ nach „i“: y korrespondiert in der Oberflächenform zu i, wenn y lexikalisch vor „+s“ auftritt.
Der folgende Automat (Tabelle 4.1 aus [BBR88]) mit zwei Köpfen repräsentiert diese Regel. Der Automat startet im Zustand 1 und wechselt die Zustände in Abhängigkeit der gelesenen Buchstabenpaare (lexikalische Form, Oberflächenform). Die Notation folgt der von [Kos83], wobei der Doppelpunkt nach dem Zustand einen Endzustand markiert und „=“ eine spezielle Art der Wildcard darstellt.

| | y | y | + | s | = | lexikalische Form |
|---|---|---|---|---|---|---|
| | i | y | = | s | = | Oberflächenform |
| state 1: | 2 | 4 | 1 | 1 | 1 | normaler Zustand |
| state 2. | 0 | 0 | 3 | 0 | 0 | +s erforderlich |
| state 3. | 0 | 0 | 0 | 1 | 0 | s erforderlich |
| state 4: | 2 | 4 | 5 | 1 | 1 | +s verboten |
| state 5: | 2 | 4 | 1 | 0 | 1 | s verboten |

Tabelle 4.1: Y-Wechsel

Im Beispiel „try+s“/„tries“ durchläuft der Automat die Zustandsfolge 1,1,1,2,3,1 und akzeptiert die Korrespondenz.
Mit dem Verfahren der 2-Ebenen Morphologie wurden bereits einige effiziente Morphologiekomponenten für verschiedene Sprachen entwickelt. So existieren für das System PC-KIMMO [Ant90] Beschreibungen in Form von Lexika und Regeln für die Sprache Englisch [Ant90] und Deutsch [Sch94]. Auch bei anderen 2-Ebenen-Systemen wie KIMMO [Kar83] oder LEXC [Kar93] existieren Beschreibungen für Englisch, Deutsch, Französisch u.a. Generell kann man jedoch zeigen, daß das Verfahren zu der Klasse der NP-vollständigen Probleme gehört (siehe [BBR88]).

### 4.1.2 Stammlexikon

Im Gegensatz zur Grundform, die per Konvention festgelegt wird, handelt es sich bei einem Stamm um einen String, der in der Regel nicht Bestandteil des Paradigmas ist, sondern als Basis für die Affigierung mit Flexiven und für Wortbildungsprozesse dient. Daher können im Lexikon zu einem Lemma mehrere Stämme zusammen mit ihren morphosyntaktischen Merkmalen eingetragen sein. Diese Vorgehensweise bedeutet eine strikte Trennung zwischen konkatenativen und nichtkonkatenativen Prozessen (wie z.B. Umlautung). Die nicht-konkatenativen Prozesse müssen nicht mehr über eine Regelkomponente ausgeführt werden, sondern werden durch explizite Kodierung der verschiedenen Stämme realisiert. Der Vor-

teil gegenüber den Grundformenlexika liegt in wesentlich effizienteren Verfahren für die Morphologiekomponente bzw. für den *lexikalischen Lookup*.

### 4.1.3 Morphemlexikon

Beim Morphemlexikon werden nicht nur die Stämme bzw. Stammorpheme gespeichert, sondern alle Morpheme als Einträge betrachtet. Die Morpheme lassen sich in Flexions-, Derivations- und Stammorpheme subklassifizieren. Mit Hilfe von Kombinationsregeln kann eine gegebene Wortform in ihre elementaren Bestandteile zerlegt werden. Der relativ aufwendige Segmentierungsalgorithmus ist jedoch für große Textmengen ungeeignet.

### 4.1.4 Vollformenlexikon

Bei den bisher betrachteten Speicherstrategien war für die Analyse (bzw. Lemmatisierung) eine zusätzliche Morphologiekomponente notwendig. Da in einem Vollformenlexikon sämtliche mögliche Wortformen gespeichert sind, können für den lexikalischen Lookup Standardsuchverfahren eingesetzt werden. Demgegenüber stehen ein erhöhter Speicherplatzbedarf und Mehraufwand bei der Akquisition und Pflege des lexikalischen Wissens.

### 4.1.5 Hybride Repräsentation

Wir wollen nun eine hybride Form der Speicherung vorstellen, die Vorteile der Stamm- und Vollformenlexika in sich vereint. Bei Vollformenlexika werden als Nachteile in erster Linie der erhöhte Speicheraufwand und der Mehraufwand bei Akquisition und Pflege der Datenbasis angeführt. Wird das Akquisitionstool jedoch mit einer Morphologiekomponente ausgestattet, die den Anwender bei der Erfassung der Daten unterstützt, so kann der Akquisitionsaufwand auf den Level der Stammformenlexika gedrückt werden. Zum Beispiel müßten in einem reinen Vollformenlexikon für verbale Einträge 29 Formen akquiriert werden. Demge-

genüber stehen bei Verwendung der Morphologiekomponente lediglich 4 Formen[3] bei starken Verben und eine Form bei schwachen Verben. Ein weiteres Argument gegen Vollformenlexika kann durch die Integration einer Morphologiekomponente entkräftet werden, nämlich der Mangel an Generalisierungsmöglichkeiten durch Angabe von Regeln. Durch die Einbettung der Morphologiekomponente werden Generalisierungen durch Angabe von Regeln unterstützt. Insgesamt sehen wir in dieser Vorgehensweise folgende Vorteile:

1. Flexibilität: Sonderformen, die nicht über die Regelkomponente abgedeckt werden, können jederzeit manuell eingetragen bzw. überarbeitet werden.

2. Effizienz: Die morphologische Analyse reduziert sich bei einfachen Formen[4] auf den lexikalischen Lookup mit Standardsuchverfahren.

Aufgrund der heutigen Speicherkapazität und des Preisverfalls im Bereich Speichermedien ist der erhöhte Speicheraufwand kein gravierender Nachteil mehr. In Fällen, wo der Speicherplatzbedarf eine Rolle spielen sollte, kann unser System durch leichte Modifikationen dahingehend geändert werden, daß nicht sämtliche Vollformen permanent gespeichert werden, sondern ein Teilausschnitt der Vollformen bei Bedarf generiert wird. Diese Vorgehensweise beeinflußt natürlich die Performance der morphologischen Analyse.

## 4.2 Lexikalische Informationen für NLP-Systeme

Bei der Entwicklung von Lexika für NLP-Systeme lassen sich grundsätzlich zwei Arten unterscheiden: Syntaktisch orientierte Systeme, die ihre lexikalischen Ein-

[3] In Grammatiken der Deutschen Gegenwartssprache wird in der Regel von vier Stammformen bei starken Verben ausgegangen. Bei der Entwicklung der Generierungsregeln hat sich jedoch gezeigt, daß die Hinzunahme einer fünften Stammform erlaubt, die Wortformen einiger „unregelmäßiger Verben“ mittels der Regeln zu erzeugen.

[4] In dem verfolgten Ansatz wird bei der Analyse eine Regelkomponente für Komposita eingesetzt, die zusammengesetzte Formen auf einfache Wortformen reduziert.

träge nach traditionellen Kategorien wie Wortklassen klassifizieren und semantisch orientierte Systeme, die ihre Einträge nach funktionalen Gesichtspunkten klassifizieren. Die Wiederverwendbarkeit von lexikalischen Ressourcen erscheint bei semantisch orientierten Systemen schwieriger. Betrachten wir beispielsweise das Lexikon für das System LADDER [HSSS78], so finden wir u.a. die lexikalischen Items ATTR und PRESENT. Es handelt es sich bei der lexikalischen Klasse ATTR um die Wortklasse der Nomen, so daß diese einer traditionellen Kategorie zugeordnet werden kann. Unter der Klasse PRESENT sind jedoch Einträge wie GIVE ME, WHAT IS und WHAT ARE zusammengefaßt, welche nicht als einzelne Einträge in einem syntaxorientierten Lexikon zu finden sind.
Die lexikalischen Informationen, die in einer lexikalischen Wissensbasis gespeichert werden, können in zwei grundsätzliche Sorten unterteilt werden: Informationen, die direkt von dem NLP-System verwendet werden und Zusatzinformationen[5] (wie Beispielsätze, Erklärungen und Kommentare), welche nicht unmittelbar von dem NLP-System verwendet werden. Systemspezifische Informationen behandeln Wissen über Morphologie, Syntax und Semantik. Auf der Ebene der systemspezifischen Informationen unterscheiden wir zwischen den Informationen, die einem lexikalischen Eintrag (im folgenden mit Intra-Lexem Informationen bezeichnet) direkt zugeordnet werden und klassenspezifischen Informationen (Inter-Lexem Informationen).

### 4.2.1 Intra-Lexem Informationen

Die meisten syntaktischen und semantischen Informationen gehören zu der Klasse der Intra-Lexem Informationen. Als erstes sind hier Informationen über die Wortklassenzugehörigkeit zu nennen. Dieser Informationstyp beinhaltet die traditionellen Kategorien wie Nomen, Verb, Adjektiv oder Adverb.
Weiterhin werden in unserem Ansatz Informationen über Valenzstrukturen bei Verben und Adjektiven den einzelnen Lexikoneinträgen zugeordnet. Hierbei wird sich die Valenzbeschreibung nicht nur auf die syntaktische Ebene beziehen, son-

[5] Diese Zusatzinformationen können als Grundlage für eine „lexikographische Arbeit“ mit der lexikalischen Wissensbasis dienen.

dern auch semantische Spezifikationen umfassen.
Zu den Intra-Lexem Informationen zählen wir außerdem statistische Informationen über die Auftrittshäufigkeit von bestimmten Wortformen und Valenzmustern. Durch eine spätere Kopplung der LKB mit Textkorpora könnten die Lexem-Beschreibungen einer statistischen Untersuchung unterzogen und so empirisch fundiert werden.

### 4.2.2 Inter-Lexem Informationen

Dieser Informationstyp setzt lexikalische Einträge mit anderen lexikalischen Einträgen oder Wörtern in Verbindung. Morphologische Informationen, wie Flexionsparadigmen, zählen zu den Inter-Lexem Informationen, die klassenspezifisches Wissen beschreiben. Aber auch Informationen über die „Durchlässigkeit“ von Wortklassen zählen zu dieser Art von Informationen[6].

[6] Z.B. kann die Regel: „Jeder Artikel kann auch als Pronomen verwendet werden“, als Inter-Lexem Information aufgefaßt werden.

# Kapitel 5

# Die lexikalische Wissensbasis von *PARLEX*

Das System der lexikalischen Wissensbasis kann konzeptuell in die drei Komponenten Akquisitionstools, Lemmatisierungskomponente und Vollformengenerierungskomponente unterteilt werden (s.a. Abbildung 5.1). Dabei greifen die Komponenten auf zwei verschiedene Datenbasen (Wortklassendatenbasis und Vollformendatenbasis) zu. Die Akquisitionstools, die ohne Kenntnis der internen Wissensrepräsentationsstrukturen benutzt werden können, unterstützen die Anwender regelbasiert bei der Erfassung neuer Daten. Die Informationseinträge beziehen sich auf morphosyntaktisches und semantisches Wissen. Ein fertig spezifizierter lexikalischer Eintrag wird mittels der Vollformengenerierungskomponente in einen Eintrag der Vollformendatenbank übersetzt. Auch die Inspektion und Veränderung von lexikalischen Einträgen wird durch die Akquisitionstools unterstützt. Dabei wird im Änderungsmodus die Konsistenz von Wissensbasis und Vollformendatenbank automatisch gewährleistet. Die Komponente der Lemmatisierung greift ausschließlich auf die Daten der Vollformendatenbank zu.

Der Algorithmus zur morphologischen Analyse ist dabei zweistufig: zuerst wird in der Vollformendatenbank nachgeschlagen, ob ein entsprechender lexikalischer Eintrag vorhanden ist. Im negativen Falle wird in einer zweiten Stufe ein Algorithmus zur deutschen Kompositabehandlung aktiviert, in dem die laufende

Wortform in bekannte lexikalische Einheiten zerlegt wird.

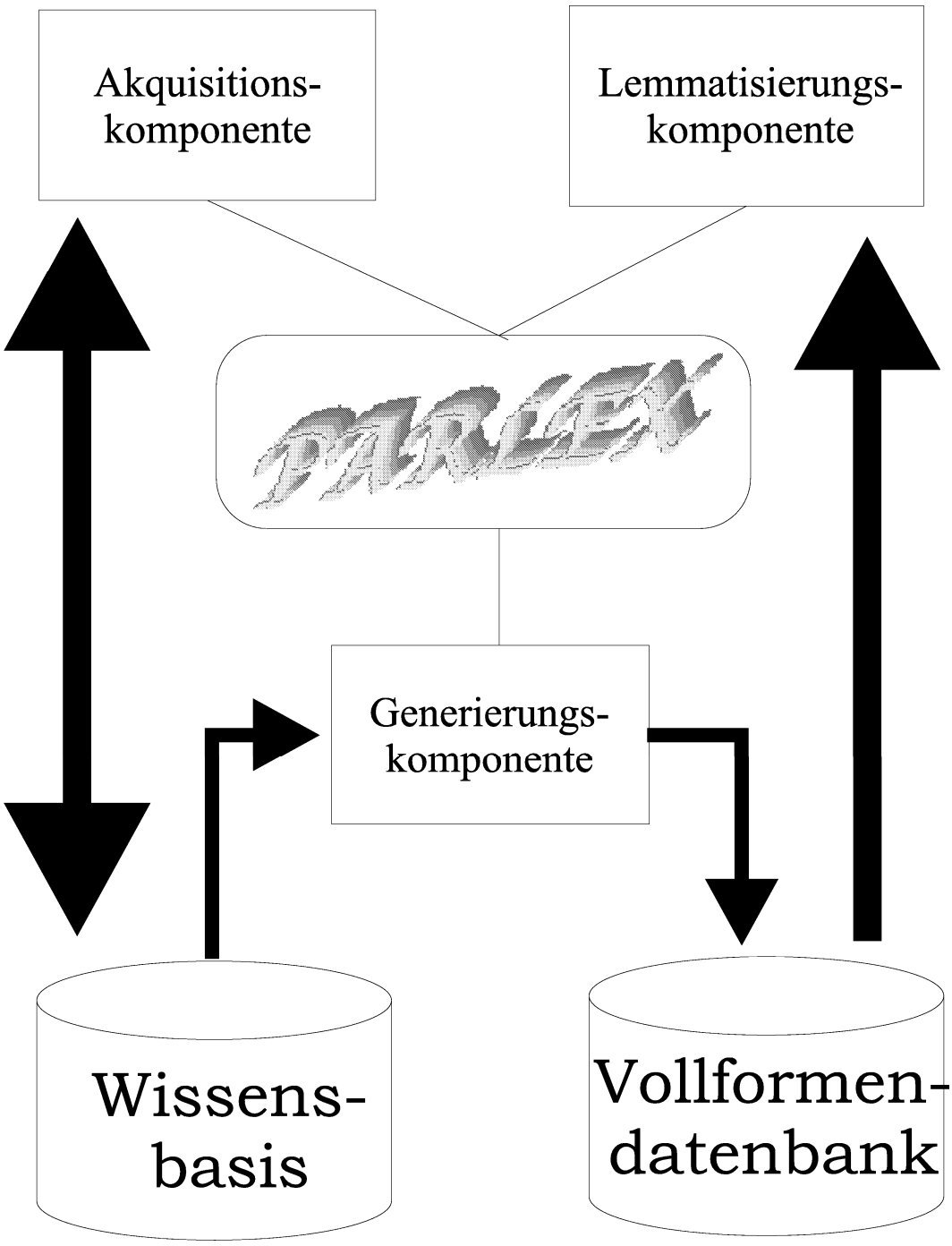


Abbildung 5.1: Lexikalische Wissensbasis - Systemübersicht

In den folgenden Abschnitten beschreibe ich zuerst das syntaktisch-semantische Wissen, welches in der Wissensbasis von $\mathcal{PARLEX}$ repräsentiert werden kann. Anschließend gehe ich allgemein auf die Merkmale der beiden Datenbasen ein, um zum Ende des Kapitels die Funktionalität der drei Komponenten zu beschreiben. Weiterhin findet die Leserin in diesem Kapitel die linguistischen Grundlagen beschrieben, die für die Konzeption der lexikalischen Wissensbasis von Belang sind.

## 5.1 Syntaktisch-semantisches Wissen in $\mathcal{PARLEX}$

Die Repräsentation der Valenz[1] erfolgt in $\mathcal{PARLEX}$ auf syntaktischer und semantischer Ebene. Valenzmuster oder Strukturen können der Klasse der Verben und der Adjektive zugeordnet werden. Da die semantische Beschreibung der Valenz sehr eng mit der semantischen Beschreibung von Nomina verknüpft ist, behandelt der nächste Abschnitt zuerst die semantische Klassifikation der Nomina.

### 5.1.1 Semantische Kategorisierung der Nomina

Zur Beschreibung der Valenz kategorisieren wir die Klasse der Nomen semantisch auf zwei verschiedenen Ebenen:

1. Differenzierung der Objekte in
   - Konkreta: sinnlich wahrnehmbare Objekte oder Dinge,
   - Abstrakta: sinnlich nicht wahrnehmbare, bzw. rein begriffliche Objekte oder Dinge.
2. Subklassifikation in kategorielle Bedeutung.

Engel [Eng91] schlägt zur Spezifikation der kategoriellen Bedeutungen der Nomina ein dreistufiges Verfahren vor, das von einer überschaubaren Menge allgemeiner Bedeutungsmerkmale ausgeht, bei Bedarf eine Menge spezieller Merkmale verwendet und im Falle sehr eingeschränkter Kombinationsmöglichkeiten Einzelwörter angibt. Nach Engel haben sich an allgemeinen Bedeutungsmerkmalen

[1] Zur Beschreibung der Valenz siehe Abschnitt 5.1.2

der ersten Stufe aufgrund langjähriger semantischer Studien die folgenden zwölf Kategorien als brauchbar erwiesen [Eng91]:

Tabelle 5.1: Kategorielle Bedeutungen der ersten Stufe

| | |
|---|---|
| akt | Geschehensablauf, Vorgang, Tätigkeit |
| geg | Gegenständliches, sinnlich Wahrnehmbares, unbelebt und zählbar |
| hum | Menschen, Menschliches, auch menschliche Körperteile |
| inst | von Menschen geschaffene Institutionen (z.B. Stadtverwaltung, Ausschuß, Parlament) |
| intell | Nur-Geistiges, Nicht-Sinnliches, Begriff u.a. |
| loc | räumliche Bestimmung |
| mat | Gegenständliches, sinnlich Wahrnehmbares, unbelebt und nicht zählbar (z.B. Feuer, Staub, Wasser) |
| plant | Pflanze(n), Pflanzliches |
| sent | Gefühl, Empfindung |
| stat | Zustand, Eigenschaft |
| temp | zeitliche Bestimmung |
| zool | Tiere |

Die Merkmale der zweiten Stufe, erst recht die Einzelwörter der dritten Stufe, können nicht mehr einfach aufgezählt werden. Sie werden nach Bedarf ausgewählt ([Eng91], Seite 359):

> ... *Wir wählen sie nach Bedarf und in der Zuversicht, später aus den induktiv gewonnen Merkmalen ein überschaubar organisiertes Inventar gewinnen zu können. Zu den besonders häufig benötigten Merkmalen dieser Stufe gehören etwa Artefakt (von Menschen hergestellter Gegenstand, z.B. Apparat), Behälter, erwachsen, Fahrzeug, ...*

Betrachten wir nun das vorhandene Inventar zur semantischen Klassifikation eines Nomens, so stellt sich die Frage, wie man Mehrfachauswahlen bei den semantischen Kategorisierungen behandelt. Bei der Klassifikation auf der obersten

Ebene in Konkreta bzw. Abstrakta erscheint es sinnvoll, Mehrfachauswahlen als Disjunktionen zu behandeln.
Bei der Behandlung von Mehrfachauswahlen kategorieller Bedeutungen sind zwei Sichtweisen möglich:

1. Die Auswahl mehrerer Bedeutungsmerkmale wird als schrittweise Verfeinerung der Bedeutung betrachtet.

2. Die Auswahl wird als disjunktive semantische Kategorisierung aufgefaßt.

In unserem System sind beide Sichtweisen integriert. Es können einem nominalen Eintrag mehrere Listen von kategoriellen Merkmalen zugeordnet werden. Dabei sind die Merkmale einer Liste konjunktiv verknüpft (schrittweise Verfeinerung) und die Listen selbst disjunktiv verknüpft.

#### 5.1.1.1 Ordnungsrelation auf den kategoriellen Bedeutungen

Um das Ziel eines *überschaubar organisierten Inventars* zu erreichen, wollen wir zusätzlich eine Ordnungsrelation auf den kategoriellen Bedeutungen definieren. Betrachtet man die vorgeschlagenen Kategorien der ersten Stufe, so erscheint eine hierarchische Anordnung sinnvoll. Engel faßt z.B. die Merkmale der ersten Stufe geg,hum,inst,mat,plant und zool unter dem Sammelbegriff konkret (konkr) und die Merkmale akt,intell,sent,stat entsprechend unter abstrakt (abstr) zusammen. Auf der zweiten Beschreibungsebene der kategoriellen Bedeutungen erscheint es weiterhin möglich, daß Subkategorien eingeführt werden, die sich nicht mehr in eine hierarchische Ordnung einfügen lassen, so daß wir die Ordnungsrelation auf den semantischen Kategorisierungen als Heterarchie definieren:

**Definition 5.1 (Semantische Ordnungsrelation)**
*Sei* $K := \{k_1, k_2, \ldots, k_n\}$ *die Menge der semantischen Merkmale in Form von kategoriellen Bedeutungen oder abstr/konkr. Die semantische Ordnungsrelation* $S$ *ist wie folgt definiert:* $(k_i, k_j) \in S \Leftrightarrow k_i \in k_j$ *und hat folgende Eigenschaften:*

1. *S ist transitiv:* $\forall k_i, k_j, k_l \in K((k_i, k_j) \in S \wedge (k_j, k_l) \in S \Rightarrow (k_i, k_l) \in S)$

2. *S ist asymmetrisch:* $\forall k_i, k_j \in K((k_i, k_j) \in S \Rightarrow (k_j, k_i) \notin S)$

3. *S ist reflexiv:* $\forall k \in K((k,k) \in S)$

Die folgende Abbildung 5.2 spiegelt die angeführten Überlegungen wieder. Die kategoriellen Bedeutungen der ersten Stufe sind in der Abbildung in eine Hierarchie eingebettet. Durch Hinzunahme der kategoriellen Bedeutungen zweiter Stufe Säugetiere (säug), Wasserlebewesen (wzool) und Wale (wal) wird das Prinzip der Hierarchie durchbrochen und wir erhalten eine Heterarchie.

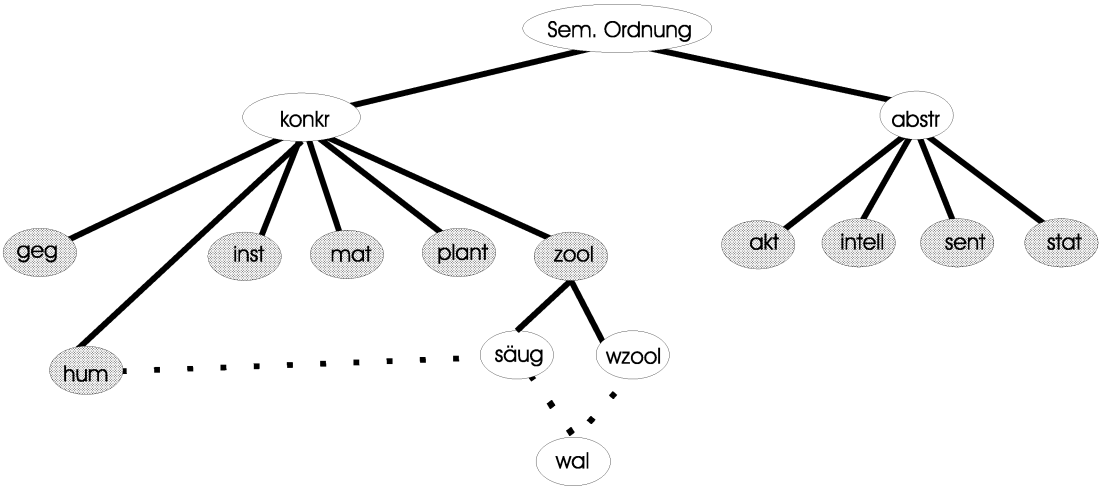


Abbildung 5.2: Semantische Ordnungsrelation (Heterarchie) auf den kategoriellen Bedeutungen

#### 5.1.1.2 Beispiele semantischer Klassifikationen bei nominalen Einträgen

Betrachten wir als erstes das Nomen *Bank*. Wenn wir den entsprechenden Kontext nicht kennen, so assoziieren wir mindestens zwei verschiedene semantische Verwendungsweisen:

1. Die Bank als ein Geldinstitut.

2. Die Bank als Sitzmöbel.

Bei den morphlogischen Eigenschaften unterscheiden sich die beiden Verwendungsweisen zwar nicht im Genus, jedoch ihre Pluralform ist unterschiedlich:

Banken - Bänke. Damit ist klar, daß die beiden Erscheinungsformen aufgrund ihrer unterschiedlichen morphologischen Ausprägungen als zwei eigenständige Einträge in der lexikalischen Wissensbasis behandelt werden müssen. Die semantische Klassifikation lautet unter Annahme der kategoriellen Bedeutungen aus Abbildung 5.2:

- Bank-Bänke: konkr,geg als das Sitzmöbel
- Bank-Banken:
  - konkr,inst als die Institution Bank
  - konkr,geg als das Gebäude Bank

Wir erhalten somit zwei Einträge in der lexikalischen Wissensbasis, wobei dem zweiten Eintrag zwei disjunkte semantische Beschreibungen zugeordnet sind.

Beim zweiten Beispiel *Ball* betrachten wir die beiden Verwendungsweisen:

- Ball als Sportgerät (Fußball, Handball, etc.)
- Ball als Veranstaltung (Tanzball, Opernball ... )

Hierbei ist festzustellen, daß sich die morphologischen Ausprägungen bei den beiden Verwendungsweisen nicht unterscheiden, so daß ein Eintrag in der lexikalischen Wissensbasis mit zwei disjunkten semantischen Beschreibungen ausreichend ist.

### 5.1.2 Die Valenztheorie

Die Einführung des Begriffs der Valenz in die Linguistik wird hauptsächlich Lucien Tesnière [Tes59, Tes80] zugeschrieben. In seinen Arbeiten zur strukturalen Syntax nimmt die Valenz eine zentrale Stellung ein. Das Grammatikmodell von Tesnière ist dabei verb-zentriert, d.h. das Verb ist der oberste Regent in einem Satz. Bei den vom Verb unmittelbar regierten Satzelementen unterscheidet Tesnière zwischen den Handelnden (actants) und den Umständen (circonstants).

Während die Anzahl der „circonstants“ nicht beschränkt ist, ist die Anzahl der „actants“ durch das Verb beschränkt. In Anlehnung an die Wertigkeit eines Atoms nennt er diese Zahl die Valenz. Die möglichen Verbvalenzen reichen von null bis drei:

- avalente Verben (kein „actant“),
- monovalente Verben (ein „actant“),
- bivalente Verben (zwei „actants“),
- trivalente Verben (drei „actants“).

Nach Tesnière werden die Handelnden durch Substantive oder deren Äquivalente ausgedrückt, die Umstände durch Adverbien. Er unterscheidet bei den „actants“ zwischen drei verschiedenen Typen[2], denen er bestimmte syntakische und semantische Funktionen zuweist. Der erste „actant“ fungiert im allgemeinen als Subjekt, seine semantische Funktion ist die des Ausführenden der Handlung. Der zweite „actant“ realisiert das Objekt im Akkusativ und bezeichnet denjenigen, der die Handlung erfährt. Den dritten „actant“ setzt Tesnière mit dem indirekten Objekt gleich, der semantisch als Nutznießer oder Geschädigter der Handlung fungiert. Kritik an seinem Ansatz wurde im wesentlichen aus zwei Gründen geübt:

1. Sein Modell ist zu sehr an den morphosyntaktischen Merkmalen des Französischen orientiert.
2. Er stellt eine zu direkte Verbindung zwischen Syntax und Semantik her.

Bei der Weiterentwicklung der Valenztheorie (siehe z.B. [HS91, Hel92, Her96]) stehen insbesondere folgende Fragen im Vordergrund:

1. Welche Ebenen der Valenz lassen sich unterscheiden und in welcher Beziehung stehen die Ebenen zueinander?

[2]entsprechend der maximalen Anzahl von „actants“.

2. Welche Wortarten besitzen noch Valenzeigenschaften und wie lassen sich diese charakterisieren?

3. Lassen sich andere Valenzgebundenheiten als „actant“ und „circonstant“ finden?

#### 5.1.2.1 Die Valenzebenen

Auch in der Valenztheorie wird die in der Grammatik übliche Unterscheidung zwischen Syntax, Semantik und Pragmatik durchgeführt. Keine Einigkeit herrscht jedoch darüber, was unter der Valenz auf den verschiedenen Ebenen zu verstehen ist und wie die Zusammenhänge zwischen den Ebenen zu beschreiben sind (vgl. dazu auch [Hel92]). Im folgenden werden wir eine kurze Charakterisierung der drei Ebenen durchführen.

**5.1.2.1.1 Syntaktische Valenz** Die syntaktische Valenz legt fest, wie viele Leerstellen ein Wort (Valenzträger) eröffnet und mit welchen Ergänzungen die Positionen zu belegen sind. Weiterhin legt die syntaktische Valenz die Modalität der Ergänzungen fest.

**5.1.2.1.2 Semantische Valenz** Die semantische Valenz regelt die Kombination von Wortbedeutungen oder Wortgruppenbedeutungen: In der Umgebung einer bestimmten Bedeutung sind nur gewisse Bedeutungen zulässig, wieder andere ausgeschlossen. Es handelt sich also um Bedeutungsangaben, die sich auf die Umgebung der betreffenden Wörter beziehen. Diese kombinatorischen Bedeutungen legen bei den Verben zum einen fest, welche semantischen Merkmale (Prädikate) eine Ergänzung aufweisen muß. Zum anderen geben sie an, welche semantische Rolle eine Ergänzung einnimmt. Wir sprechen im ersten Fall von *kategorieller Bedeutung*, im zweiten Fall von *relationaler Bedeutung*.
Die Ausweitung des Valenzbegriffs auf die semantische Ebene ist besonders wichtig, da viele sprachliche „Fehler“ nur als Verstöße gegen Bedeutungsbeschränkungen erklärt werden können. Aus der Sicht der Grammatikentwicklung bedeutet dies die Vermeidung von Übergeneralisierungen. Weiterhin ist die Auflösung

von Mehrdeutigkeiten oftmals nur durch den Rückgriff auf semantisches Wissen zu leisten.

**5.1.2.1.3 Pragmatische Valenz** Unter pragmatischer Valenz wird allgemein der Einfluß verstanden, den die Kommunikationssituation auf die Regularitäten der syntaktischen und semantischen Valenz hat. In bestimmten Dialogsituationen oder Textsorten kann es z.B. erlaubt sein, obligatorische Ergänzungen wegzulassen (Ellipsen) oder Leerstellen mit Elementen zu besetzen, die den definierten semantischen Restriktionen nicht entsprechen (Metaphernbildung).

### 5.1.2.2 Valenzrepräsentation in $\mathcal{PARLEX}$

Das Ziel, ein Verb bzw. Adjektiv bezüglich seiner Umgebung möglichst vollständig zu beschreiben, führt uns zu zwei Ebenen der Valenzspezifikation:
Auf der syntaktischen Ebene beschreiben wir die Valenzrahmen, indem wir die Anzahl der Positionen (Slots) festlegen und bestimmen, mit welchen syntaktischen Strukturen die verschiedenen Slots gefüllt werden können. Die Definition dieser Valenzrahmen erfolgt unabhängig von einem spezifischen lexikalischen Eintrag; es handelt sich also um Inter-Lexem Informationen, die zu einer „syntaktischen Subklassifizierung“ der Wortklasse führen[3]. Als Ergänzungen sind zur Zeit Nominalphrasen (NP) und Präpositionalphrasen (PP) zugelassen.
Auf der zweiten Ebene werden die semantischen Einschränkungen und die Art der Ergänzung festgelegt. Diese Informationen sind als Intra-Lexem Informationen realisiert, d.h. einem bestimmten lexikalischen Eintrag zugeordnet. Die semantischen Einschränkungen können durch die Angabe von kategoriellen Bedeutungen bis hin zur Angabe von Einzelwörtern erfolgen. Bei der Spezifikation der semantischen Einschränkungen unterscheiden wir, ob es sich um Merkmale handelt, die auf eine Ergänzung zutreffen dürfen oder ausgeschlossen sind.
Heringer [Her96] betrachtet als Komplemente der Valenzframes Nominalphrasen, Präpositionalphrasen und Äquationsphrasen. Er subklassifiziert die Hauptverben

[3]Die Einteilung führt jedoch nicht zu disjunkten Klassen, da einem lexikalischen Eintrag mehrere syntaktische Valenzrahmen zugeordnet sein können.

dann in 14 Subklassen[4], wobei die Anzahl der Slots von 1 bis 3 reicht (Tabelle 5.2 aus [Her96]):

Tabelle 5.2: Syntaktische Valenz-Frames der Wortklasse Verb

| Monovalent mit Default 1 _nom | Bivalent mit Default 1 _nom 2 _akk | Trivalent mit Default 1 _nom 2 _akk 3 _dat |
|---|---|---|
| v_nom | v_nom_akk | v_nom_akk_dat |
| | v_nom_dat | v_nom_akk_all |
| | v_nom_gen | v_nom_akk_gen |
| | v_nom_prä | v_nom_akk_prä |
| | v_nom_äqu | v_nom_akk_äqu |
| | | v_nom_dat_prä |
| | | v_nom_dat_äqu |
| | | v_nom_prä_prä |

Bei der Wortklasse der Adjektive unterscheidet Heringer zwischen 8 syntaktischen Valenz-Frames (Tabelle 5.3 aus [Her96]):

Tabelle 5.3: Syntaktische Valenz-Frames der Wortklasse Adjektiv

| Monovalent mit Default 1 _nom | Bivalent mit Default 1 _nom 2 _prä | Trivalent mit Default 1 _nom 2 _prä 3 _prä |
|---|---|---|
| a_nom | a_nom_akk | a_nom_dat_gen |
| | a_nom_dat | a_nom_prä_prä |
| | a_nom_gen | |
| | a_nom_prä | |
| | a_nom_äqu | |

Bei der Beschreibung von Valenzstrukturen gibt es keine zuverlässigen Kriterien

[4]Es existieren theoretisch $5^3$ Kombinationen

zur Unterscheidung von „valenzgebunden“ und Adjunkten. Weiterhin gestaltet sich die Trennung der Argumente in obligatorische und fakultative Angaben als schwierig, so daß Jacobs [Jac94] den Begriff der *Valenzmisere* prägte[5]. Bei der lexikalischen Wissensbasis wird diese Problematik berücksichtigt, indem die getroffene Klassifizierung nicht als Dogma angesehen wird, sondern das System stellt eine Auswahl von syntaktischen Valenzmustern zur Verfügung, die als Vorschläge dienen und jederzeit erweiterbar sind.

**5.1.2.2.1 Beispiele für Valenzbeschreibung** Betrachten wir etwa das Verb spielen, so lassen sich nach Helbig [HS91] vier Varianten mit unterschiedlichen Bedeutungen unterscheiden[6]:

1. spielen als dreistelliges Verb im Sinne „sich mit Spielen beschäftigen“,
   *Er spielt mit seinem Freund um Geld.*

2. spielen als zweistelliges Verb im Sinne „nicht ernstnehmen“,
   *Er spielt nur mit dem Mädchen.*

3. spielen als zweistelliges Verb im Sinne „darstellen“,
   *Kainz spielt den Hamlet.*

4. spielen als zweistelliges Verb im Sinne „vor sich gehen, geschehen“,
   *Der Vorfall spielte in Berlin.*

Dem dreistelligen Verb „spielen“ können wir die Valenz-Frames v_nom_akk_prä und v_nom_prä_prä zuordnen.

[5] Zur Unterscheidung von Angaben und Ergänzungen sowie fakultativer und obligatorischer Ergänzung wurden einige Testverfahren entwickelt. Diese Testverfahren, wie z.B. der Reduktionstest zur Unterscheidung von Angaben und Ergänzung oder der Eliminationstest zur Behandlung der Modalität, sind bereits für menschliche Experten schwierig anzuwenden und lassen sich erst recht nicht maschinell operationalisieren.

[6] Alle Beispielsätze sind aus [HS91] entnommen

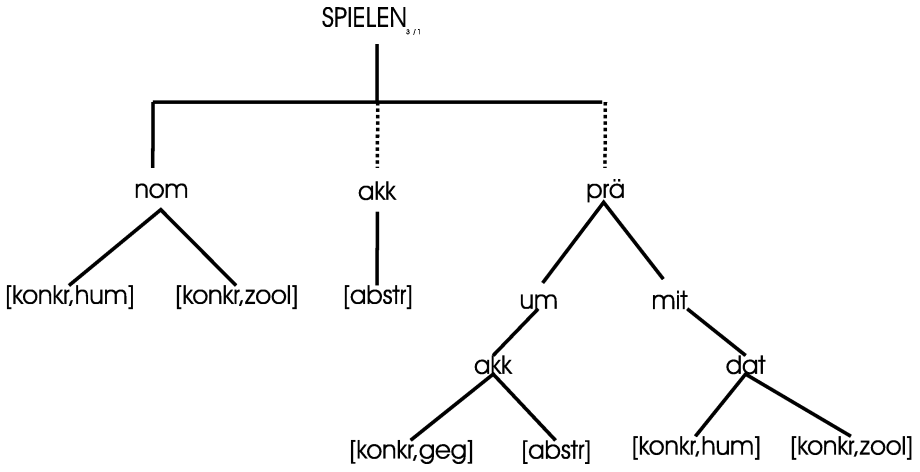


Abbildung 5.3: Eine Valenzbeschreibung für das Verb SPIELEN_nom_akk_prä

Die nominale Ergänzung der ersten Position kann semantisch als etwas Menschliches (hum) oder Tierisches (zool) charakterisiert werden. Die Akkusativergänzung ist semantisch mit etwas Abstraktem belegt, wie etwa Lotterie oder Schach. Beim ersten Valenz-Frame kann die Präpositionalergänzung mit den Präpositionen „um“ oder „mit“ besetzt sein. Bei der Präposition „um“ steht die regierte NP im Akkusativ und ist semantisch etwas Konkretes, Gegenständliches oder etwas Abstraktes. Bei der Präposition „mit“ steht die regierte NP im Dativ und ist semantisch als etwas Menschliches oder Tierisches charakterisiert. Alternativ kann auch etwas Gegenständliches diese Position füllen.

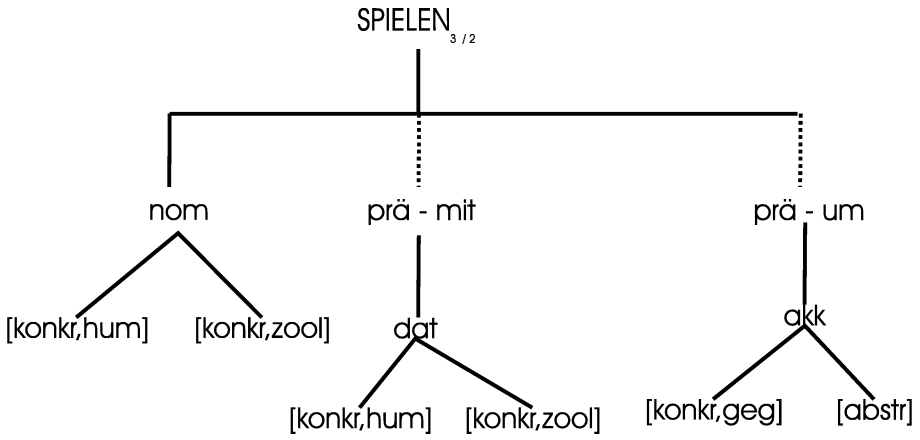


Abbildung 5.4: Eine Valenzbeschreibung für das Verb SPIELEN_nom_prä_prä

Beim zweiten Valenz-Frame ist die erste Präpositionalphrase mit der Präposition „mit“ realisiert und die zweite Präpositionalphrase mit „um“. Die semantischen Beschreibungen stimmen mit denen des ersten Frames überein. Bei beiden Frames sind die Positionen 2 und 3 als fakultative Ergänzungen realisiert.
Als zweites Beispiel möchte ich das Adjektiv „bekannt“ betrachten, dem man den Adjektiv-Frame a_nom_dat zuordnen kann.

*Der Artikel ist dem Wissenschaftler bekannt*

Eine Besonderheit ist nach Heringer [Her96] die syntaktische Behandlung der ersten Position (des Einser-Slots). Diese Position ist erst einmal blockiert und wird in verschiedenen Situationen unterschiedlich kodiert. In Verbalkomplexen erscheint der Einser-Slot z.B. über die durchlässigen Kopulaverben als Subjektiv. Der Einser-Slot ist mit einer Nominalphrase im Nominativ belegt und läßt sich semantisch kaum weiter einschränken. Der Zweier-Slot ist als NP im Dativ realisiert und wird für das Beispiel semantisch auf Personen oder Tiere eingeschränkt.

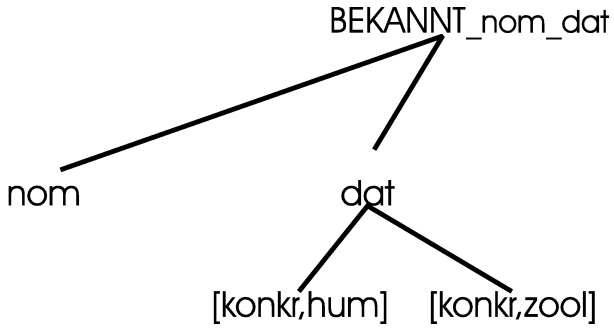


Abbildung 5.5: Eine Valenzbeschreibung für das Adjektiv BEKANNT_nom_dat

## 5.2 Wissensbasis und Vollformen-DB

Bevor die Struktur der Wissensbasis behandelt wird, sollen erst einmal Begriffe wie Wort, Wortformen und die Einteilung der lexikalischen Kategorien beschrieben werden.

### 5.2.1 Wort, Wortformen und Lexeme

Ein Wort wie *Haus* manifestiert sich in möglichen Wortformen Haus, Hauses, Häuser und Häusern. Komplexe Ausdrücke der natürlichen Sprache setzen sich aus Wortformen zusammen, z.B.:

> *Das Haus gefällt der Familie Widdig.*

In der deutschen Orthographie werden Wortformen durch Abstände voneinander getrennt, wobei Satzzeichen wie Punkt oder Komma zu berücksichtigen sind:

> *Eine Wortform ist eine Kette von alphanumerischen Zeichen mit Leerzeichen auf jeder Seite. Sie kann Bindestriche und Apostrophe enthalten, aber keine anderen Markierungen.*

Bei der Bildung von verschiedenen Wortformen unterscheidet man drei morphologische Prozesse:

- Flexion nennt man die systematische Variation, mit der ein Wort sich an verschiedene syntaktische Umgebungen anpaßt bzw. verschiedene syntaktische und semantische Funktionen ausübt.

- Derivation bezeichnet die Verbindung einer freien Wortform mit einem Affix. Sie beinhaltet sowohl die Suffigierung als auch die Präfigierung einer Wortform.

- Komposition bezeichnet die Verknüpfung zweier oder mehrerer frei vorkommender Wortformen zu einer neuen Wortform.

Wortformen werden in sogenannte Flexionsparadigmen zusammengefaßt, die Menge der Wortformen in einem Paradigma nennen wir Wort. Damit ist der Begriff Wort ein Abstraktum, das sich nur in Form der zugehörigen Wortformen konkretisiert.

> Wort $=_{def}$ {zugehörige analysierte Wortformen}

In der traditionellen Morphologie werden Wortformen in Morpheme zerlegt (segmentiert). Unter Morphemen versteht man die kleinsten, bedeutungstragenden Einheiten einer Sprache, aus denen die Wortformen zusammengesetzt sind. Wir unterscheiden bei den Morphemen zwischen Stammorphemen, im folgenden als *Lexeme* bezeichnet und den (Endungs-)*Morphemen*, [Her96], Seite 55:

> *„Lexeme und Morpheme sind die Elemente aus denen sich die Satzstrings aufbauen. Morpheme sind strukturell orientiert und eher für die syntaktische Struktur zuständig: Sie bilden kleinere, überschaubare Listen, die Listen sind abgeschlossen, sie werden höchstens über Jahrhunderte erweitert oder verändert. Lexeme sind auf den Außensinn orientiert: Sie bilden größere, oft unüberschaubare Listen; die Listen sind offen, sie können sich im Sprachwandel leichter erweitern und reduzieren.“*

So besteht die Wortform *springen* aus dem Lexem *spring* und dem Morphem *en*. Den umgekehrten Prozeß, der Zusammensetzung von Morphemen zu Wortformen, nennt man Konkatenation. Die Konkatenation eines Lexems mit verschiedenen Flexions- und Derivationsmorphemen ergibt Wortformen mit unterschiedlicher syntaktischer Kombinatorik. Zum Beispiel kann das Lexem *spring* sowohl mit dem Morphem *t* als auch mit *st* verbunden werden. Im ersten Fall ergibt sich eine Verbform, deren syntaktische Kombinatorik einen Nominativ der dritten Person Singular oder der zweiten Person Plural fordert, im zweiten Fall dagegen eine Verbform, die sich nur mit einem Nominativ der zweiten Person Singular verbindet.

### 5.2.2 Die lexikalischen Kategorien

Alle Grammatiken unterscheiden bestimmte Wortarten, Wortklassen oder lexikalische Kategorien. Die unterschiedlichen Definitionen beruhen i.a. auf einem der drei folgenden Verfahren:

1. Das flexematische Verfahren leitet aus verschiedenen Flexionsparadigmen die Wortklassen ab. Die Hauptschwäche dieses ansonsten sehr klaren Ver-

fahrens besteht darin, daß die Vielzahl der unveränderlichen Wörter nicht differenziert werden können.

2. Das semantische Verfahren unterstellt jeder Wortklasse per se eine bestimmte Bedeutung: Z.B. Nomina bezeichnen Dinge, Verben Vorgänge oder Zustände oder Adjektive Eigenschaften oder Beschaffenheiten.

3. Das distributionelle Verfahren klassifiziert die Wörter nach ihrer Umgebung, in der sie vorkommen können.

Es werden häufig Mischverfahren zur Klassifikation der Wörter vorgeschlagen, wie z.B. nach semantischen und distributionellen Gesichtspunkten [BS77] oder nach flexivischen, distributionellen und semantischen Merkmalen [Eng91].
Bei der Gliederung der lexikalischen Kategorien (Klassen) folgen wir der Definition und Einteilung von Heringer ([Her96], Seite 55):

> *Eine lexikalische Kategorie ist eine Menge von Lexemen, die sich syntaktisch analog verhalten, im Prinzip kommutieren. Nicht-Kommutation muß durch distributionelle Beschränkungen erklärt werden. Im Gegensatz zu den traditionellen Wortarten spielen dabei semantische Gesichtspunkte keine Rolle, wenngleich Lexeme gleicher Kategorie auch semantische Ähnlichkeiten haben dürfen.*

Ziel der Kategorisierung ist eine Zerlegung des Lexikons, so daß jedes Lexem in mindestens einer Kategorie vertreten ist. Heringer fordert weiterhin, daß die lexikalischen Kategorien durchlässig sind, also syntaktische Übertritte zugelassen werden [Her96]. Die lexikalischen Hauptklassen lauten[7]:

- Verb (V): Verben sind flektierbare Lexeme. Sie sind lokal definierbar durch die Verbmorpheme (VM). Die Verbalmorpheme sind subklassifiziert nach PERSON, NUMERUS, MODUS, semantisch auch nach TEMPUS. Die Verben selbst lassen sich differenziert subklassifizieren in Vollverben mit ihren (syntaktischen) Valenzen (V_val) und Verben, die Verbalkomplexe regieren. Beispiele: $\{arbeit, atm, brenn, frier, \dots\}$

[7] Die Beschreibungen sind aus [Her96] entnommen bzw. aus diesen Ausführungen abgeleitet.

- Nomen (N): Nomina sind deklinierbar und lokal definierbar durch die Nominalmorpheme (NM). Die NM sind subklassifiziert nach Deklinationsklasse, Numerus und Genus. Die wichtigsten Subklassen der Nomina sind die Substantive N_sub und die Eigennamen N_ne. Heringer [Her96] faßt außerdem die Pronomina unter der Klasse Nomen zusammen. Hier weichen wir von der Einteilung Heringers ab, da die starke Subklassifizierung der Pronomina eine eigene Klasse motiviert.
  Beispiele: $N_{sub} : \{Abend, Berg, Betrieb, Einsatz, \dots\}$

- Pronomen (PR): Pronomina sind Wörter, die alleine eine Nominalphrase bilden können und nicht zusammen mit Determinierer und attributiven Adjektiven vorkommen. Die Klasse der Pronomina ist stark subklassifiziert. So unterscheiden Thielen und Schiller [TS96], ob die Pronomina nomenbegleitende (attributierende) oder NP-ersetzende (substituierende) oder auch adverbiale Funktionen haben. Bei Personalpronomina, ob sie stets irreflexiv oder auch reflexiv gebraucht werden, oder nur reflexiv benutzt werden und damit zu den echten Reflexivpronomen zählen.
  Beispiele: $PR_{prs} : \{du, dich, er, es, ich, \dots\}$

- Adjektiv (A): Adjektive sind deklinierbar, bilden jedoch keine Deklinationsklassen. Die Adjektivmorpheme (AM) überlappen sich mit den NM und den Determiniermorphemen, weniger mit den VM. Im Gegensatz zu den Nomen, wo nur Untermengen der NM auf jedes Nomen anwendbar sind, ist auf jedes Adjektiv das vollständige AM-Paradigma anwendbar. Die AM sind subklassifiziert nach NUMERUS, KASUS und GENUS. Adjektive haben sowohl Eigenschaften von Nomen, als auch von Verben. Sie variieren nach Kasus und Genus, sie haben eine Valenz und weisen Kasus auf.
  Beispiele: $\{abstrus, ahnungsvoll, aktiv, allein, allgemein, \dots\}$

- Präposition (P): Präpositionen sind nicht flektierbar. Sie sind zweistellige Bindewörter, die N regieren und ihnen einen Kasus zuweisen. Die Subklassifikation erfolgt dabei über den regierten Kasus.
  Beispiele: $\{ab, auf, hinter, neben, über, \dots\}$

- Äquation (Ä): Äquationen sind nicht flektierbar. Sie regieren N und bilden Äquationsphrasen. Ä weisen den regierten N keinen Kasus zu, sie sind durchlässig und weisen den Kasus einem anderen N weiter.
  Beispiele: $\{als, wie, für, statt, außer, \dots\}$

- Determinierer (D): Determinierer sind deklinierbar. Die DM bilden eine Untermenge der AM. Die Kategorie D zerfällt in mehrere Subklassen, wobei zwei traditionell als Artikel bezeichnet werden, sie verhalten sich syntaktisch wie die anderen Determinierer.
  Beispiele: $\{das, dem, den, der, des, die, \dots\}$

- Adverbien (ADV): Adverbien sind nicht flektierbar (können jedoch evtl. gesteigert werden). Adverbien sind entweder von V oder ADV dependent.
  Beispiele: $\{sehr, teilweise, fast, besonders, \dots\}$

- Konjunktionen (KON): Konjunktionen sind nicht flektierbar. Sie verbinden gleichartige und gleichstufige syntaktische Einheiten. Sie sind zweistellig und syntaktisch gesehen symetrisch, d.h. wenn $x_1$ KON $x_2$ wohlgeformt ist, dann auch $x_2$ KON $x_1$.
  Beispiele: $\{und, aber, oder, \dots\}$

- Subjunktion (SUB): Subjunktionen sind nicht flektierbar. Wie P und KON sind es zweistellige Bindewörter, die Klauseln und propositionale Einheiten verschiedener Stufen verbinden.
  Beispiele: $\{daß, ob, obwohl, weil, während\}$

- Partikel (PTL): Partikel sind nicht flektierbar. Sie dominieren nicht und sind syntaktisch ungebunden.
  Beispiele: $\{auch, besonders, bloß, \dots\}$

- Satzwort (SZW): Satzwörter sind syntaktisch völlig selbstständig und bilden Einwortsätze.
  Beispiele: $\{bitte, eins, hallo, \dots\}$

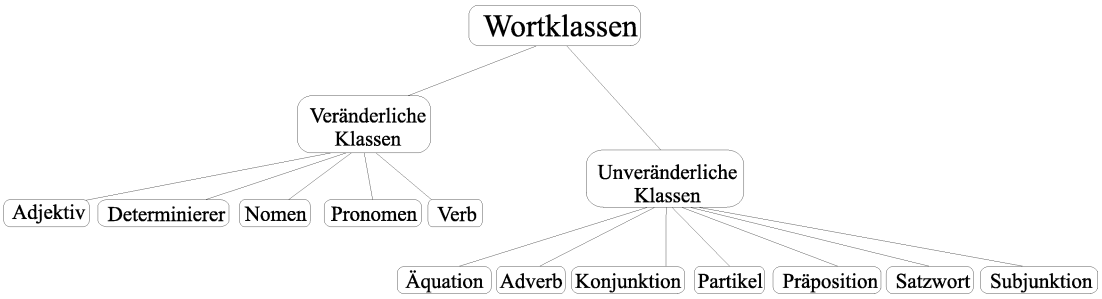


Abbildung 5.6: Realisierte Wortklassen in der LKB

Die Speicherung des Faktenwissens erfolgt in einer relationalen Datenbank, deren logische Struktur ich im folgenden in Form von Entity-Relationship-Modellen[8] (ERM) erläutern werde.
Bei der Modellierung der Wortklassen unterscheiden wir zwischen veränderlichen und unveränderlichen Klassen. Bei den unveränderlichen Klassen werden lediglich zwei Wortklassen gesondert behandelt:

1. Präpositionen, da sie in einer *Artikelform* vorkommen, die eine Kurzform von Artikel und Präposition darstellt.

2. Adverbien, da eine Untermenge der Adverbien steigerbar ist.

Sämtlichen Hauptentitäten (im Sinne von Wortklassen) sind neben den wortklassenspezifischen Merkmalen auch systeminterne Merkmale zugeordnet. Die systeminternen Merkmale geben Auskunft über die Art der Speicherung des lexikalischen Eintrags (VOLLF) und ob ein neuer Eintrag in der Vollformendatenbank generiert werden muß (GFLAG). Weiterhin wird die Herkunft des lexikalischen Eintrags in dem Merkmal E_ART festgehalten. Hierbei unterscheiden wir zwischen manuell eingetragenen oder inspizierten Einträgen, abgeleiteten Informationen oder automatisch erzeugten Einträgen. Außerdem können in einem Merkmal FREITEXT einer Entität beliebige Kommentare zugeordnet werden. Der Primärschlüssel sämtlicher Wortklassen ist durch eine Integer-Zahl realisiert, die vom System automatisch vergeben wird. In den folgenden Beschreibungen werden wir ausschließlich auf die wortklassenspezifischen Merkmale eingehen.

[8] Erläuterungen zur ER-Modellierung finden sich in Anhang B.

#### 5.2.2.1 Wortklasse Verb

Bei der Wortklasse der Verben wird in der Faktenbasis neben den morphosyntaktischen Informationen auch Wissen über Valenzstrukturen auf syntaktischer und semantischer Ebene abgespeichert. Abbildung 5.7 zeigt die logische Datenbankstruktur für das Faktenwissen dieser Wortklasse. Das morphosyntaktische Wissen ist in der Haupttabelle LKB_VERB und je nach Art des Verbs in STARK_VERB oder VOLL_VERB realisiert.

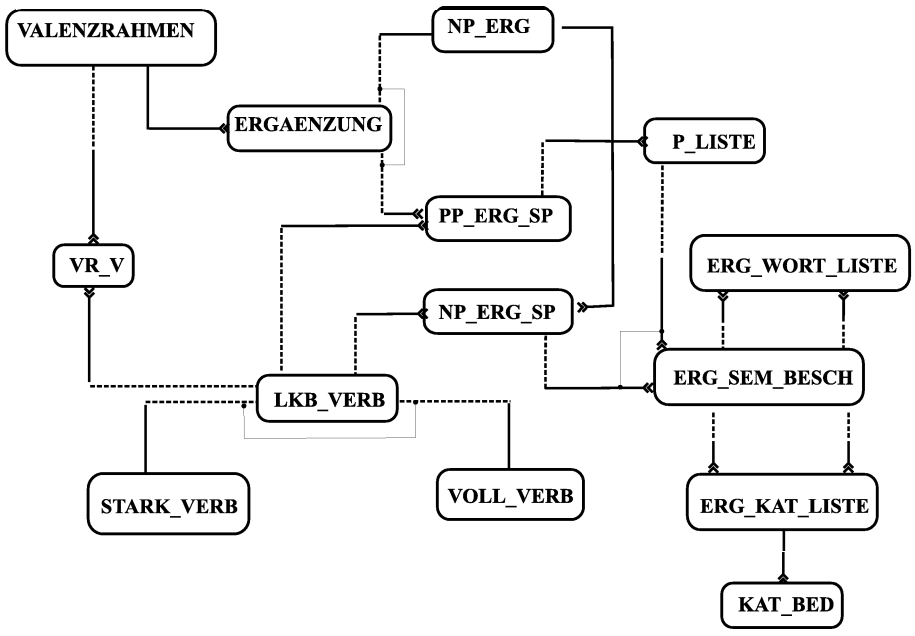


Abbildung 5.7: Entity-Relationship-Modell der Wortklasse Verb

In der Abbildung 5.7 sind weiterhin sowohl die Entitäten und ihre Beziehungen für die allgemeinen syntaktischen Valenzrahmen modelliert als auch diejenigen für die Valenzbeschreibungen auf semantischer Ebene. Hierbei sind die Beschreibungen der syntaktischen Valenzstrukturen als Inter-Lexem Informationen zu verstehen und die Einschränkungen auf semantischer Ebene als Intra-Lexem Informationen modelliert, die einem spezifischen lexikalischen Eintrag direkt zugeordnet sind. Die Entität VALENZRAHMEN beschreibt die allgemeinen Merkmale einer syntaktischen Valenzstruktur in Form eines Bezeichners und der Angabe der Anzahl der Positionen, etwa k. Ein syntaktischer Valenzrahmen hat 1 ... k Ergänzungen,

die durch die Ergänzungsart (wie etwa Nominalphrase oder Präpositionalphrase) und ihre Position charakterisiert sind. Handelt es sich um eine Nominalphrase, so hat diese Ergänzung noch eine Beziehung zu der Entität NP_ERG, die hauptsächlich durch das Merkmal Kasus charakterisiert ist. Die Intra-Lexem Informationen betreffen die Art der Ergänzung, d.h. ob eine Ergänzung obligatorisch oder fakultativ ist, und die Beschreibung der semantischen Einschränkungen von Positionen in Form von Listen von kategoriellen Bedeutungen oder auch Wortlisten in Form von Nomina. Bei präpositionalen Ergänzungen können eine oder mehrere Präpositionen angegeben werden, die wiederum eine Nominalphrase regieren. Auch für diese NP können semantische Restriktionen formuliert werden. Morphosyntaktisch wird ein lexikalischer Verbeintrag durch die folgenden Informationen (Merkmale der Entität LKB_VERB) beschrieben:

- Form des Infinitivs (INF);
- Konjugationsart (KONJ);
- Bildung der zusammengesetzten Formen (PASS und PERF);
- Art der Partizip II Form (GE_PII);
- adjektivische Verwendung (ADJ).

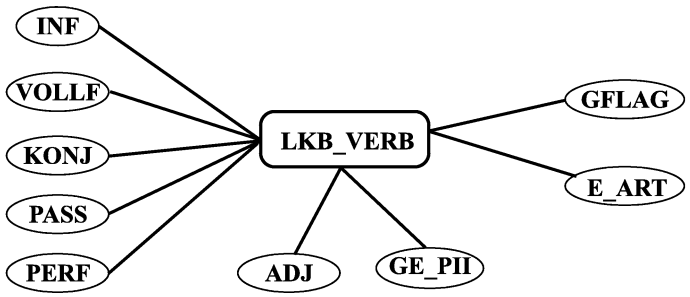


Abbildung 5.8: Merkmale der Entität LKB_VERB

Das verbale Paradigma besteht im Deutschen aus 29 Formen. Bei der Konjugation der Verben unterscheiden wir zwischen den schwachen (oder regelmäßigen) und

den starken Verben[9]. Sämtliche Formen der schwachen Verben lassen sich aus der Infinitivform ableiten, bei den starken Verben reichen in der Regel die Angabe von 4 Stammformen:

- Infinitiv: enthält den Stammvokal für die 1. Person Singular und den gesamten Plural Präsens.
- 3. Person Singular Präsens: enthält den Stammvokal für die 2. Pers. Singular Präsens.
- 3. Person Singular Präteritum: Stammvokal gilt für das gesamte Präteritum.
- Partizip II

Betrachtet man die Verben, die i.a. als unregelmäßige Verben behandelt werden, so kann man feststellen, daß zur Ableitung aller Formen die Hinzunahme einer fünften Stammform (Konjunktiv II) häufig ausreicht. Diese vier bzw. fünf Stammformen sind als Merkmale der Entität STARK_VERB realisiert.

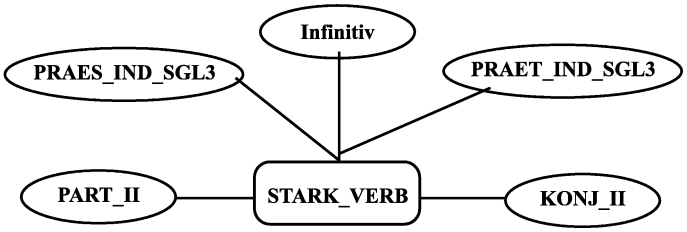


Abbildung 5.9: Merkmale der Entität STARK_VERB

Falls nicht alle Formen eines Verbs mittels der fünf Stammformen und der realisierten Regeln generiert werden können, so werden in einer Entität VOLL_VERB sämtliche Formen realisiert.

[9]Der in der Literatur häufig zu findenden Unterscheidung zwischen schwachen, starken und unregelmäßigen Verben wird in unserem System bei der Operationalisierung nicht gefolgt. Die unregelmäßigen Verben werden in der Klasse der starken Verben mitbehandelt.

#### 5.2.2.2 Wortklasse Adjektiv

Das Modell der Wortklasse Adjektiv besteht aus einer Hauptentität LKB_ADJEKTIV und Entitäten zur Beschreibung der syntaktischen und semantischen Valenz[10]. Eine Entität der Klasse LKB_ADJEKTIV ist durch die folgenden Merkmale gekennzeichnet:

- Ob ein Adjektiv attributiv und/oder prädikativ verwendet werden kann, wird durch die beiden Merkmale PRAEDIKATIV und ATTRIBUTIV beschrieben, deren Wertebereiche jeweils die Menge $\{J, N\}$ ist.
- Die Anzahl der Wortformen eines adjektivischen Paradigmas ist abhängig davon, ob das betreffende Adjektiv deklinierbar (DEKL) und steigerbar (STEIGERB) ist.
- Das gesamte Paradigma läßt sich aus der Kenntnis einer Positiv- (POS), einer Komparativ- (KOMP) und einer Superlativform (SUP) ableiten.
- Das Adjektiv kann weiterhin nach den Bedeutungsklassen (BED):
  - quantifikative Adjektive,
  - referentielle Adjektive,
  - klassifikative Adjektive,
  - Herkunftsadjektive

  subklassifiziert werden.

Wenn es sich um ein „vollständiges“ adjektivisches Paradigma handelt, besteht dieses aus 144 Formen. Da im Gegensatz zu den Klassen Verb oder Nomen, die Klasse der Adjektive keine (bzw. kaum) Irregularitäten bei der Generierung sämtlicher Wortformen aufweist, wurde bei dieser Klasse auf eine mögliche Speicherung sämtlicher Wortformen in der lexikalischen Wissensbasis verzichtet.

[10]Da die Modellierung der Valenz analog zur Valenz der Klasse der Verben erfolgt, verzichte ich an dieser Stelle auf eine weitere Beschreibung

#### 5.2.2.3 Wortklasse Nomen

Neben den Verben ist die Wortklasse der Nomina die Wortklasse mit dem reichsten Formeninventar für die Deklination. Auch bei dieser Wortklasse ist es grundsätzlich möglich, einen lexikalischen Eintrag in der Wissensbasis in Form eines Vollformeneintrags zu realisieren. Die Art der Deklination ist durch die Auswahl der Deklinationsschemata für Singular und Plural getrennt festgelegt. Falls sich die Stammformen in Singular und Plural unterscheiden, wird ein lexikalischer Nomeneintrag durch die Angabe der Wortformen Nominativ Singular und Plural charakterisiert. Eine notwendige Angabe ist weiterhin das Genus.

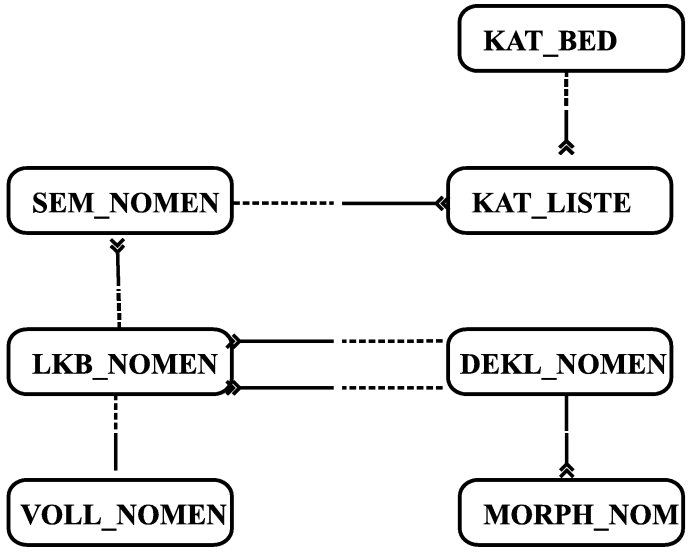


Abbildung 5.10: ER-Modell der Wortklasse Nomen

Die Beziehungen zur Deklinationsentität DEKL_NOM sind gesperrt, falls es sich bei der Entität um ein nominalisiertes Adjektiv handelt. In diesem speziellen Fall werden sämtliche Flexionsformen unter Berücksichtigung des variablen Genus mittels Regeln erzeugt.
Die Verwaltung der Deklinationsschemata kann ohne konkrete Verbindung zu einer existierenden Entität Nomen durchgeführt werden. Wir haben als Voreinstellung die drei Singulartypen Null-Singular, (e)s-Singular und (e)n-Singular und die fünf Pluraltypen e-Plural, Null-Plural, (e)n-Plural, er-Plural und s-Plural im

System integriert[11].
Einem nominalen Eintrag können eine oder mehrere semantische Beschreibungen zugeordnet werden, wobei auf der ersten Ebene in SEM_NOMEN zwischen konkret und abstrakt unterschieden wird und auf der zweiten Ebene in KAT_LISTE eine oder mehrere kategorielle Bedeutungen zugeordnet werden. Eine nominale Entität wird durch eine weitere semantische Unterscheidung zwischen Gattungsnamen, Stoffnamen und Eigennamen charakterisiert. Dieses Attribut ist der Hauptentität LKB_NOMEN zugeordnet.

#### 5.2.2.4 Wortklasse Determinierer

Die Informationen der Klasse der Determinierer sind in einer Entität zusammengefaßt. Im Singular werden die Formen nach Genus unterschieden, weiterhin wird angegeben, ob es sich um einen definiten oder indefiniten Determinierer handelt. Die Klasse der Determinierer ist stark subklassifiziert, wobei folgende Werte für das Subklassen-Merkmal zugelassen sind:

- Definite Artikel (det-def);
- Demonstrativ-Determinierer (det-dem);
- Possessiv-Determinierer (det-pss);
- Interrogativ-Determinierer (det-int);
- definite Quantoren (qnt-def), indefinite Quantoren (qnt-ind) und Negationen (qnt-neg).

Bei Demonstrativ- und Possessiv-Determinierer wird durch das Merkmal PRON_GEN angegeben, ob die entsprechenden Pronomenformen automatisch generiert werden sollen.

[11] Die Charakterisierung erfolgt nach der Endung im Genitiv. Die Klammern geben entsprechende Subdeklinationsschemata an.

#### 5.2.2.5 Wortklasse Pronomen

Die Informationen der Wortklasse Pronomen sind auf die beiden Entitäten LKB_-PRONOMEN und PRONFORMEN verteilt. Da die Klasse der Pronomen stark subklassifiziert ist, haben wir in Form von logischen Sichten[12] drei Gruppen von Pronomen-Subklassen mit gleichen Merkmalen erzeugt.

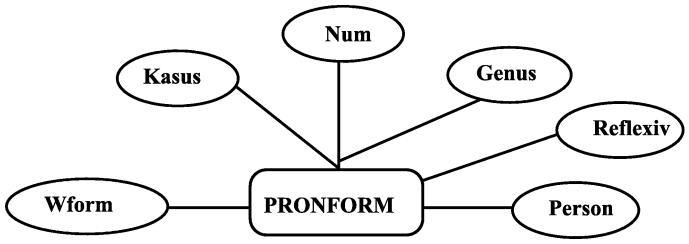


Abbildung 5.11: Merkmale der Entität PRONFORM

Lediglich die Subklasse der Personalpronomen realisiert sämtliche Merkmale der Entität PRONFORMEN (siehe Abbildung 5.11). Falls die Form eines Personalpronomens auch reflexiv verwendet werden kann ($Reflexiv = J$), so wird eine Entität der Klasse Reflexivpronomen (automatisch) erzeugt.
Bei der zweiten Gruppe von Pronomen handelt es sich um die Subklassen der einfachen, definiten und indefiniten Pronomen sowie der Interrogativ- und Demonstrativpronomen. Hier werden neben dem Merkmal Wform noch Kasus, Num und Genus zur Beschreibung benötigt.
Die letzte Gruppe wird aus den beiden Subklassen Possessiv- und Reflexivpronomen gebildet, bei deren Beschreibung zusätzlich zu den Merkmalen der zweiten Gruppe noch Person realisiert ist.

#### 5.2.2.6 Wortklasse Präposition

Das Modell zur Beschreibung der Präpositionen besteht aus der Hauptentität der LKB_PRAEP sowie der Entität PRAEP_KASUS, in der die möglichen Kasi

[12]Bei den sogenannten Views handelt es sich bei der späteren Implementierung nicht um physikalische Datenbanktabellen, sondern um eine logische Verknüpfung von Merkmalen einer oder mehrerer Tabellen (bzw. Entitäten).

angegeben werden, und evtl. einer Artikelform, die eine Kurzform von Artikel und Präposition bezeichnet (z.B. „übers“ statt über das). Das Merkmal P_ART gibt die möglichen Stellungen an:

- vorangestellte Präpositionen (praep),
- nachgestellte Präpositionen (post),
- vor- oder nachgestellte Präpositionen (praep/post),
- das Nomen umschließende Präpositionen (zirk).

Außerdem gibt es noch die Subklasse der präpositionalen Pronomen, die nach ihrer Bedeutung in lokale, temporale oder direktionale Pronominalpräpostionen unterteilt werden.

#### 5.2.2.7 Wortklasse Adverb

Ein Vertreter der Klasse Adverb ist vollständig durch die Merkmale der Entität LKB_ADVERB beschrieben. Falls es sich um ein steigerbares Adverb handelt, werden neben der einfachen Wortform zusätzlich die Komparativ- und Superlativ-Form in den entsprechenden Merkmalen eingetragen. Die Adverbien sind in die Subklassen der modifikativen (mod), graduativen (grd) und satzmodifizierenden (stz) Adverbien unterteilt.

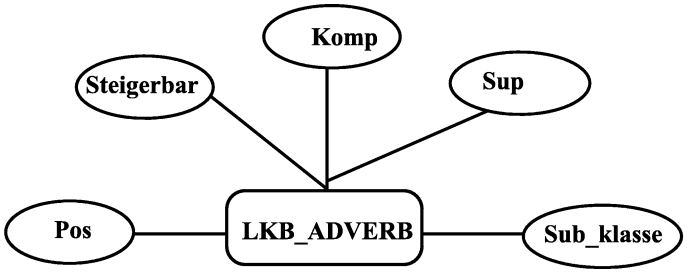


Abbildung 5.12: Merkmale der Entität LKB_ADVERB

#### 5.2.2.8 Sonstige Wortklassen

Die restlichen (unveränderlichen) Wortklassen Äquation, Konjunktion, Partikel, Satzwort und Subjunktion sind in einer Entität zusammengefaßt. Zu den Merkmalen dieser Entität gehören neben der Wortform die Wortklasse sowie evtl. eine Subklasse. Wie bei allen Hauptentitäten wird als Primärschlüssel eine eindeutige interne Integer-Zahl vergeben. Wir gehen davon aus, daß das Paar Wortform/Wortklasse jeweils nur einmal vorkommt, so daß wir einen eindeutigen Schlüssel auf diesem Merkmalspaar definiert haben.

## 5.3 Akquisitionskomponente

Die Akquisitionskomponente unterstützt die Anwender bei der Erfassung von lexikalischem Wissen (bzw. lexikalischen Einträgen). In dieser Unit sind die morphologischen Regeln zur Erzeugung sämtlicher Wortformen eines lexikalischen Eintrags integriert.

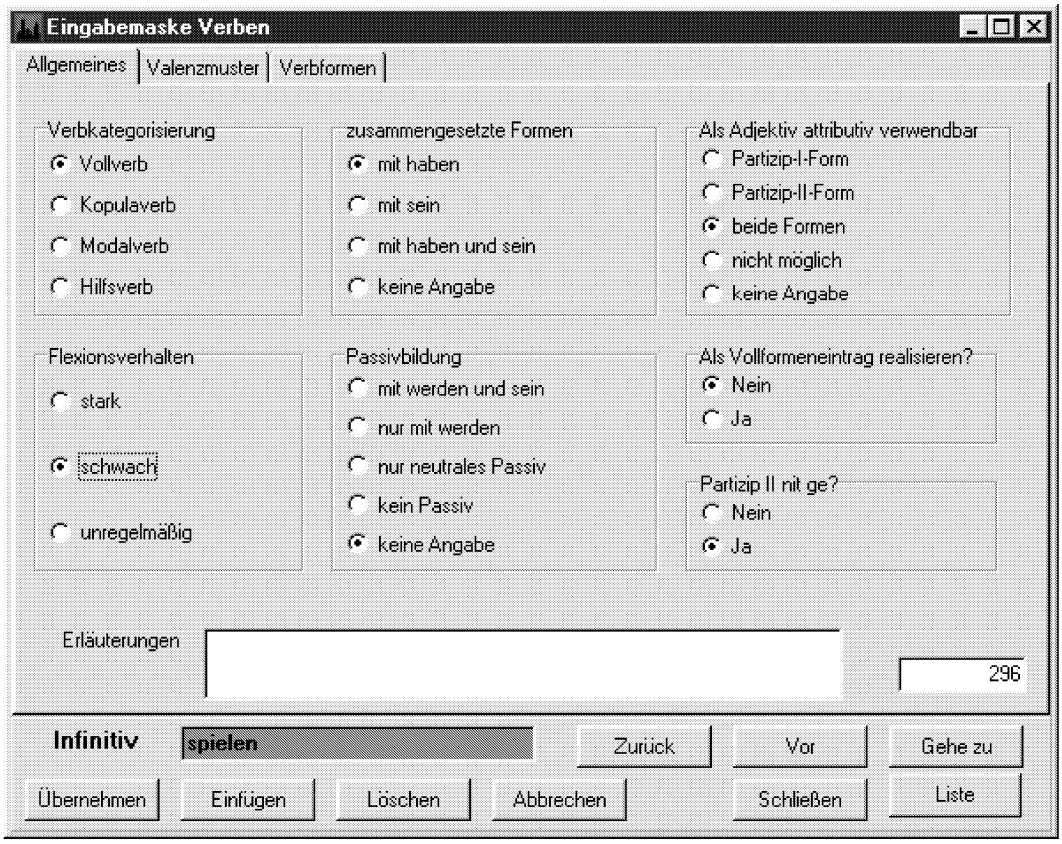


Abbildung 5.13: Akquisitionsfenster für die Wortklasse Verb

Die Fensteroberfläche ist in der Regel als mehrseitiges Formular aufgebaut, wobei auf der ersten Seite die allgemeinen Informationen eingetragen bzw. angezeigt werden, die zur Erzeugung des gesamten Paradigmas notwendig sind. Die weiteren Formulare behandeln spezifischere Einträge wie Valenzmuster oder semantische Beschreibungen. Wenn es sich um eine Wortklasse handelt, bei er ein Eintrag als Vollformeneintrag in der LKB spezifiziert werden kann, so können auf einer

weiteren Seite sämtliche Wortformen ediert werden.

Betrachten wir beispielsweise die Klasse der Verben (Abbildung 5.13), so werden auf der Startseite der Infinitiv und Informationen zu zusammengesetzten Formen etc. eingetragen. Handelt es sich um ein starkes oder unregelmäßiges Verb, so schaltet das System automatisch auf die dritte Seite um, in der die weiteren Stammformen eingetragen werden.

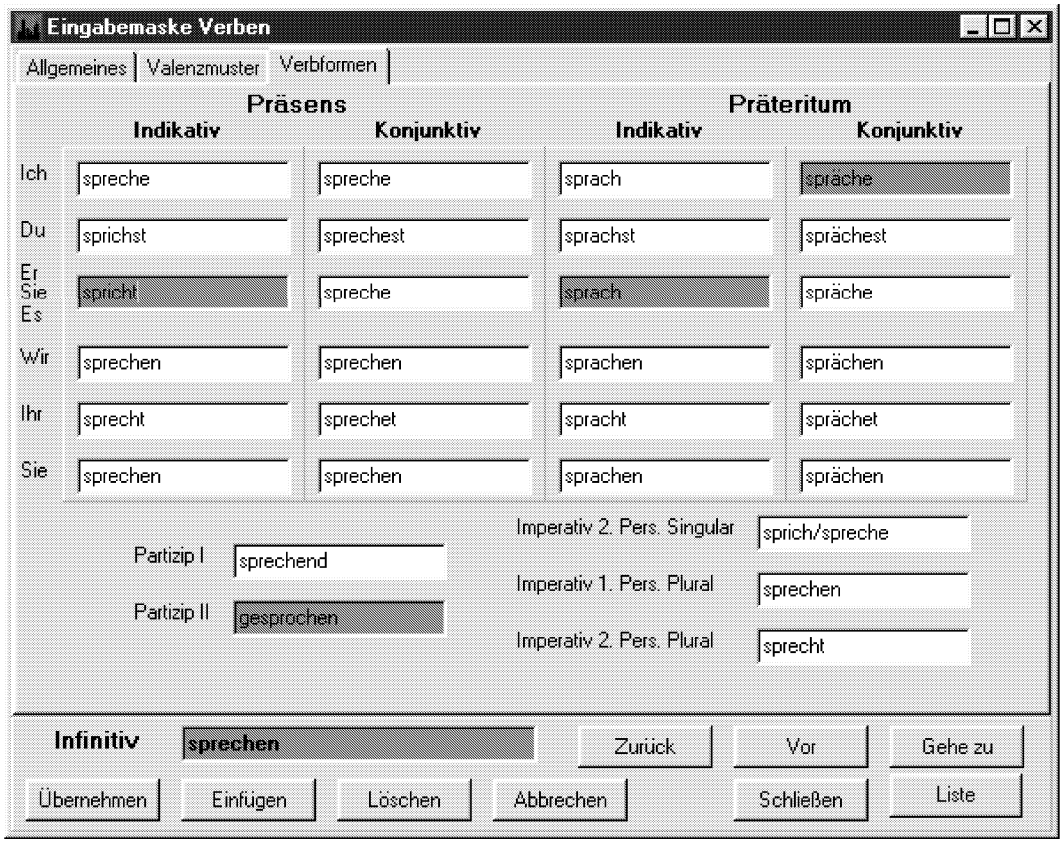


Abbildung 5.14: Akquisitionsfenster für die Wortklasse Verb - Morphologie

Nachdem die letzte Stammform akquiriert wurde, generiert das System sämtliche Wortformen des verbalen Paradigmas. Die Formengenerierung basiert dabei auf den folgenden Regeln[13]:

[13]Die Regeln sind untergliedert nach Regeln der Gegenwarts- ($GR_i$) und Vergangenheitsformen ($VR_j$).

GR1 e-Ausfall vor *l*:

Bei dem Ableitungssuffix *eln* wird in der 1. Pers. Singular das *e* unterdrückt. (ich handle - er handelt)

Außerdem wird das *e* im Konjunktiv bei den Singularformen und bei der 2. Pers. Plural unterdrückt.

GR2 e-Ausfall nach *l*:

Bei Verben, deren Stamm auf Konsonant + eln endet, wird nach dem *l* in der 1. Pers und 3. Pers Plural das *e* unterdrückt. Ebenfalls bei den gesamten Pluralformen des Konjunktiv I.

GR3 e-Ausfall bei „s-Laut“:

Bei „s-Laut“ (s, ß, x, z) wird das *s* in der 2. Pers. Singular unterdrückt. (du ließt, du hext)

GR4 e-Einschub bei Konsonant + m/n:

Bei Verben, deren Stamm auf Konsonant + m/n endet, wird in der 2. und 3. Pers. Singular und in der 2. Pers. Plural vor der Finitendung ein *e* eingeschoben. (du atm est, er ebn et)

Ausnahmen: nicht bei Stammauslaut *lm, ln, rm, rn.*

GR5 e-Einschub nach Dental:

Bei Verben, deren Stamm auf *d* oder *t* endet, wird in der 2. und 3. Pers. Singular und in der 2. Pers. Plural vor der Finitendung ein *e* eingeschoben.

GR6 ss-ß:

Bei schwachen Verben, deren Stamm auf *ss* endet, wird in der 2. Pers und 3. Pers Singular und in der 2. Pers Plural *ss* durch *ß* ersetzt. Weiterhin greift für die 2. Pers Singular Regel GR3.

GR7 e-Ausfall nach Konsonant + ern:

Bei Verben, deren Stamm auf Konsonant + ern endet, wird nach dem *r* in der 1. Pers und 3. Pers Plural das *e* unterdrückt. Ebenfalls bei der 2. Pers Sing. Konjunktiv I und den gesamten pluralen Konjunktivformen.

GR8 optionaler e-Ausfall vor r:
Bei Verben, deren Stamm auf Konsonant (außer {g, h, l, n}) + ern endet, kann vor dem *r* das *e* bei der 1. Pers. Singular Präsens und 1. Pers. und 3. Pers. Sing Konjunktiv (optional) ausfallen.

GR9 Modalverben (außer *brauchen*) und das Verb *wissen* haben in der 3. Pers. Singular kein *t*.

VR1 Bei starken Verben, deren Stamm auf *s* oder *ß* endet, wird in der 2. Pers. Singular ein *e* eingeschoben.

VR2 Starke Verben, deren Stamm auf *d* oder *t* endet, schieben bei der 2. Pers. Singular und Plural immer ein *e* ein.

VR3 Schwache Verben, deren Stamm auf *d* oder *t* endet, schieben vor dem t-Suffix ein *e* ein.

VR4 Bei schwachen Verben, deren Stamm auf Konsonant + m/n endet, wird vor dem t-Suffix ein *e* eingeschoben.
Ausnahme: nicht bei Stammendungen *lm, ln, rm, rn.*

VR5 Bei schwachen Verben, deren Stamm auf *ss* endet, wird *ss* durch *ß* ersetzt.

In dem Fall, daß die Regeln nicht alle Formen korrekt generieren, können während der Akquisition Formen manuell nachbearbeitet werden. Das System schaltet in diesem Fall automatisch auf den Modus Vollform um.

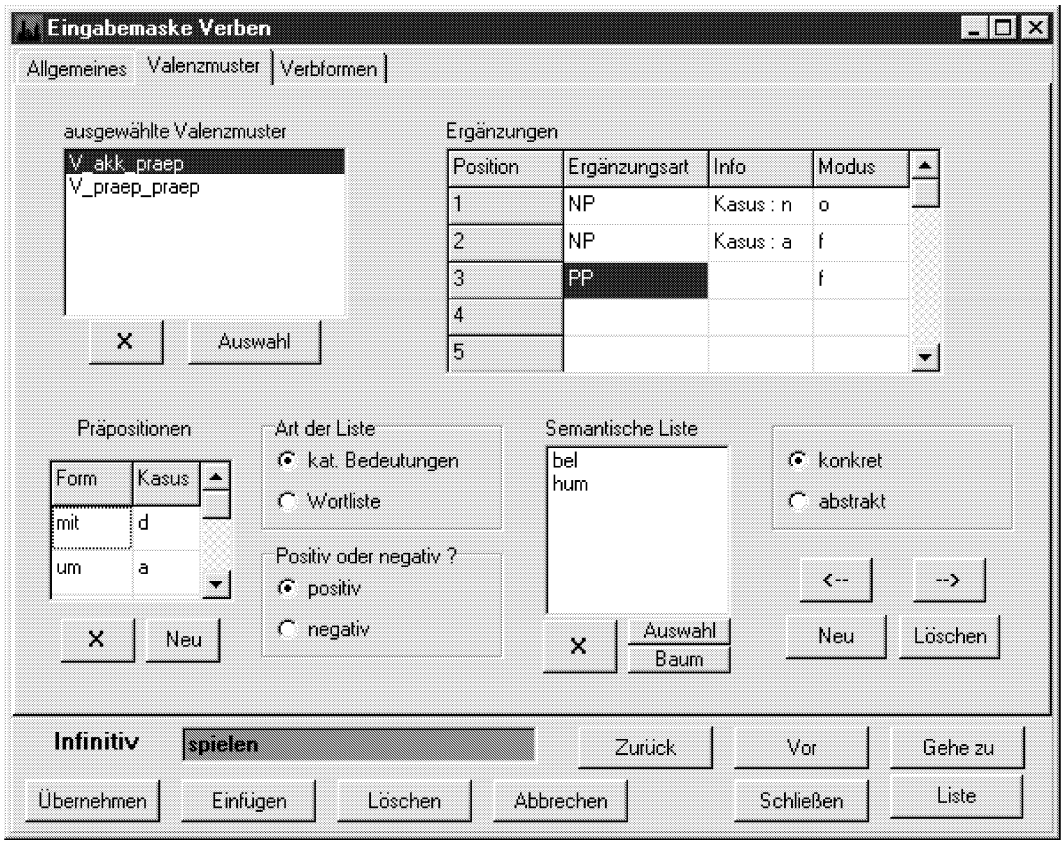


Abbildung 5.15: Akquisitionsfenster für die Wortklasse Verb - Valenzbeschreibung

Abbildung 5.15 zeigt das Akquisitionsfenster zur Beschreibung von Verbvalenzen. In dem oberen Teil können die vorhandenen syntaktischen Valenzmuster ausgewählt werden, im unteren Teil werden die semantischen Einschränkungen spezifiziert. Wir sehen eine Spezifikation für das Verb *spielen*. Es sind zwei dreistellige Valenzmuster eingetragen worden, wobei zur Zeit die dritte Stelle des Valenzmusters V_akk_praep ausgewählt ist. Bei dieser fakultativen Ergänzung wurden zwei Präpositionen angegeben, wobei die Präposition „mit“ syntaktisch auf den Kasus Dativ und semantisch auf etwas Konkretes, Belebtes, Menschliches eingeschränkt wurde.

## 5.4 Vollformengenerierungskomponente

Die Generierungskomponente erzeugt, ausgehend von den Einträgen in der lexikalischen Wissensbasis, zu jeder möglichen Wortform einen Eintrag in der Vollformendatenbank mit den entsprechenden morphosyntaktischen und semantischen Informationen. Dabei sind die Einträge allgemein wie folgt kodiert:

Wortform|Grundform|POS|POS-SUB|INFORMATION

An der ersten Position ist die Wortform aufgeführt, es folgt die Grundform. Die nächsten beiden Positionen kodieren die Wortklasse (POS)[14] und evtl. eine Subklasse (POS-SUB). Anschließend folgt ein Informationsstring, dessen Kodierung abhängig von der Wortklasse bzw. Subwortklasse ist. Je nach Wortklasse handelt es sich um morphologische und semantische Beschreibungen, bis hin zu Valenzbeschreibungen auf syntaktischer und semantischer Ebene.
Die folgende Tabelle gibt eine Übersicht über die morphologischen Merkmale, ihre Kodierung und ihr Vorkommen bei den verschiedenen Wortklassen.

Tabelle 5.4: Morphologische Merkmale und ihre Kodierung

| | | | |
|---|---|---|---|
| **Kasus** | nominativ | Nomen, Adjektiv, Determinierer, Pronomen, Präpositionen | n |
| | akkusativ | | a |
| | dativ | | d |
| | genitiv | | g |
| | unspezif. | | u |
| **Person** | 1 | Pronomen, Verb | 1 |
| | 2 | | 2 |
| | 3 | | 3 |
| | unspez. | | 0 |
| **Numerus** | singular | Adjektive, Determinierer, Pronomen, Verb | e |
| | plural | | m |

[14] Part of Speech

| | | | |
|---|---|---|---|
| **Genus** | fem | Adjektive, Determinierer, Pronomen, Nomen | F |
| | mask | | M |
| | neut | | N |
| | unspez. | | U |
| **Definitheit** | definit | Determinierer, Nomen | D |
| | indefinit | | o |
| | nullartikel | | z |
| | unspezif. | | * |
| **Flexion** | stark | Adjektiv | h |
| | schwach | | w |
| | gemischt | | t |
| | prädikativ | | v |
| **Steiger-ungsgrad** | positiv | Adjektiv, Adverb | p |
| | komparativ | | k |
| | superlativ | | s |
| **Tempus** | Präsens | Verb | G |
| | Präteritum | | V |
| | unspez. | | O |
| **Modus** | indikativ | Verb | i |
| | konjunktiv | | c |
| | imperativ | | b |
| **infinite Formen** | partizip 1 | Verb | E |
| | partizip 2 | | Z |
| | infinitiv | | I |

### 5.4.1 Kodierung der Valenz

Bei der Klasse Adjektiv und Verb werden Informationen über Valenzbeschreibungen kodiert, die Auskunft über syntaktische und semantische Einschränkungen

geben.

Tabelle 5.5: Valenzbeschreibung bei den Wortklassen Adjektiv und Verb

| | | |
|---|---|---|
| VR | ::= | ANZAHL.1:ERG1. ... ANZAHL:ERG |
| ERG | ::= | ART.PHRASE |
| ERG1 | ::= | o.np.n.SEMANTIK |
| ART | ::= | o \| f |
| PHRASE | ::= | NP \| PP \| AEP \| ... |
| NP | ::= | np.KASUS.SEMANTIK |
| PP | ::= | pp.PRAEPL |
| KASUS | ::= | n \| a \| g \| d \| u |
| SEMANTIK | ::= | POSL;NEGL |
| POSL | ::= | $< s >$ SEM1/SEML $< /s >$ SEM \| |
| | | $< w >$ WLISTE $< /w >$ SEM \| * |
| NEGL | ::= | $< s >$ SEM1/SEML $< /s >$ SEM \| |
| | | $< w >$ WLISTE $< /w >$ SEM \| * |
| SEM | ::= | $< s >$ SEM1/SEML $< /s >$ SEM \| |
| | | $< w >$ WLISTE $< /w >$ SEM \| $\epsilon$ |
| SEM1 | ::= | k \| a |
| SEML | ::= | KATB KLISTE \| * |
| KLISTE | ::= | ,KATB KLISTE \| $\epsilon$ |
| KATB | ::= | bel \| hum \| ... |
| WLISTE | ::= | N NL |
| NL | ::= | ,N NL \| $\epsilon$ |
| N | ::= | Frau \| Apfel \| Haus \| ... |
| PRAEPL | ::= | $< p >$ P.KASUS.SEMANTIK PL \| * |
| PL | ::= | :P.KASUS.SEMANTIK PL \| $< /p >$ |
| P | ::= | ab \| von \| ... |
| ANZAHL | ::= | 1 \| $\cdots$ \| 9 |

Die Informationen betreffen den syntaktischen Rahmen und die semantischen Einschränkungen auf den einzelnen Positionen. Bei den semantischen Einschränkungen handelt es sich um semantische Merkmale in Form von kategoriellen Bedeutungen oder sogar um die Angabe bestimmter Wörter. Es wird grundsätzlich zwischen möglichen (POSL) und ausgeschlossenen Merkmalen (NEGL) bzw. Wörtern unterschieden.

**Beispiel 5.1 (Valenzbeschreibung)**
*Dem Verb „essen“ ordnen wir zwei veschiedene Valenzmuster zu: es kann an der zweiten Position eine Nominalphrase im Akkusativ als Ergänzung realisiert sein, oder die zweite Position ist durch eine Präpositionalphrase im Dativ besetzt. Es handelt sich bei der zweiten Position um eine fakultative Ergänzung.*

1. *< vr >2.1:o.np.n.< s >k/bel,hum< /s >;*.*
   *2:f.np.a.< s >k/bel,plant< /s >;*< /vr >*

2. *< vr >2.1:o.np.n.< s >k/bel,hum< /s >;*.*
   *2:f.pp.< p >von.d.< s >k/bel,plant< /s >;*< /p >< /vr >*

*Jeder Valenzrahmen ist in die Zeichenkette < vr > < /vr > eingeschlossen.*

### 5.4.2 Wortklassenspezifische Kodierung

Der Informationsstring ist abhängig von der Wortklassenzugehörigkeit[15]. Ich möchte hier exemplarisch die Kodierung der Klasse Nomen und Verben behandeln.
Bei einem Eintrag der lexikalischen Klasse Nomen werden die morphosyntaktischen und semantischen Informationen in der folgenden Form kodiert:

$$M_1\text{: } \ldots \text{ : } M_K.\text{SEM1}.S_1\text{: } \ldots \text{ : } S_L$$

Ein morphologisches Merkmalsbündel $M_i$ besteht aus den Informationen zu Kasus, Numerus, Genus und Definitheit. Mögliche Alternativen werden durch Dop-

[15]Eine Übersicht der morphologischen Informationen, nach Wortklassen aufgeteilt, befindet sich in Anhang A.

pelpunkt getrennt. SEM1 beschreibt die semantische Kategorisierung nach Gattungsnamen (G), Stoffnamen (S) und Eigennamen (E). Die semantischen Merkmalsbündel $S_i$ sind wie folgt kodiert: An der ersten Stelle ist angegeben, ob es sich um ein Konkreta (k) oder Abstrakta (a) handelt. Es folgt ein „/“ und die Angabe von $\geq 0$ Bezeichnungen von kategoriellen Bedeutungen (die Trennung erfolgt durch Kommata).

**Beispiel 5.2 (Kodierung nominaler Informationen)**
*Frau|frau|N|sub|neF*:aeF*:deF*:geF*.G.k/belebt,hum*
*Apfel|apfel|N|sub|neM*:aeM*:deM*.G.k/belebt,plant*
*Ton|ton|N|sub|neM*:aeM*:deM*.G.k/unbelebt*
*Ton|ton|N|sub|neM*:aeM*:deM*.S.k/unbelebt,geg*
*Töne|ton|N|sub|nmM*:amM*:gmM*.G.k/unbelebt*

## 5.5 Lemmatisierungskomponente

Bei automatischen Lemmatisierungsverfahren lassen sich allgemein drei Teilaufgaben[16] unterscheiden:

1. Segmentierung: Der Text wird eingelesen und in relevante (lexikalische) Einheiten separiert.

2. Wortformenanalyse: Untersuche sämtliche Informationen, die durch die Wortformen repräsentiert werden.

3. Disambiguierung von Mehrdeutigkeiten.

Bei automatischen Lemmatisierungsverfahren wird häufig die Segmentierung vernachlässigt. Ich werde jedoch zeigen, daß bei einer intelligenten Segmentierung die Probleme der weiteren Arbeitsschritte reduziert werden können. Bei der Analyse der Wortformen reichen die Verfahren von der reinen regelbasierten Vorgehensweise bis hin zu vorwiegend lexikonbasierten Ansätzen.

[16] Eine strikte Trennung nach diesen Teilaufgaben findet sich in den Algorithmen nicht wieder.

Die Wortformenanalyse erfolgt in unserem Ansatz als zweistufiger Prozeß: zuerst wird versucht, eine Wortform auf einen Eintrag in der Vollformendatenbank abzubilden. Falls die Suche erfolglos war, wird in der zweiten Stufe eine spezielle Kompositakomponente angestoßen, in der komplexe Einheiten auf einfache Wortformen reduziert werden. Eine Sonderbehandlung sollte für Eigennamen, Abkürzungen und Zahlen vorgesehen werden[17].

Die dritte Aufgabe besteht in der Auflösung von Mehrdeutigkeiten, wobei wir wortbezogene und kontextbezogene bzw. satzbezogene Verfahren unterscheiden. Im ersten Fall kann lediglich über das Merkmal der Groß- und Kleinschreibung disambiguiert werden. Voraussetzung für kontextbezogene Verfahren ist eine idealerweise vollständige syntaktische und semantische Analyse des Satzes bzw. Kontextes.

Neben den Verfahren, die zur Disambiguierung aus einer Umgebungsanalyse gewonnene linguistische Informationen verwenden, können zur reinen Wortartendisambiguierung noch stochastische Verfahren eingesetzt werden. Dabei wird die Disambiguierung alleine aufgrund von Wahrscheinlichkeiten des Auftretens einer Wortform-Wortart-Kombination in ihrem lokalen Kontext durchgeführt. Jeder Wortform eines laufenden Textes werden zuerst die für sie möglichen Wortarten zugewiesen. Aus diesen Informationen wird ein Netz aller möglichen Wortform-Wortart-Kombinationen aufgebaut, wobei den Übergängen zwischen Wortarten Wahrscheinlichkeiten zugeordnet werden. Außerdem werden die Knoten mit sogenannten lexikalischen Wahrscheinlichkeiten gewichtet, welche die Wahrscheinlichkeit angeben, daß eine Wortform einer bestimmten Wortklasse angehört. Die eigentliche Disambiguierung erfolgt dann durch die Auswahl des Pfades, der die höchste kombinatorische Wahrscheinlichkeit aufweist. Formal handelt es sich bei diesem Verfahren um ein diskretes Hidden-Markov-Modell (HMM) erster Ordnung, d.h. es werden nur Übergangswahrscheinlichkeiten für direkt benachbarte Zustände betrachtet. Formal besteht ein HMM erster Ordnung aus:

1. einer Menge von Symbolen $W := \{w_1, \ldots, w_n\}$, hier die laufenden Wortformen des zu betrachtenden Textes.

[17] zur Zeit noch nicht im System implementiert

2. einer Menge von Zuständen $S := \{s_1, \ldots, s_m\}$, hier die Wortklassen.

3. einer Menge von $m^2$ Übergangswahrscheinlichkeiten zwischen Zuständen $P := \{p(s_1|s_1), \ldots, p(s_i|s_j)|1 \leq i, j \leq m\}$, hier die Übergangswahrscheinlichkeiten zwischen den Wortklassen.

4. einer Menge von Observationswahrscheinlichkeiten $L := \{p(w_1|s_1), \ldots, p(w_k|s_l)|1 \leq k \leq n, 1 \leq l \leq m\}$, hier die lexikalischen Wahrscheinlichkeiten.

Dann kann für eine gegebene Folge von i Symbolen unter allen Tagfolgen die wahrscheinlichste mit der folgenden Gleichung bestimmt werden:

$$\max_s \prod_{j=1}^{i} p(s_j|s_{j-1}) \times p(w_j|s_j)$$

Prinzipiell sind diese Modelle nicht auf Bigramme beschränkt, sondern es sind beliebig lange Ketten denkbar. Die Anzahl der Parameter steigt jedoch exponentiell an, so daß in der Praxis Modelle niedriger Ordnung (erster oder zweiter Ordnung) angewendet werden (siehe z.B. [Chu88, CKPS92, Kem93]). Das Verfahren der Hidden-Markov-Ketten für Wortartendisambiguierung gilt als generell sprachunabhängig, jedoch besteht die große Schwierigkeit in der Gewinnung der Parameter. Die Güte des Verfahrens hängt von der Zuverlässigkeit der Werte der Übergangswahrscheinlichkeiten und der lexikalischen Wahrscheinlichkeiten ab. Diese Wahrscheinlichkeiten beruhen in der Regel auf manuell getaggten Korpora, in denen die Häufigkeit bestimmter Abfolgen von Tags ermittelt werden.

### 5.5.1 Segmentierung

Ziel der Segmentierung ist es, den zu analysierenden Text einzulesen und auf relevante Einheiten abzubilden, wobei wir zwischen lexikalischen Einheiten im Sinne von Lemmata bzw. Einträgen in der Vollformendatenbank und Sonderzeichen wie Satzzeichen, Klammern etc. unterscheiden müssen. Auch das Erkennen von Zahlen kann durch die Stufe der Segmentierung geleistet werden.

Ein einfaches Verfahren liest den Text zeichenweise von links nach rechts und unterdrückt sämtliche Ziffern und Sonderzeichen. Das Ende einer relevanten Einheit ist bei Erkennen eines Leerzeichens oder speziell definierten Separatoren erreicht. Bei diesem Verfahren gehen selbstverständlich viele Informationen, wie Satzbegrenzer, Zahlen usw., verloren. Die Erweiterung des Verfahrens auf Zahlen und sonstige Zeichenkombinationen führt jedoch zu einigen inkorrekten Segmentierungen: Zahlen mit Dezimalkomma oder -punkt und Datumsangaben werden getrennt. Weiterhin ist eine Unterscheidung zwischen Abkürzungspunkten und Punkten als Satzendzeichen nicht möglich. Das Verfahren kann bei der Segmentierung ebenfalls keine wortinternen Klammerungen erkennen.

#### 5.5.1.1 Die Segmentierung in $\mathcal{PARLEX}$

Bei der Segmentierung werden die vorkommenden Strings $s_j$ auf Tokens $t_i$ abgebildet. Wir unterscheiden vier Mengen von Zeichen:

1. Buchstaben $\{A, \ldots, Z\}$ und $\{a, \ldots, z\}$
2. Ziffern: $\{1, \ldots 9, 0\}$
3. Sonderzeichen: $\{(, ), \{, \}, \ldots\}$
4. Satzzeichen: $\{, , ; , ., :, !, ?\}$

Bei der Behandlung von Klammern und Satzzeichen lassen sich allgemein verschiedene Fälle unterscheiden:

1. Es kommen keine Klammern oder Satzzeichen in einem String vor;
2. Ein String endet mit einem Satzzeichen;
3. Ein String beginnt und endet mit einer Klammer;
4. Ein String beginnt mit einer öffnenden Klammer;
5. Ein String endet mit einer schließenden Klammer.

Im zweiten Fall besteht das Token aus dem String ohne dem Satzzeichen. Falls das Satzzeichen ein Punkt ist, kann es sich um ein Satzendzeichen oder einen Abkürzungspunkt handeln. Ist das nächste Zeichen ein Kleinbuchstabe oder ein Satzzeichen außer dem Punkt, so handelt es sich um einen Abkürzungspunkt. Falls jedoch die nächste Wortform mit einem Großbuchstaben beginnt, kann die Frage nach Satzendzeichen nur durch Kontextwissen eindeutig beantwortet werden. Eine andere Möglichkeit, diese Frage mit hoher Wahrscheinlichkeit richtig zu beantworten, besteht in der Integration einer Abkürzungsdatenbank. Dann hängt die Güte des Verfahrens von der Größe bzw. Qualität der Datenbank ab.
Im dritten Falle werden Anfangs- und Endzeichen weggelassen. Bei den letzten beiden Fälle muß der String daraufhin untersucht werden, ob das Pendant der Klammer im Inneren des Strings vorkommt.

### 5.5.2 Wortformenanalyse

Die Wortformenanalyse geschieht in einem zweistufigen Prozeß, wobei in der ersten Stufe das zu untersuchende Token direkt in der Vollformendatenbasis nachgeschlagen wird. Bei Mißerfolg wird in der zweiten Stufe die Kompositakomponente aktiviert und versucht, das Token auf mehrere lexikalische Einheiten zu reduzieren.

#### 5.5.2.1 Kompositazerlegung

Die Komposition im Deutschen ist eine sehr produktive und gebräuchliche Art der Wortbildung. In Texten der Gegenwartssprache zählt man im Durchschnitt zwischen 5 und 15 Prozent Komposita.

Bei der Wortbildung unterscheidet Fleischer [FBS95] den wortstrukturellen und den nominationstheoretischen Ansatz:
Beim wortstrukturellen Ansatz strebt man nach einer syntaktisch orientierten Analyse der Wortstruktur, deren Ziel es ist, zwischen den Prinzipien der Satzsyntax und jenen der Wortsyntax einen engen Zusammenhang zu suchen.

Der nominationstheoretische Ansatz stellt dagegen die Benennungsfunktion der komplexen Wörter in den Vordergrund. Dabei ist eine Nominationseinheit ein sprachlicher Ausdruck, der einen Wirklichkeitsausschnitt als Klasse von Gegenständen repräsentiert.
Die Zerlegungsstrategie bei Komposita beruht auf der folgenden Definition:

**Definition 5.2**
*Jedes n-gliedrige Kompositum $K_n$ hat die Form:*

$$K_n := L_1 \triangleright F_1 \bullet L_2 \triangleright F_2 \bullet \cdots \bullet L_n$$

*wobei gilt:*

1. *$F_i$ Fugenelement des Lexems $L_i$ für $1 \leq i \leq n$,*
2. *$\triangleright$ Verknüpfung des Lexems mit seinem Fugenelement,*
3. *$\bullet$ Kopplung der Lexeme $L_i$ und $L_{i+1}$ durch:*
   (a) *einfache Konkatenation $L_i \circ L_{i+1}$*
   (b) *Konkatenation mit Bindestrich $L_i - L_{i+1}$*
   (c) *Konkatenation mit Schrägstrich $L_i / L_{i+1}$*

*Das Kompositum $K_n$ hat dieselbe grammatische Funktionsklasse wie das Lexem $L_n$.*

Die meisten Komposita weisen eine binäre Struktur auf. Die Konstituenten eines Kompositums stehen in Relation zueinander. Bis auf eine kleine Gruppe von Kopulativkomposita ist die Reihenfolge der Lexeme relevant, wobei das Zweitglied[18] die Wortart, Genus und Flexionstyp bestimmt. Man spricht vom Kern oder Kopf des Kompositums. Der Kopf eines Determinativkompositums wird durch die erste Konstituente näher bestimmt (z.B. *Hauptstadt*). Die wichtigsten Zusammensetzungen im Deutschen sind die Nominalkomposita, abhängig von der Wortart des Kopfes werden diese in Nomen-, Adjektiv- oder Partizipialkomposita

[18] Bei $n > 1$ das n-te Lexem

eingeteilt, die folgende Tabelle zeigt die Arten der Nominalkomposita mit ihren Häufigkeiten:

Tabelle 5.6: Nominalkomposita mit Häufigkeitsangaben

| Kompositumtyp | Häufigkeit | Beispiel |
|---|---|---|
| Nomen | 83,6 % | Gruppenfoto |
| Adjektiv | 8,6 % | gruppenkonform |
| Partizip II | 5,0 % | gruppenbestimmt |
| Partizip I | 2,8 % | gruppenbildend |

Die Flexion innerhalb eines Kompositums ist gelöscht. Um eine effiziente Kompositakomponente erstellen zu können, müssen wir die vorkommenden Fugenformen untersuchen. Die Form richtet sich vor allem nach Wortart und Flexionstyp, lautlicher und morphologischer Gestalt des Wortausgangs, nach der Bedeutung und vereinzelt nach der Länge des Lexems. Fugenformen kommen im Deutschen nur bei Nomina, Adjektiven und Verbalstämmen vor. Neben den Fugenformen werden bei der deutschen Wortbildung außerdem noch Tilgungs- oder Umlautungsoperationen durchgeführt. Bei den Verben kommt neben der Null-Fuge nur noch „+e“ (z.B. Tragetasche) als Fugenelement[19] vor.
Die Nomina weisen im Deutschen den größten Formenreichtum an Fugen auf. Die folgende Tabelle zeigt, welche Fugenelemente und zugehörigen Operationen in dem implementierten Komposita-Algorithmus Berücksichtigung finden:

[19]Bei der Beschreibung der Fugenelemente werden die folgende Bezeichnungen verwendet:

- $-s_1 \dots s_k$ Tilgung der Strings $s_1 \dots s_k$,
- Umlautung U,
- füge Null(+0) oder die Strings $s_1 \dots s_k$ als Fugenelemente hinzu.

Tabelle 5.7: Fugenformen nach Nomina

| Fuge | zusätzliche Operation | Beispiel |
|---|---|---|
| +0 | | Dampfmaschine |
| | U | Töchterschule |
| | -e | Wettbüro |
| | -en | Südhang |
| | -n | Osterwetter |
| +al | | Medizinalrat |
| | -um | Gymnasialbildung |
| +e | | Hundefutter |
| | U | Gästebuch |
| +en | | Sternenhimmel |
| | -a | Firmenschild |
| | -um | Museenverwaltung |
| | -us | aphorismenreich |
| | -os | Mythentradition |
| | -s | Heroenkult |
| +een | -us | Kakteensammlung |
| +ens | | herzensgut |
| +er | | Geisterfahrer |
| | U | Blätterwald |
| +es | | Geistesblitz |
| +ien | | Prinzipienreiter |
| +n | | Lungenmaschine |
| +nen | | Studentinnentreffen |
| +ns | | Namenszug |
| +s | | Antrittsrede |
| | -e | Gebirgsjäger |
| | -en | Weihnachtsmann |
| +ß | -sse | Adreßbuch |
| +ten | | Bautenschutz |

Bei Adjektiven wird in zusammengesetzten Formen in der Regel die Positivform verwendet, sehr beschränkt kommen Superlativformen vor (Schwerstarbeit oder weitestgehend).

Tabelle 5.8: Fugenformen nach Adjektiven

| Fuge | zusätzliche Operation | Beispiel |
|---|---|---|
| +0 | | Kleinwagen |
| | -isch | Disziplinarverfahren |
| +al | -ell | Experimentalphysik |
| +ar | -är | Popularphilosophie |
| +o | | Brutalowestern |
| | -isch | Thermodynamik |

Bei der Analyse der Wortformen können Ambiguitäten bezüglich der Aufspaltung auftreten. Betrachten wir z.B. die Wortform *Staubecken*, so stellen wir fest, daß wir die Wortform auf zwei verschiedene Arten aufspalten können, nämlich *Staubecken* und *Staub-ecken*. Bei mehrgliedrigen Komposita wie *Schranktürschlüssel* stellt sich ebenfalls die Frage, ob es sich um ein linksverzweigtes Kompositum *[Schrank-tür]-schlüssel* oder rechtsverzweigtes *Schrank-[tür-schlüssel]* handelt.
Da im Deutschen die meisten Wortzusammensetzungen eine binäre Struktur aufweisen, beschränkt sich der implementierte Algorithmus auf die Behandlung von zweistelligen Komposita. Die Lemmatisierung kann mit oder ohne Kompositabetrachtung durchgeführt werden.

## 5.6 Kopplung von Lexikalischer Wissensbasis und Parser

Die Kommunikation des Parsers mit der lexikalischen Wissensbasis erfolgt in der jetzigen Version über Files. Der zu analysierende Text wird im ersten Schritt lemmatisiert und das Ergebnis in einem Textfile an den Parser übergeben. Bei der Lemmatisierung wird der Text einmal von rechts nach links durchlaufen. In dem File werden für jede Position des Textes sämtliche Alternativen der analysierten Wortform gespeichert, weiterhin werden Satzzeichen und Klammern gesondert markiert. Abbildung 5.16 zeigt das Ergebnis-File für den Beispielsatz „Die Studenten berufen eine Vollversammlung wegen der Studiengebühren ein.“

```
1:Die|der/die/das|D|det-def|neFDw:aeFDw:nmMDw:nmFDw:
nmNDw:amMDw:amFDw:amNDw
2:Studenten|Student|N|sub|nmM*:gmM*:dmM*:amM*.G.
3:berufen|einberufen|V|HV|1mGi:3mGi:1mGc:3mGc:1mOb.ein.ST.*.*.*.
4:eine|ein/eine/eines|D|qnt-ind|neFot:aeFot:nmMot:nmFot:
nmNot:amMot:amFot:amNot
4:eine|einen|V|HV|1eGi:1eGc:3eGc:1eOb.*.SCHW.*.*.*.
5:Vollversammlung|VollVersammlung|<kompositum>A-N
</Kompositum>|N|<bestimmung>voll|voll|A|*|vp.</bestimmung>
<Basis>Versammlung|Versammlung|N|sub|neF*:geF*:deF*:aeF*.G.
</basis>
6:ein|ein/eine/eines|D|qnt-ind|neMot
```

Abbildung 5.16: Output-File der Lemmatisierungskomponente

Da die lexikalische Wissensbasis neben den morphosyntaktischen Informationen auch Wissen über Valenzstrukturen von Verben und Vertretern anderer Wortklassen enthält, möchte ich weiterhin eine Parsing-Strategie für den lexikalischen Parser vorschlagen, die wir als „verbzentriertes Parsen“ bezeichnen. Damit diese Strategie erfolgreich eingesetzt werden kann, müssen die folgenden beiden Bedingungen erfüllt sein:

1. Das Lemmatisierungsprogramm muß eine genaue Zuordnung von Satz bzw.

Satzendzeichen gewährleisten, damit der zu analysierende Text in Sätze segmentierbar ist.

2. Hauptverben müssen in der lexikalischen Wissensbasis mit ihren Valenzstrukturen beschrieben sein.

Sind diese Bedingungen erfüllt, erscheint folgende Parsing-Strategie sinnvoll: Der Text wird in Satzeinheiten unterteilt und anschließend wird das Hauptverb eines Satzes extrahiert. Durch die zugeordneten Valenzmuster werden nun in Form der zu füllenden Slots Erwartungen durch logische Formeln definiert, die erfüllt sein müssen, falls das Valenzmuster auf den zu analysierenden Satz anwendbar ist. Bei den Erwartungen handelt es sich nicht nur um Beschreibungen auf syntaktischer Ebene, sondern es werden zusätzlich semantische Angaben zur Analyse herangezogen.
Betrachten wir etwa die Valenzbeschreibung des Verbs „spielen“ (siehe Abbildungen 5.1.2.2.1 und 5.1.2.2.1), so werden durch die Beschreibung verschiedene Erwartungen definiert, die mittels des Parsers geprüft werden. Auf der ersten Position wird in jedem Fall eine Nominalergänzung im Nominativ erwartet, die evtl. semantisch auf etwas Konkretes, Menschliches eingeschränkt ist. Das Verb hat noch zwei weitere Slots, die Erwartungen bez. einer Nominalphrase im Akkusativ und einer Präpositionalphrase mit der Präposition „um“ oder „mit“ beschreiben. Legen wir einmal dieses Valenzmuster als das einzig mögliche zugrunde, so würde der Beispielsatz „Die Mutter spielt mit dem Kind“ erkannt, jedoch „Die Mutter spielt mit dem Schach“ aufgrund der semantischen Restriktion bei der Präpositionalphrase mit „um“ abgewiesen. Die Anzahl der möglichen Parse-Läufe ist bei dieser Strategie abhängig von der Anzahl der Valenzbeschreibungen des Hauptverbs.
Die Parsing-Strategie kann nur verfolgt werden, wenn für alle Hauptverben Valenzbeschreibungen auf (zumindest) syntaktischer Ebene existieren. Da diese Forderung in der Praxis nicht erfüllt werden kann, muß für den Fall, daß keine Valenzzuordnung erfolgt ist, eine Alternativstrategie implementiert werden. Da nach Heringer von den möglichen syntaktischen Valenzmuster nur etwa 20 realisiert sind, schlagen wir in diesen Fällen folgende Strategie vor: Im Vorfeld des Haupt-

verbs findet sich immer eine Nominalphrase im Nominativ, so daß zuerst nach dieser Erwartung gesucht wird. Anschließend können die im Deutschen bekannten syntaktischen Muster, geordnet nach ihrer Auftrittshäufigkeit, überprüft werden.

# Teil II

# Der Semantische Inspektor - Ein Werkzeug für die quantitative Linguistik

# Kapitel 6

# Einleitung

In Zusammenarbeit mit Prof. Heringer wurde ein Programm entwickelt, das mittels statistischer Verfahren neue Aspekte der semantischen Beschreibung hervorbringen soll. Der Semantische Inspektor kann somit als ein Werkzeug der *quantitativen Linguistik* angesehen werden, die sich nach Köhler und Rieger [KR93] zwar nicht grundsätzlich in ihren Zielen von anderen Richtungen der Linguistik unterscheiden, durch den Einsatz von mathematischen Modellen wohl aber in ihren Methoden.

Ausgangspunkt sind Forschungen von Prof. Heringer, die zum Ziel haben, das (semantische) Umfeld eines bestimmten Wortes (im Folgenden auch als Stichwort bezeichnet), bezogen auf ein bestimmtes Textkorpus, mit empirischen Methoden zu bestimmen und plausibel zu visualisieren. Ein Resultat sind sogenannte Sterndarstellungen, welche die Affinitätswerte[1] der am häufigsten vorkommenden Wörter im Umfeld des Stichwortes darstellen. Das entwickelte Programm visualisiert diese Werte in Form von Strahlen eines Sterns. Während die Sterndarstellung die affinsten Wörter in Beziehung zum Stichwort darstellt, verfügt das Programm weiterhin über Komponenten, welche die affinsten Wörter zueinander in Beziehung setzen. Dabei kommen mit dem statistischen Verfahren der *Multidimensionalen Skalierung* (MDS-Verfahren) und dem konnektionistischen Verfahren der

[1] Die Definition dieses Ähnlichkeitsmaßes erfolgt weiter unten.

*selbstorganisierenden Karten* SOM[2] zwei Strukturen-entdeckende Verfahren[3] zum Einsatz, die Zusammenhänge zwischen Objekten aufdecken können.
Gerade bei lexikalisch ambigen Stichwörtern könnten sich dabei Anordnungen[4] in dem Sinne ergeben, daß sich Mengen von Objekten bilden, die eine enge Beziehung zueinander haben und eine schwache zu anderen Objekten, und die im Kontext einer bestimmten Bedeutung auftreten.

Das folgende Kapitel erläutert Aspekte der Mehrdeutigkeit von natürlichen Sprachen. Anschließend stelle ich das System des Semantischen Inspektors vor.

[2]self-organizing maps

[3]Nach Backhaus [Bac96] sind Strukturen-entdeckende Verfahren solche multivariaten Verfahren, deren primäres Ziel in der Entdeckung von Zusammenhängen zwischen Variablen oder Objekten liegt.

[4]Wir sprechen hier nicht von Clustern, da bei multivariaten Analysemethoden zwischen Cluster- und Projektionsverfahren unterschieden wird (s.a. Abschnitt 8.3)

# Kapitel 7

# Ambiguität natürlicher Sprachen

Ambiguitäten in dem weiteren Sinn von natürlichsprachlichen Mehrdeutigkeiten gibt es in verschiedenen Formen. Unterscheiden wir bei natürlicher Sprache die Ebenen Syntax, Semantik und Pragmatik, so können wir mindestens drei Typen der Ambiguität festmachen:

- Eine *syntaktische Ambiguität* ist durch alternative Kategorisierungen der Oberfläche charakterisiert. Berücksichtigt man nicht die Unterscheidung zwischen Groß- und Kleinschreibung, so ist die Wortform *ARBEITEN* syntaktisch ambig, weil sie die Kategorien eines Nomen im Plural und des Infinitivs eines Verbs hat. In dem Beispiel *Susanne mag lange ARBEITEN* bleibt die syntaktische Ambiguität der Wortform auch innerhalb des komplexen Ausdrucks erhalten.

- Eine *semantische Ambiguität* liegt vor, wenn die Oberfläche eine Kategorie, aber mehrere wörtliche Bedeutungen hat. Die Wortform *BANK* ist nicht syntaktisch ambig, weil sie nur als feminines Nomen im Singular kategorisiert ist. Sie ist aber semantisch ambig, weil ihr die unterschiedlichen Bedeutungen von „Geldinstitut“ und „Sitzgelegenheit“ zugeordnet sind.

- Eine *pragmatische Ambiguität* besteht in alternativen Verwendungsmöglichkeiten einer wörtlichen Bedeutung in einem gegebenen Interpretationskontext. So kann z.B. der Satz *„Nimm das Auto zum Einkaufen!“* pragmatisch

ambig sein, wenn der Interpretationskontext zwei Autos aufweist, die gleichermaßen als Referenzkandidaten in Frage kommen.

Die syntaktischen und semantischen Ambiguitäten können durch den Kontext disambiguiert werden. Pragmatische Ambiguitäten können per Definition nicht über den Kontext aufgelöst werden, da sie durch eine unzureichend eingeschränkte Referenzbeziehung zum Kontext entstehen.

## 7.1 Polysemie und Homonymie

Polysemie und Homonymie gehören zur Klasse der lexikalischen Ambiguitäten[1]. Bei der Definition und Abgrenzung der beiden Begriffe gibt es unter den Linguisten verschiedene Ansätze. In [Meh93] werden zwei geläufige Kriterien genannt:

- Diachrone Rückführbarkeit der verschiedenen Bedeutungen auf ein Wort. Falls ein solcher gemeinsamer Ursprung existiert, spricht man von Polysemem, sonst von Homonymen.
- Synchrone Verwandschaft der Bedeutungen. Falls eine Verwandschaft zwischen den unterschiedlichen Bedeutungen existiert, spricht man von Polysemie, ansonsten von Homonymie.

Andere Autoren geben die Unterscheidung ganz auf oder stellen sie zurück.

> [Her81]:
> *„Ich befasse mich nur mit dem, was ich Ambiguität nenne und als Vorliegen mehrerer Bedeutungen eines Ausdrucks eingeführt habe. Ambiguität ist sowohl der Polysemie wie der Homonymie vorgeordnet, falls man diesen Unterschied machen will.“*

In seinen Arbeiten zum *Generativen Lexikon* [Pus91, Pus95, PB96] unterscheidet Pustejovsky in Anlehnung an Weinreich zwischen kontrastiver Ambiguität im

[1] Wir behandeln die Begriffe semantische Ambiguität und lexikalische Ambiguität als Synonyme.

Sinne von mindestens zwei Bedeutungsvarianten ohne Überschneidung (Homonymie) und komplementärer Ambiguität im Sinne von ergänzenden Bedeutungen (Polysemie). Bei komplementärer Polysemie betrachtet Pustejovksy noch die Spezialfälle der logischen Polysemie, bei der ein Wechsel der lexikalischen Kategorie nicht stattfindet. Er ordnet diese Klasse von logischen Polysemien anschließend nach der Art der Bedeutungsverschiebungen. Im Folgenden möchte ich die unterschiedlichen Arten von Ambiguitäten noch einmal durch Beispielsätze illustrieren.

**Beispiel 7.1 (Unterschiedliche Arten von Ambiguitäten)**

1. (a) *Der Mann sitzt auf der* Bank.
   (b) *Die* Bank *sperrte den Kredit.*
   (c) *Die Frau betrat die* Bank.
2. (a) *Mary aß einen* Apfel.
   (b) *Mary pflückte die Äpfel vom Baum.*
3. (a) *John traveled to* New York.
   (b) New York *kicked the mayor out of office.*

Betrachtet man die Beispielsätze aus 1, so kann man feststellen, daß die Bedeutung von *Bank* aus 1a wenig mit der Bedeutung des Worts *Bank* aus 1b und 1c gemeinsam hat. In diesem Fall würde man von Homonymen sprechen. Bei den Sätzen 1b und 1c sind die Bedeutungsunterschiede jedoch nicht so klar. Handelt es sich im zweiten Satz um die *Bank* als „Institution“, so ist im dritten Satz die *Bank* als „Gebäude“ gemeint. In diesem Fall würde man eher von verwandten Bedeutungen und somit von Polysemen sprechen.
Die Sätze[2] aus 2 und 3 beschreiben Beispiele für logische Polysemien. Im ersten Fall spricht Pustejovsky von „Plant/Food“-Alternation, den Zweiten bezeichnet er als „Place/People“-Alternation.

[2] Aus [Pus95], Seite 31.

# Kapitel 8

# Der Semantische Inspektor

In diesem Kapitel beschreibe ich die Funktionalität des Semantischen Inspektors. Ich beginne mit der Aufbereitung von Belegsammlungen und der Berechnung der Affinität. Anschließend beschreibe ich das Verfahren der Sterndarstellung und die Struktur-entdeckenden Verfahren des Semantischen Inspektors.

## 8.1 Belegsammlungen und die Berechnung der Affinität

Ausgangspunkt für alle drei Verfahren stellen sogenannte Belegsammlungen für ein bestimmtes Stichwort dar. Ein Beleg besteht aus einem Vorfeld von r Objekten (Wörtern), dem Stichwort und einem Nachfeld von ebenfalls r Objekten. Ein Stichwort steht dabei in der Regel nicht für ein Wort, sondern für eine Menge von interessierenden Wörtern, z.B. könnte für das Stichwort *liebe* eine Menge wie folgt aussehen: $\{liebe, liebe, lieber, liebevoll, liebeshungrig, \dots\}$.
Die Belegsammlungen werden aus großen Textkorpora extrahiert, wobei die Belege ohne Rücksicht auf syntaktische Strukturen und Satzgrenzen etc. herausgeschnitten werden[1]. Es werden somit nicht Sätze betrachtet, die das Stichwort

[1] Bei der Erzeugung einer Belegsammlung werden in der jetzigen Programmversion externe Assembler-Programme eingesetzt, die von Markus Ohlenroth von der Universität Augsburg entwickelt wurden.

enthalten, sondern es wird davon ausgegangen, daß zwischen dem Stichwort und seiner Umgebung Abhängigkeiten mit einer gewissen Reichweite existieren. Anschließend werden sämtliche Wortformen der Belege auf ihr Lemma abgebildet. Diese Lemmatisierung erfolgt in der jetzigen Programmversion mittels einer Abgleichtabelle mit zur Zeit ca. 400.000 Einträgen[2].
Nachdem die Belege lemmatisiert sind, wird für jedes vorkommende Lemma ein Affinitätswert berechnet, der durch die folgende Formel definiert ist:

**Definition 8.1 (Affinität)**
*Es sei die mittlere Distanz* $mittl\_Distanz := \frac{Distanzsumme\_zum\_Stichwort}{Anzahl\_der\_Vergleiche}$ *und die relative Häufigkeit* $rel\_Häufigkeit := \frac{abs\_Häufigkeit}{Gesamtanzahl\_Wörter}$. *Dann definieren wir die Affinität zu einem Stichwort wie folgt:*

$$Affinität := \frac{mittl\_Distanz^{\lambda}}{rel\_Häufigkeit^{(1-\lambda)}}, \lambda \in [0,1]$$

Ein Lemma ist also umso affiner zum Stichwort, je häufiger es vorkommt und je geringer der Abstand zum Stichwort im Beleg ist. Die Formel ist mit $\lambda$ parametrisiert, so daß eine Gewichtung der beiden Einflußgrößen „Nähe zum Stichwort“ und „Anzahl der Vorkommen“ möglich ist.
Beim ersten Prototyp des Semantischen Inspektors wurde bei der Lemmatisierung der Belege die Stelle des Stichwortes unverändert gelassen. Eine Untersuchung von verschiedenen Belegsammlungen zeigte jedoch, daß durch die Nichtbeachtung von Wortzusammensetzungen, die Berechnung der Affinitätswerte stark „verfälscht“ wurde. Betrachten wir als Beispiel das Stichwort *Mutter*, so kann man vermuten, daß *Kind* als Objekt häufig im Kontext des Stichworts anzutreffen ist. Bei der Lemmatisierung ohne Behandlung von Wortzusammensetzungen werden jedoch Wortformen wie *Mutter-Kind-Kuren* oder *Mutter-Kind-Turnen* nicht in die Betrachtungen mit einbezogen. Aufgrund dieser Beobachtung haben wir zwei weitere Versionen entwickelt, die bei der Position des Stichwortes unterschiedliche Aufspaltungen durchführen: Eine Version führt bei der Lemmatisierung zusätzlich

[2]In einer späteren Programmversion soll das PARLEX-System integriert werden, so daß durch den Einsatz der lexikalischen Wissensbasis die Lemmatisierung verbessert werden kann und darüber hinaus durch den Einsatz der Parserkomponente auch syntaktisches Wissen zur Auflösung von Mehrdeutigkeiten dienen kann.

eine Kompositabehandlung durch, so daß die abgespaltenen Lemmata unmittelbar vor bzw. nach der Position des Stichwortes eingefügt werden. Diese Variante läßt jedoch bei der Betrachtung der Ergebnisse leider keine Rückschlüsse darauf zu, wie oft ein Lemma alleine oder durch eine zusammengesetzte Stichwortform in die Berechnungen eingeflossen ist.
Die zweite Variante betrachtet das angegebene Stichwort als eine Art Wurzel. Der String der Stichwortposition wird dann in einen vorderen Teil, die Wurzel und einen hinteren Teil aufgespalten. Die abgespaltenen Teile erhalten eine zusätzliche Markierung, so daß zwischen im Beleg vorkommenden Lemmata und vom Stichwort abgetrennten Strings unterschieden werden kann[3].

## 8.2 Die Sterndarstellung

In der Komponente der Sterndarstellung wird der Affinitätswert eines Objektes durch die Länge des Strahls wiedergespiegelt. Je kürzer der Strahl (je kleiner der Affinitätswert) eines Objektes ist, umso affiner ist das Objekt zum Stichwort. Die Objekte werden dabei einfach in der alphabetischen Reihenfolge um das Stichwort angeordnet.
Bei der Darstellung haben die Anwender die Möglichkeit, eine Auswahl von Objekten zu treffen, wobei zwischen den beiden folgenden Optionen gewählt werden kann:

1. Die Auswahl erfolgt nur nach dem Affinitätswert, d.h. in der nach Affinitätswerten sortierten Liste werden verschiedene Objekte ausgewählt.

2. Die Liste der affinsten Objekte wird zuerst nach bestimmten Wortklassen gefiltert, und anschließend wird eine Auswahl von Objekten getroffen.

Abbildung 8.1 zeigt das Sterndisplay des Semantischen Inspektors mit einem Stern für das Stichwort „schön“, wobei der Darstellung 583 Belege mit Radius

[3]In der weiteren Darstellung werden sowohl die Lemmata, als auch die abgetrennten Strings als Objekte bezeichnet.

5 zugrundeliegen. Die beiden Einflußgrößen „Nähe zum Stichwort“ und „Auftrittshäufigkeit“ wurden gleich gewertet. Dabei wurde die Liste der affinsten Objekte nach den Wortklassen Nomen, Adjektiv und Verb gefiltert und anschließend die ersten 50 Objekte ausgewählt. Betrachten wir die Klasse der Nomen, so finden wir unter den affinsten Objekten Wörter wie „Natur“, „Mensch“, „Bild“ oder „Frau“. Bei den Adjektiven finden sich Wörter wie „zeitlos“, „bunt“ oder „alt“.

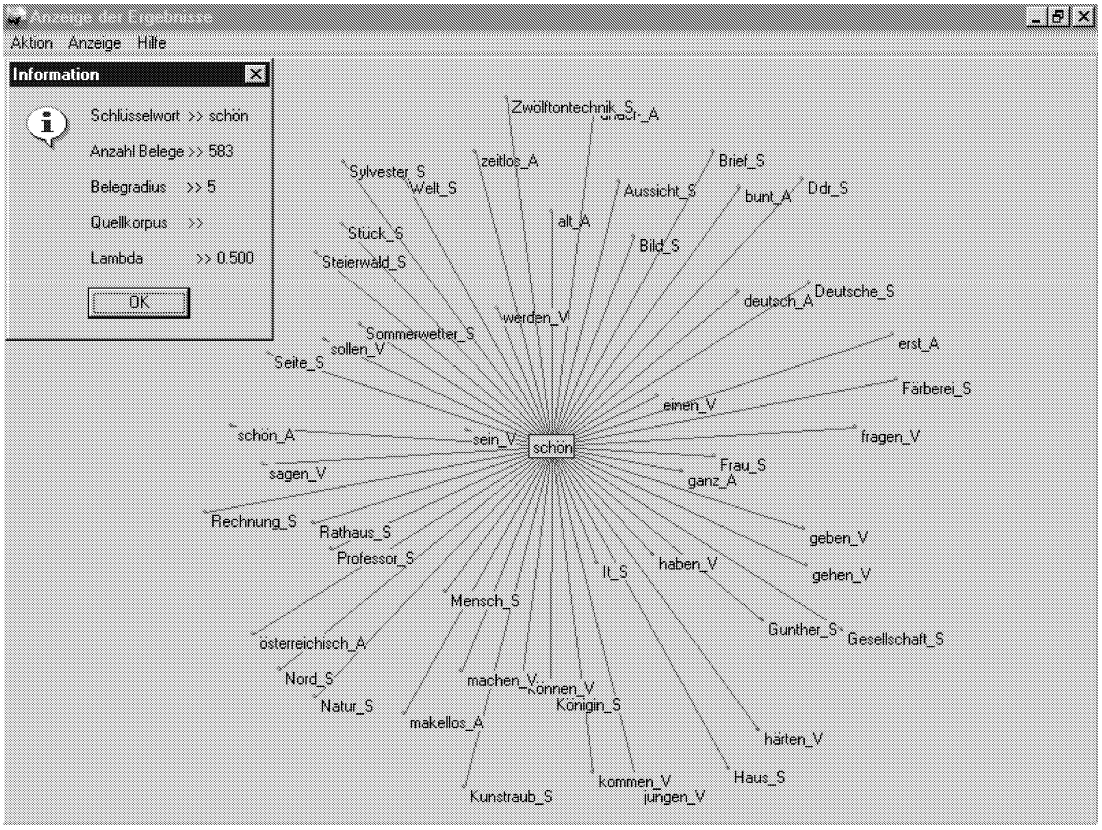


Abbildung 8.1: Das Sterndisplay des Semantischen Inspektors

In Abbildung 8.2 wurde mit dem Beispiel *Bank* im Gegensatz zu dem Stichwort *schön* ein extrem polysemes Stichwort ausgewählt. Die Darstellung beruht auf der Auswertung von 1224 Belegen mit Radius 8 und Affinitätsparameter $\lambda = 0.5$, berücksichtigt wurden die 80 affinsten Objekte der Klasse Substantiv, Verb und Adjektiv. Es zeigt sich, daß neben den wegen ihrer Häufigkeit zu erwartenden Hilfsverben wie „sein“ eine große Anzahl von Wörtern hohe Affinitätswerte erhal-

ten, die in engerer semantischer Verbindung zum Stichwort stehen. Betrachten wir etwa die Klasse der Substantive, so finden wir Wörter wie Hypothek, Aufsichtsbehörde, Börse und Prozent, die in dem Kontext „Bank als Institution“ stärker vertreten sind. Eine andere semantische Zuschreibung „Bank als Sitzmöbel“ spiegelt sich in Wörtern wie sitzen, Stuhl, ruhen, etc. wieder. Weiterhin finden sich auch Wörter in der Sterndarstellung, die andere semantische Zuschreibungen vermuten lassen, wie Dreh(bank) oder Korallen(bank). Die Tatsache, daß sich ein Stichwort nicht unbedingt auf zwei (oder einige wenige) semantische Zuschreibungen reduzieren läßt, wird uns bei der Behandlung der Strukturen-entdeckenden Methoden noch eingehender beschäftigten.

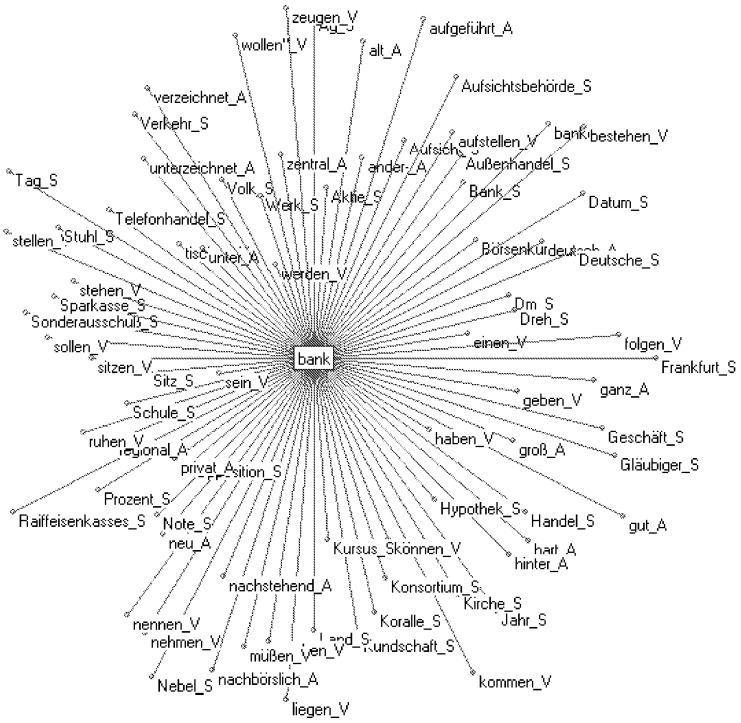


Abbildung 8.2: Beispiel einer Sterndarstellung für das Stichwort Bank

## 8.3 Strukturen-entdeckende Verfahren

Generell lassen sich die Strukturen-entdeckenden Verfahren nach Backhaus [Bac96] in Clusteranalysen, Faktorenanalyse und Multidimensionale Skalierung unterteilen.
Bei der Faktorenanalyse versucht man eine Vielzahl von beobachteten Variablen auf wenige Einflußgrößen zu reduzieren. Die Reduktion erfolgt dabei mit dem Ziel, die Variablen derart zu bündeln, daß voneinander unabhängige Beschreibungs- und Erklärungsvariablen neu entdeckt werden. Können die Variablen auf eine Beurteilungsdimension von maximal drei verdichtet werden, so lassen sich die Objekte im Raum dieser Dimension graphisch darstellen.
Bei der Clusteranalyse wird eine Bündelung von Objekten angestrebt. Das Ziel ist dabei, die Objekte so zu Clustern zusammenzufassen, daß die Objekte in einer Gruppe möglichst ähnlich und die Gruppen untereinander möglichst unähnlich sind.
Im Gegensatz zur Faktorenanalyse werden bei der MDS nicht die subjektiven Beurteilungen von Eigenschaften der untersuchten Objekte erhoben, sondern es werden nur wahrgenommene globale Ähnlichkeiten zwischen den Objekten erfragt. Mittels der MDS werden die diesen Ähnlichkeiten zugrundeliegenden Wahrnemungsdimensionen abgeleitet.
Der Semantische Inspektor unterstützt zwei unterschiedliche Verfahren bei der multivariaten Datenanalyse. Mit dem Verfahren der *Multi-Dimensionalen-Skalierung* (MDS) wird ein bekanntes statistisches Verfahren eingesetzt, demgegenüber steht mit den *selbstorganisierenden Karten* (siehe [Koh97]) ein konnektionistisches Verfahren. Während die MDS zu den Projektions- bzw. Positionierungsverfahren zählt, nimmt das konnektionistische Verfahren der selbstorganisierenden Karten eine Sonderstellung ein, da es sowohl zur Reduzierung der Anzahl der Objekte durch Clusterbildung, als auch zur Projektion der Daten in einen niedrigdimensionierten Raum benutzt werden kann (siehe auch [Kas97]). Im Rahmen dieser Arbeit wird nur das MDS-Verfahren behandelt[4].

[4]Das Verfahren der selbstorganisierenden Karten wurde im Rahmen einer Studienarbeit [Mü98] in den Semantischen Inspektor integriert, bzw. läuft als eigenständige Anwendung unter dem Namen PolySom.

### 8.3.1 Das Verfahren der Multi-Dimensionalen Skalierung

Die Multidimensionale Skalierung (MDS) dient der Darstellung von (hochdimensionalen) Objekten in einem niedrigdimensionalen Repräsentationsraum. Dabei geht man von Ähnlichkeiten der Objekte aus, die entweder aus den p an ihnen beobachtbaren Merkmalswerten berechnet oder direkt beobachtet werden und sucht eine Konfiguration der Objekte im Repräsentationsraum derart, daß die Ähnlichkeiten durch ein vorgegebenes Distanzmaß möglichst gut repräsentiert werden. Bei der MDS unterscheidet man zwischen metrischen und nicht-metrischen Verfahren.

Die metrischen Verfahren, die auf Torgenson [Tor58] zurückgehen, benötigen konkrete Werte für die Ähnlichkeiten der n zu betrachtenden Objekte in Form einer Distanzmatrix. Wie beim Grundmodell der klassischen metrischen Skalierung kann man Ansätze der metrischen MDS in ein Distanz- und ein Raummodell einteilen. Im Distanzmodell werden die für die $n = 1 \ldots N$ Objekte erhobenen Ähnlichkeitsdaten in Distanzen oder Unähnlichkeiten $d_{n,m}$ $(mit\ n, m = 1 \ldots N)$ transformiert. Dabei muß das Distanzmaß für $n, m = 1 \ldots N$ folgende Bedingungen notwendigerweise erfüllen:

1. $d_{n,n} = 0$ und $d_{n,m} \geq 0$
2. $d_{n,m} = d_{m,n}$

Falls das Distanzmaß auch noch die Dreiecksungleichung erfüllt, spricht man von einem metrischen Distanzmaß.

Durch das Raummodell sollen die Objekte durch N Punkte $x_1, \ldots, x_N$ in einem k-dimensionalen Raum $\Re^k$ so repräsentiert werden, daß die metrischen Distanzen $d(n, m) = d(x_n, x_m)$ der Objekte die durch das Distanzmodell vorgegebenen Distanzen $d_{n,m}$ möglichst gut approximiert. Als metrische Distanzen werden dabei in der Regel Minkowski-Metriken eingesetzt.

**Definition 8.2 (Minkowski q-Metrik ($L_q$-Metrik))**
*Die $L_q$-Metriken sind definiert durch*

$$d_q(n, m) = (\sum_{s=1}^{t} |x_{ns} - x_{ms}|^q)^{1/q}\ mit\ q > 0$$

*Wobei $x_{ns}$ für den Koordinatenwert des Objektes n auf der Achse s steht und t die Dimension des Raumes angibt. $L_2$ heißt auch Euklidische Distanz.*

Die verschiedenen Ansätze der metrischen MDS-Verfahren lassen sich dadurch unterscheiden, wie im Raummodell die Approximation zwischen den Distanzen $d_{n,m}$ und der Metrik $d_q(n, m)$ definiert und das so entstehende Approximationsproblem gelöst wird. Beispiele für metrische MDS-Verfahren sind *Nonlinear Mapping* und die *Haupt-Koordinaten-Methode* (siehe auch [HE95])
Die Verfahren der nicht-metrischen MDS benötigen nur Informationen über den Grad der Ähnlichkeit von Objekten, d.h. etwa welches Objektpaar ist sich am ähnlichsten, am zweitähnlichsten usw. Diese Eigenschaft macht die nicht-metrischen Verfahren besonders interessant für sozialwissenschaftliche Forschungen, wo häufig „subjektive Wahrnehmungen von Objekten durch Personen“ [Bac96] anhand paarweiser Ähnlichkeitseinschätzungen untersucht werden. Eine Auskunftsperson muß in diesem Fall lediglich die subjektiv empfundene Ähnlichkeit oder Unähnlichkeit einschätzen. Aus diesen Ähnlichkeitsurteilen läßt sich mit Methoden der nicht-metrischen MDS dann die Konfiguration der Objekte im Wahrnehmungsraum der Person ableiten. Ein Beispiel für nicht-metrische Verfahren, ist das Verfahren von Kruskal, das auch im Semantischen Inspektor angewendet wird[5].
Als Vorteile des Verfahrens sind laut [Bac96] zu nennen:

- Die relevanten Eigenschaften der Objekte können unbekannt sein.
- Es erfolgt keine Beeinflussung der Ergebnisse durch die Auswahl der Eigenschaften und deren Verbalisierung.

Der gravierende Nachteil liegt darin, daß die Ergebnisse der MDS schwierig zu interpretieren sind, da der Bezug zwischen den gefundenen Dimensionen des Wahrnehmungsraums und den empirisch erhobenen Eigenschaften der Objekte nicht besteht.

[5]Das Verfahren wird in 8.3.2.2 erläutert

### 8.3.2 Das MDS-Verfahren im Semantischen Inspektor

Dieser Abschnitt behandelt das verwendete MDS-Verfahren des semantischen Inspektors. Dies beinhaltet die Kodierung der Distanzmatrix und die Beschreibung des nicht-metrischen Verfahrens von Kruskal.

#### 8.3.2.1 Die Kodierung der Distanzmatrix

Um die MDS auf unsere Fragestellung der Lage von Wörtern im semantischen Umfeld eines bestimmten Stichworts anwenden zu können, muß als erstes eine geeignete Kodierung der Distanzmatrix gefunden werden.
Ausgangspunkt ist eine geordnete Liste der affinsten Objekte, aus der analog zur Komponente der Sterndarstellung eine Auswahl von n Objekten getroffen werden kann. Bei der Berechnung der Distanzmatrix für die anzuordnenden n Objekte wird in den Belegen die mittlere Distanz zwischen den Objekten berechnet. Bei der Ermittlung der Matrix tritt jedoch das Problem auf, daß nicht alle Objekte innerhalb eines Belegs vorkommen und somit nicht miteinander vergleichbar sind. Im Semantischen Inspektor sind daher verschiedene Strategien zum „Auffüllen“ der Belege bzw. zum Berechnen der Distanzmatrix entwickelt worden:

1. Aus den Originalbelegen werden nur solche Wörter betrachtet, die zu den d affinsten Objekten gehören.

   (a) Bei der ersten Methode werden die wirklich vorkommenden Objekte in der Reihenfolge ihrer Positionen links und rechts vom Stichwort angeordnet. Anschließend wird der Beleg mit den nicht vorkommenden Objekten entsprechend ihrer Position in der Affinitätsliste aufgefüllt.

   (b) Bei der zweiten Auffüllmethode wird die Originalposition der tatsächlich vorkommenden Objekte beibehalten und die nicht vorkommenden Objekte werden an die freien Positionen gesetzt.

   Bei beiden Methoden erhalten wir Belege mit einem Radius von mindestens[6] d.

[6] Bei Mehrfachvorkommen eines Objekts wird der Radius größer als d

2. Eine andere Strategie ist, die Originalbelege beizubehalten und bei unvergleichbaren Objekten einen künstlichen Distanzwert für dieses Paar von Objekten zu setzen. Auch hier haben wir zwei Methoden implementiert.

   (a) Für jeden Beleg wird ein hoher Distanzwert bei unvergleichbaren Objekten eingesetzt.

   (b) Nur wenn am Schluß der Berechnung ein Nulleintrag in der Distanzmatrix existiert, wird ein künstlicher Wert für dieses Paar von Objekten eingesetzt.

   Bei diesen beiden Methoden gehen die Originalvergleiche unterschiedlich stark in die Ergebnisse ein.

#### 8.3.2.2 Das Verfahren von Kruskal

Im Semantischen Inspektor wurde das nichtmetrische Verfahren von Kruskal [Kru64a, Kru64b] implementiert, das von allen nichtmetrischen Verfahren am weitesten verbreitet ist. Nichtmetrische Verfahren setzen lediglich voraus, daß zwischen der erhobenen Ähnlichkeitsrangreihenfolge der Objektpaare und den Objektdistanzen folgender monotoner Zusammenhang besteht:

**Definition 8.3 (Monotoniebedingung für nichtmetrische Verfahren)**
*Es sei* $(n, m) \succ (k, j)$*, wenn die Objekt n,m gemäß der Datenerhebung ähnlicher sind als k,j. Falls die Objektpaare gleich ähnlich sind, schreiben wir* $(n, m) \sim (k, j)$*. Ferner bezeichnen wir mit* $d_{n,m}$ *die Objektdistanzen im Merkmalsraum, dann lautet die Monotoniebedingung:*

$$(n, m) \succ (k, j) \Rightarrow d_{n,m} < d_{k,j}$$

$$(n, m) \sim (k, j) \Rightarrow d_{n,m} = d_{k,j}$$

Das ursprüngliche Verfahren von Kruskal ist mittlerweile mehrfach modifiziert worden, der folgende grundsätzliche Ablauf ist jedoch allen Verfahren gemeinsam:

1. Für eine Startkonfiguration $X^0$ aus einem t-dimensionalen Raum ($t \leq N-1$) werden für $q \geq 0$ die $L_q$-Distanzen $d_q^0(n,m) \equiv d_{n,m}^0$ gemäß Definition 8.2 berechnet.

2. Die Distanzen verstoßen in aller Regel gegen die Monotoniebedingung 8.3. Durch eine monotone Regression der Distanzen $d_{n,m}^0$ auf die Ähnlichkeitsordnung aller Objektpaare werden Werte $\delta_{n,m}^0$ bestimmt, die 8.3 erfüllen.

3. Durch den Vergleich der Distanzen $d_{n,m}^0$ mit den Werten $\delta_{n,m}^0$ wird gemessen, wie gut die Konfiguration $X^0$ die Monotoniebedingung erfüllt. Das Maß wird als Stress bezeichnet.

4. Mittels einer Gradientenmethode wird die Konfiguration $X^0$ so verschoben, daß der Stress der neuen Konfiguration $X^1$ möglichst kleiner ist als der Stress von $X^0$.

5. Die Schritte werden solange ausgeführt, bis eine Stopregel in Kraft tritt oder sich der Stresswert nicht mehr ändert.

#### 8.3.2.3 Die Ergebnisdarstellung beim MDS-Verfahren

Die Ergebnisse der Multidimensionalen Skalierung können im Semantischen Inspektor als zweidimensionale Karte graphisch dargestellt werden. Die Bewertung der Güte der Ergebnisse erfolgt dabei mit den beiden folgenden Stressmaßen:

**Definition 8.4 (Stressmaße von Kruskal)**

$$STRESS1 := \sqrt{\frac{\sum_k \sum_l (d_{kl} - \delta_{kl})^2}{\sum_k \sum_l d_{kl}^2}}$$

$$STRESS2 := \sqrt{\frac{\sum_k \sum_l (d_{kl} - \delta_{kl})^2}{\sum_k \sum_l (d_{kl} - \bar{d})^2}}$$

*mit $\bar{d}$ Mittelwert der Distanzen.*

Ob eine Konfiguration gut oder schlecht an die Monotoniebedingung angepaßt ist, wird häufig anhand der Werte aus Tabelle[7] 8.1 überprüft. Bei der Verwendung der Faustregel muß man jedoch beachten, daß der Stresswert auch entscheidend von der Dimension, der Objektanzahl und dem Distanzmaß abhängig ist. Grundsätzlich erscheinen formale Kriterien zweitrangig, wenn sich die Objektkonfiguration gut interpretieren läßt.

Tabelle 8.1: Anhaltspunkt für die Güte der Anpassung

| Anpassungsgüte | STRESS1 | STRESS2 |
|---|---|---|
| gering | 0.2 | 0.4 |
| ausreichend | 0.1 | 0.2 |
| gut | 0.015 | 0.1 |
| ausgezeichnet | 0.025 | 0.15 |
| perfekt | 0 | 0 |

Um die Leistungsfähigkeit des implementierten MDS-Verfahren zu demonstrieren, möchte ich als erstes Beispiel eine konstruierte Belegsammlung für das Stichwort *Bank* betrachten. Dabei erzeugten wir Belege für drei Bedeutungsvarianten des Wortes Bank:

1. Bank als Sitzmöbel
2. Bank als Institution
3. Bank als Gebäude

Zu jeder Variante wählten wir dann willkürlich 20 Wörter, die das jeweilige semantische Umfeld darstellen sollen. Während die Menge der Wörter, die zu der ersten Bedeutungsvariante „Sitzmöbel“ gehören, disjunkt zu den beiden anderen Mengen ist, haben die Wortmengen der beiden anderen Bedeutungsvarianten 10 gemeinsame Elemente.

[7] Tabelle aus [Bac96]

1. $SITZMÖBEL := \{hart, Park, Baum, sitzen, Sonne, Schule, drücken, Lehrer, liegen, Clochard, schlafen, Kirche, Tisch, Stuhl, Anklage, Opposition, ausruhen, Auswechsel, Rück, Holz\}$

2. $INSTITUTION := \{zur, Wechselkurs, Aktie, Finanzen, Devisen, verzinsen, Kredit, Hypothek, Noten, Bundes, Konto, Sparbuch, Dresdener, Post, Sparkasse, abheben, Geld, D\_Mark, umtauschen, Kleingeld\}$

3. $GEBÄUDE := \{in, Fassade, Gebäude, hoch, groß, auf, Scheine, Münzen, einzahlen, Check, Konto, Sparbuch, Dresdener, Post, Sparkasse, abheben, Geld, D\_Mark, umtauschen, Kleingeld\}$

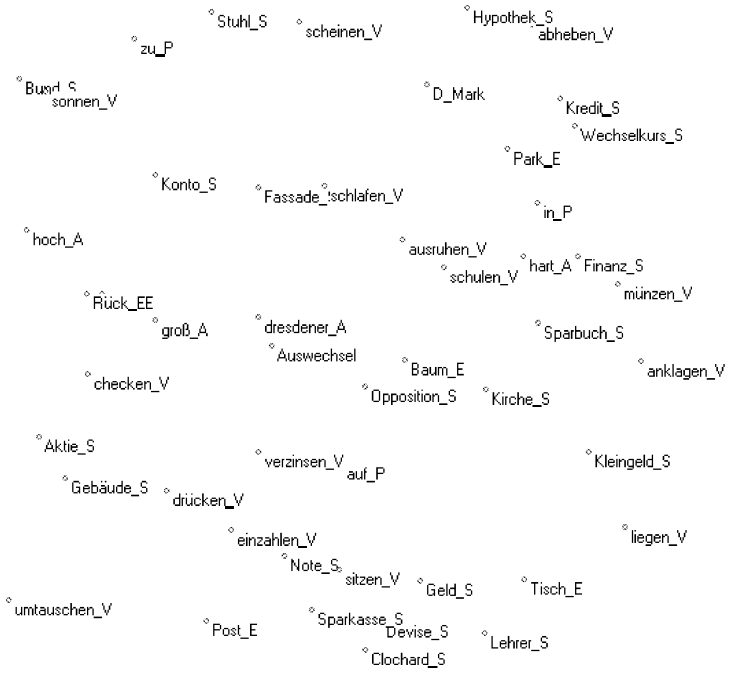


Abbildung 8.3: 1. Iteration bei der MDS für den Kunstbeleg *Bank*

Anschließend wurden 500 Belege mit Radius 5 erzeugt, wobei für jeden Beleg zufällig eine der drei Mengen ausgewählt wurde und aus dieser wiederum zufällig 10 Wörter, die dann zusammen mit dem Stichwort in der Mitte den Beleg bildeten.

Da die Reihenfolge im Beleg willkürlich gewählt wurde und damit die Distanzen zum Stichwort keine Rolle spielen, wurde die Affinitätsberechnungen mit dem Parameter $\lambda = 0$ durchgeführt.
Abbildung 8.3 zeigt das Ergebnis nach einer Iteration, es lassen sich noch keine Gruppen oder Cluster erkennen. In der nächsten Abbildung 8.4 ist das Ergebnis nach 1001 Iterationen dargestellt.

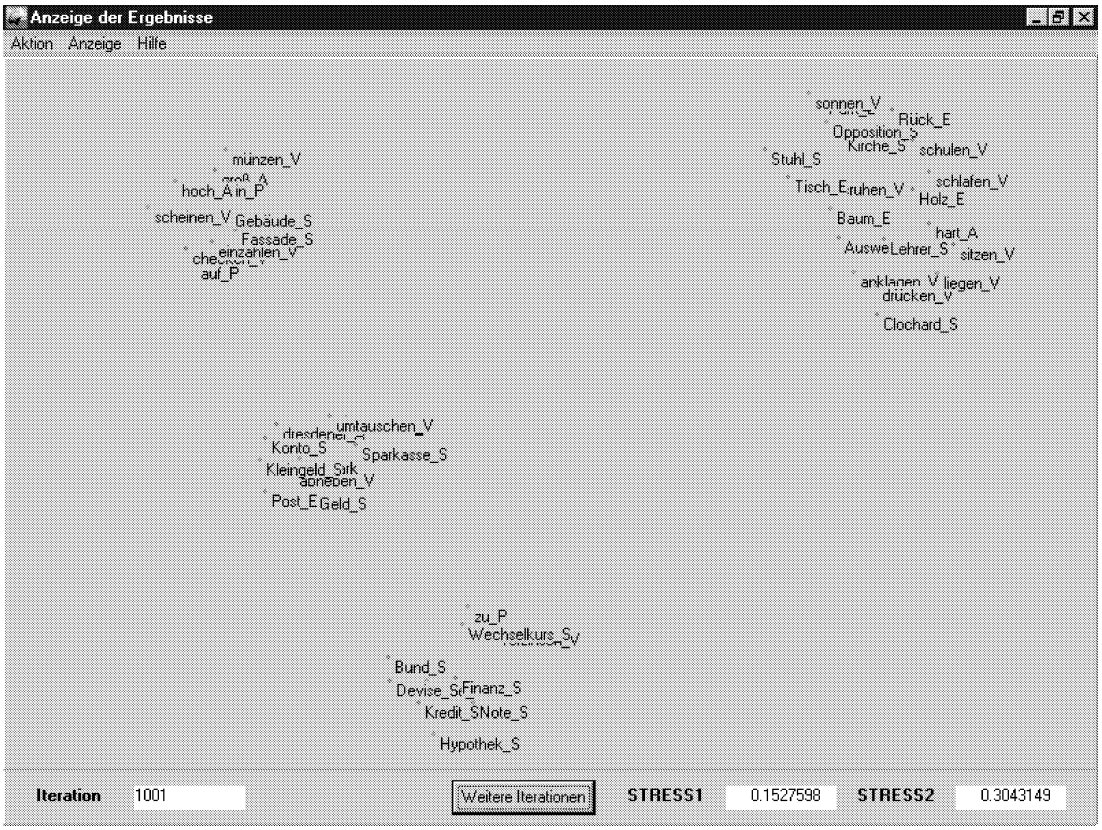


Abbildung 8.4: Ergebnis nach 1001 Iterationen

Nun haben sich vier Cluster herausgebildet, die folgende Eigenschaften aufweisen:

1. Der Cluster in der oberen rechten Hälfte besteht aus genau der Menge von Wörtern, die für die Bedeutungsvariante „Sitzmöbel“ definiert wurde.

2. Die drei Cluster in der linken Hälfte liegen nahe zusammen, sind aber weit vom ersten Cluster entfernt.

3. Der mittlere Cluster enthält genau die zehn Wörter, die in den beiden Mengen INSTITUTION und GEBÄUDE vorkommen.

4. Die beiden anderen Cluster bestehen aus den jeweils anderen zehn Wörtern.

Insgesamt entspricht dieses Ergebnis unserer Vorstellung, daß die beiden Bedeutungsvarianten „Institution“ und „Gebäude“ mehr Gemeinsamkeiten haben als die Bank als „Sitzmöbel“
Die Stresswerte sind nach der Faustregel ausreichend[8].

Obwohl die Resultate für die künstlich erzeugten Belegsammlungen vielversprechend erschienen, konnten wir bisher diese Ergebnisse mit „realistischen“ Beispielen noch nicht erzielen, d.h. es bildeten sich keine klar abgrenzbaren Cluster. Die erhaltenen Anordnungen der Objekte hingen stark davon ab, wie das Problem der „Missing Values“ (d.h. Unvergleichbarkeit der Objekte in einem Beleg) gelöst wurde (s.a. Abschnitt 8.3.2.1). Bei Verwendung einer Auffüllmethode, d.h. in den Belegen werden nicht vorkommende Objekte ergänzt, erreicht das System sehr gute Stresswerte. Diese Konfigurationen hoher Güte haben jedoch wenig Aussagekraft, da sich sämtliche Objekte zu einem Cluster verdichten. Die Zuordnung eines festen (hohen) Affinitätswerts bei nicht vergleichbaren Objekten führte trotz einer großen Anzahl von Iterationen zu schlechten Stresswerten, woraus wir schließen mußten, daß die Anzahl der realistischen Vergleiche zu gering war.
Wir haben viele Testläufe mit dem Stichwort *Bank* durchgeführt, da zwei semantische Varianten von Bank sich auch morphologisch im Plural unterscheiden: die Bank als Institution (Plural Banken) und die Bank zum Verweilen (Plural Bänke). Bei der Untersuchung verschiedener Belegsammlungen stellten wir fest, daß bei dem vorhandenen Textmaterial die Kontexte mit Bank in der Bedeutung Institution überwogen. Daher wurde eine spezielle Belegsammlung von Prof. Heringer erzeugt, bei der Banken und Bänke gleich gewichtet waren. Die Belegsammlung

[8] In Testläufen konnten die Stresswerte durch mehr Iterationen noch erheblich verringert werden. Ich habe diese Stufe gewählt, da nach mehr Iterationen die Labels in der Graphik stärker überlagerten.

bestand aus 1224 Belegen mit Radius 8. Auch diese Manipulation brachte nicht die gewünschten Ergebnisse. Wir stellten fest, daß die Bedeutungsvariationen bei Bänken so stark sind, daß auch bei Gleichgewichtung sich noch keine interpretierfähigen Cluster herausbildeten.

# Kapitel 9

# Ausblick

Der Semantische Inspektor ist als ein Werkzeug der quantitativen Linguistik zu betrachten. Er erlaubt die graphische Darstellung der affinsten Objekte eines bestimmten Stichworts. Der Versuch, die Objekte selbst zueinander in Beziehung zu setzen, wurde mittels der Multidimensionalen Skalierung (MDS) und dem konnektionistischen[1] Verfahren der Selbstorganisierenden Karten (SOM) durchgeführt.

Die bisher erzielten Ergebnisse sind von linguistischer Seite noch nicht zufriedenstellend. Ein Grund liegt sicherlich in der Lemmatisierung der Belege. Wir erhoffen uns durch den Einsatz von $\mathcal{PARLEX}$ oder auch externen Programmen wie GERTWOL[2] bessere Ergebnisse.

Mit dem konnektionistischen Verfahren der selbstorganisierenden Karten wurde ein weiteres Strukturen-entdeckendes Verfahren integriert, das im Gegensatz zur MDS den Schwerpunkt mehr auf die Erhaltung der lokalen Strukturen legt. Der Vergleich[3] der beiden Ergebnisse sollte aus linguistischer Sicht geschehen.

Weiterhin schlagen wir vor, die Verfahren des Semantischen Inspektors auf speziellere Textsammlungen wie Fachkorpora, in denen wahrscheinlich die Variation

[1] In dieser Arbeit nicht beschrieben, siehe [Mü98]

[2] German-Two-level System von Lingsoft [Lin98]. Die Arbeitsgruppe von Prof. Heringer experimentiert mit diesem kommerziell zur Verfügung stehenden Morphologie-Tool, welches auf der theoretischen Grundlage der 2-Ebenen Morphologie basiert.

[3] Dieser Vergleich konnte in dieser Arbeit nicht mehr angestellt werden, da das Programm PolySom den Linguisten erst seit kurzem zur Verfügung steht.

der Bedeutungen eines Stichworts geringer ist, anzuwenden.

# Kapitel 10

# Schlußbemerkungen

In dieser Arbeit wurde im ersten Teil die Repräsentation und Akquisition von natürlichsprachlichem Wissen für die maschinelle Verabreitung durch einen Parser behandelt. Es wurde ein Konzept für den natürlichsprachlichen Parser vorgestellt, das auf einer Erweiterung der monadischen Logik zweiter Stufe beruht. Wir konnten zeigen, daß für das Kalkül ein äquivalentes Automatenmodell existiert, wobei der konstruktive Äquivalenzbeweis als Grundlage für die Konzeption des Parsers dient. Durch die Konzeption des Parsers und die Form der lexikalischen Wissensbasis kann in dem System $\mathcal{PARLEX}$ nicht nur das lexikalische Wissen dynamisch erweitert und verändert werden, sondern auch die Regeln der Grammatik lassen sich erweitern bzw. modifizieren.

Die lexikalische Wissensbasis beinhaltet morphosyntaktische Informationen, aber auch semantische Beschreibungen. Durch die Integration von Valenzstrukturbeschreibungen auf syntaktischer und semantischer Ebene bietet sich eine Parsing-Strategie an, die das Verb mit seinen Valenzen in den Mittelpunkt stellt.

In der Vollformengenerierungskomponente wurde festgelegt, wie das morphosyntaktische und semantische Wissen für die Weiterverarbeitung durch den Parser kodiert ist. Durch die Trennung von lexikalischer Wissensbasis und Vollformendatenbank sind wir in der Lage, beliebige Ausgabeformate durch leichte Modifikationen in der Generierungskomponente zu erzeugen, und somit das Wissen für

andere Systeme zur Verfügung zu stellen[1]

Bei der Akquisition wurde in dieser Arbeit das Hauptaugenmerk auf die manuelle Erfassung und Modifikation durch den menschlichen Benutzer gelegt. Es wurde eine Umgebung geschaffen, in der Wissen ohne die Kenntnis der internen Strukturen eingetragen und modifiziert werden kann. Auch die Konsistenz der Wissensbasis und der Vollformendatenbank ist durch das System gewährleistet. Im Rahmen der Akquisition wurde weiterhin ein Algorithmus entwickelt, der auf der Grundlage einer Liste von (regelmäßigen) Verben die Wissensbasis um etwa 3000 Einträge erweiterte. Hinsichtlich der teilautomatischen Erfassung von lexikalischem Wissen sind Erweiterungen des Systems wünschenswert.

Der zweite Teil meiner Arbeit behandelte mit dem Semantischen Inspektor ein Werkzeug für die quantitative Linguistik. Durch die Einbettung verschiedener Darstellungsmethoden hoffen wir, Sprachwissenschaftlern ein Werkzeug an die Hand geben zu können, das einen Beitrag zur Beschreibung von semantischen Aspekten leisten kann.

Auf den ersten Blick scheinen die Gemeinsamkeiten beider Systeme bei der Notwendigkeit der Lemmatisierung zu enden. Wir glauben jedoch, daß beide Systeme voneinander profitieren können. Durch den Einsatz des lexikalischen Parsers im Semantischen Inspektor könnten spezifischere und qualitativ bessere Belegsammlungen erzeugt werden, z.B. kann durch eine zusätzliche syntaktische Analyse eine sicherere Wortklassenzuordnung durchgeführt werden.

Aber auch umgekehrt könnte die Wissensbasis von $\mathcal{PARLEX}$ qualitativ aufgewertet werden. Betrachten wir nämlich die Repräsentation der Verbvalenzen, so ist in der lexikalischen Wissensbasis vorgesehen, daß die Umgebung eines Verbs auch durch die Angabe von Wortlisten beschrieben werden kann. Die Sterndarstellung des Semantischen Inspektors könnte als Grundlage für die Erzeugung

[1] Die Wiederverwendbarkeit lexikalischer Ressourcen ist meines Erachtens jedoch nur soweit möglich, wie die linguistischen „Grundideen“ übereinstimmen. Ich teile den Optimismus vieler Autoren in dieser Hinsicht nicht.

solcher Wortlisten dienen, indem die affinsten Nomen eines Verbs für die Erstellung der Listen herangezogen werden. Dabei ist festzustellen, daß bei der jetzigen Version die Sterndarstellung nur zur Visualisierung möglicher Kandidaten dienen kann. Da die dargestellten Objekte jedoch Lemmata repräsentieren, müßte manuell entschieden werden, welche der Objekte zu einer Nominalphrase mit bestimmtem Kasus zählen könnte. Denkbar wäre jedoch auch eine Version, bei der die Nomen nicht lemmatisiert werden, so daß eine direkte Zuordnung zu einem Slot mit bestimmtem Kasus möglich wäre.

# Anhang A

# Kodierung der morphologischen Merkmale

In dem folgenden Anhang werden für die veränderlichen Wortklassen die Kodierung der morphologischen Merkmale und die berücksichtigten Subklassen nochmals aufgeführt.

## A.1 Übersicht

Tabelle A.1: Morphologische Merkmale und ihre Kodierung

| Merkmalstyp | Merkmal | Wortklassen | Code |
|---|---|---|---|
| **Kasus** | nominativ | Nomen, Adjektiv, Determinierer, Pronomen, Präpositionen | n |
| | akkusativ | | a |
| | dativ | | d |
| | genitiv | | g |
| | unspezif. | | u |
| **Person** | 1 | Pronomen, Verb | 1 |
| | 2 | | 2 |
| | 3 | | 3 |
| | unspez. | | 0 |
| **Numerus** | singular | Adjektive, Determinierer, Pronomen, Verb | e |
| | plural | | m |
| **Genus** | fem | Adjektive, Determinierer, Pronomen, Nomen | F |
| | mask | | M |
| | neut | | N |
| | unspez. | | U |
| **Definitheit** | definit | Determinierer, Nomen | D |
| | indefinit | | o |
| | nullartikel | | z |
| | unspezif. | | * |
| **Flexion** | stark | Adjektiv | h |
| | schwach | | w |
| | gemischt | | t |
| | prädikativ | | v |
| **Steigerungs-grad** | positiv | Adjektiv, Adverb | p |
| | komparativ | | k |
| | superlativ | | s |
| **Tempus** | Präsens | Verb | G |
| | Präteritum | | V |
| | unspez. | | O |
| **Modus** | indikativ | Verb | i |
| | konjunktiv | | c |
| | imperativ | | b |
| **infinite Formen** | partizip 1 | Verb | E |
| | partizip 2 | | Z |
| | infinitiv | | I |

# A.2 Die Wortklasse Verb (V)

## A.2.1 Morphologische Merkmale

| Merkmalsbündel | (Person,Numerus,Tempus,Modus) | | | | | |
|---|---|---|---|---|---|---|
| | **Singular** | | | **Plural** | | |
| | **1. Pers.** | **2. Pers.** | **3. Pers.** | **1. Pers.** | **2. Pers.** | **3. Pers.** |
| Präsens indikativ | 1eGi | 2eGi | 3eGi | 1mGi | 2mGi | 3mGi |
| Präsens konjunktiv | 1eGc | 2eGc | 3eGc | 1mGc | 2mGc | 3mGc |
| Prät. indikativ | 1eVi | 2eVi | 3eVi | 1mVi | 2mVi | 3mVi |
| Prät. konjunktiv | 1eVc | 2eVc | 3eVc | 1mVc | 2mVc | 3mVc |
| Imperativ | | 2eOb | | 1mOb | 2mOb | |
| Infinitiv | I | | | | | |
| Partizip I | E | | | | | |
| Partizip II | Z | | | | | |

## A.2.2 Subklassen

- Hauptverb (HV)
- Kopulaverb (KV)
- Modalverb (MV)
- Hilfsverb (AV)

## A.3 Die Wortklasse Nomen (N)

### A.3.1 Morphologische Merkmale

| Merkmalsbündel | (Kasus,Numerus,Genus,Definitheit) | | | |
|---|---|---|---|---|
| | **Nominativ** | **Akkusativ** | **Dativ** | **Genitiv** |
| **Feminin Singular** | neF* | aeF* | deF* | geF* |
| **Feminin Plural** | nmF* | amF* | dmF* | gmF* |
| **Maskulin Singular** | neM* | aeM* | deM* | geM* |
| **Maskulin Plural** | nmM* | amM* | dmM* | gmM* |
| **Neutrum Singular** | neN* | aeN* | deN* | geN* |
| **Neutrum Plural** | nmN* | amN* | dmN* | gmN* |

Das Merkmal der Definitheit variiert nur bei nominalisierten Adjektiven wie z.B *Angestellte*.

# A.4 Die Wortklasse Pronomen (PR)

## A.4.1 Morphologische Merkmale

| Mermalsbündel | (Kasus,Person,Numerus,Genus) | | | |
|---|---|---|---|---|
| | **Nominativ** | **Akkusativ** | **Dativ** | **Genitiv** |
| **1. Pers. Sing** | neU1 | aeU1 | deU1 | geU1 |
| **2. Pers. Sing** | neU2 | aeU2 | deU2 | geU2 |
| **3. Pers. Sing Fem** | neF3 | aeF3 | deF3 | geF3 |
| **3. Pers. Sing Mask** | neM3 | aeM3 | deM3 | geM3 |
| **3. Pers. Sing Neut** | neN3 | aeN3 | deN3 | geN3 |
| **1. Pers. Plur** | nmU1 | amU1 | dmU1 | gmU1 |
| **2. Pers. Plur** | nmU2 | amU2 | dmU2 | gmU2 |
| **3. Pers. Plur** | nmU3 | amU3 | dmU3 | gmU3 |

## A.4.2 Subklassen

- Einfaches Pronomen (pro)
- Personalpronomen (prs)
- Definitpronomen (def)
- Indefinitpronomen (ind)
- Interrogativpronomen (int)
- Reflexivpronomen (rfl)
- Possessivpronomen (pss)
- Demonstrativpronomen (dem)

# A.5 Die Wortklasse Adjektiv (A)

## A.5.1 Morphologische Merkmale

| Merkmalsbündel | (Kasus,Genus,Numerus,Flexion,Steigerungsgrad) | | | |
|---|---|---|---|---|
| | | **Fem** | **Mask** | **Neut** |
| | **Singular** | | | |
| stark | Nominativ | nFehp | nMehp | nNehp |
| | Akkusativ | aFehp | aMehp | aNehp |
| | Dativ | dFehp | dMehp | dNehp |
| | Genitiv | gFehp | gMehp | gNehp |
| schwach | Nominativ | nFewp | nMewp | nNewp |
| | Akkusativ | aFewp | aMewp | aNewp |
| | Dativ | dFewp | dMewp | dNewp |
| | Genitiv | gFewp | gMewp | gNewp |
| gemischt | Nominativ | nFetp | nMetp | nNetp |
| | Akkusativ | aFetp | aMetp | aNetp |
| | Dativ | dFetp | dMetp | dNetp |
| | Genitiv | gFetp | gMetp | gNetp |
| | **Plural** | | | |
| stark | Nominativ | nFmhp | nMmhp | nNmhp |
| | Akkusativ | aFmhp | aMmhp | aNmhp |
| | Dativ | dFmhp | dMmhp | dNmhp |
| | Genitiv | gFmhp | gMmhp | gNmhp |
| schwach | Nominativ | nFmwp | nMmwp | nNmwp |
| | Akkusativ | aFmwp | aMmwp | aNmwp |
| | Dativ | dFmwp | dMmwp | dNmwp |
| | Genitiv | gFmwp | gMmwp | gNmwp |
| gemischt | Nominativ | nFmtp | nMmtp | nNmtp |
| | Akkusativ | aFmtp | aMmtp | aNmtp |

Fortsetzung nächste Seite

| | | | | Fortsetzung Plural |
|---|---|---|---|---|
| Merkmalsbündel | (Kasus,Genus,Numerus,Flexion,Steigerungsgrad) | | | |
| | | **Fem** | **Mask** | **Neut** |
| | Dativ | dFmtp | dMmtp | dNmtp |
| | Genitiv | gFmtp | gMmtp | gNmtp |

Die Tabelle zeigt die Kodierung für die Positiv-Form. Die letzte Position ist ansonsten mit k (komparativ) oder s (superlativ) kodiert.

# A.6 Wortklasse Determinierer (D)

## A.6.1 Morphologische Merkmale

| Mermalsbündel | (Kasus,Numerus,Genus,Definitheit,Flexion) | | | | |
|---|---|---|---|---|---|
| | | **Fem** | **Mask** | **Neutrum** | **Unspez.** |
| Singular | Nominativ | neFDw | neMDw | neNDw | neUDw |
| | Akkusativ | aeFDw | aeMDw | aeNDw | aeU |
| | Dativ | deFDw | deMDw | deNDw | deUDw |
| | Genitiv | geFDw | geMDw | geNDw | geUDw |
| Plural | Nominativ | nmUDw | | | |
| | Akkusativ | amUDw | | | |
| | Dativ | dmUDw | | | |
| | Genitiv | gmUDw | | | |

In der Tabelle ist das Merkmal der Definitheit auf *definit* (D) gesetzt[1]. Das Merkmal der Flexion gibt an, wie ein nachfolgendes Adjektiv flektiert sein muß (stark (h), schwach (w) oder gemischt (t)).

## A.6.2 Subklassen

- Definite Artikel (det-def)
- Demonstrativ-Determinierer (det-dem)
- Possessiv-Determinierer (det-pss)
- Interrogativ-Determinierer (det-int)
- Definite Quantoren (qnt-def)
- Indefinite Quantoren (qnt-ind)
- Negationen (qnt-neg)

[1] indefinit wird mit dem Charakter o kodiert.

# Anhang B

# Notation der Entity-Relationship Modelle

Die Entity-Relationship Methode zählt zu den Datenmodellierungsmethoden und geht auf Arbeiten von Chen (siehe [Che76, CK91]) zurück. Da sich seit Einführung der Methode verschiedene Darstellungsformen etabliert haben, stelle ich im folgenden die in dieser Arbeit verwendete Notation[1] vor.
Generell unterscheidet man bei ER-Modellen zwischen Entitäten und Beziehungen:

**Definition B.1 (Entität)**
*Eine Entität ist ein Objekt, das wirklich oder imaginär vorhanden ist und über das Informationen gesammelt oder gespeichert werden müssen.*

**Definition B.2 (Beziehung)**
*Eine Beziehung ist eine wichtige Verbindung zwischen zwei Entitäten.*

Jeder Entität werden Attribute zugeordnet, die dazu dienen, die Entität zu beschreiben. Dabei unterscheiden wir zwischen fakultativen und obligatorischen Attributen.
Graphisch stellen wir eine Entität durch eine Softbox mit dem Namen in Großbuchstaben dar. Eine Beziehung wird durch eine Linie dargestellt, die zwei Entity-

[1] Diese entspricht dem Standard, der in den CASE-Werkzeugen der Firma ORACLE verwendet wird. Eine ausführliche Beschreibung findet sich in [Bar92].

Boxen miteinander verbinden. Dabei gibt die Art der Linie Auskunft über die Optionalität und den Grad der Beziehung. Die häufigste Beziehung ist eine 1:N-Beziehung, wie sie in Abbildung B.1 dargestellt ist.

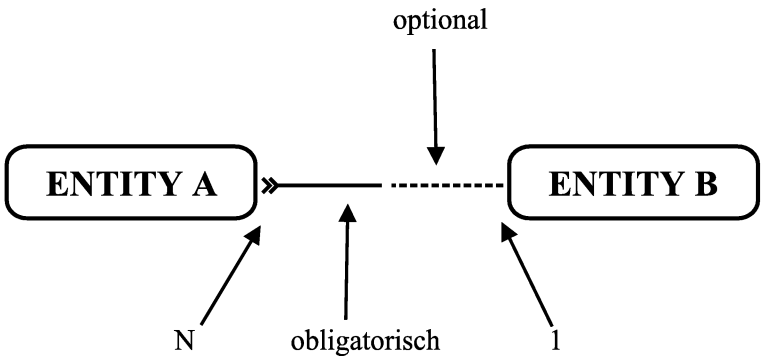


Abbildung B.1: Graphische Darstellung von Entität und Beziehung

Das Diagramm bedeutet von links nach rechts gelesen:

- Jede Entity vom Typ A muß zu genau einer Entity vom Typ B in Beziehung stehen.
- Jede Entity B kann zu einer oder mehreren Entities vom Typ A eine Beziehung haben.

Zum besseren Verständnis der Diagramme können auch die Beziehungen benannt werden, hierbei sollten Verben zur Beschreibung verwendet werden.
Neben 1:N-Beziehungen und 1:1-Beziehungen haben wir bei der Beschreibung der lexikalischen Wissensbasis noch einen speziellen Typ der Exklusivität bzw. „entweder-oder“-Beziehung benutzt. Dieser Typ wird verwendet, wenn zwei oder mehr Beziehungen derselben Entität sich gegenseitig ausschließen. In Abbildung B.2 ist eine binäre „Exklusiv-Oder“-Beziehung dargestellt, die folgende Semantik hat:

- Jede Entity vom Typ A muß entweder genau eine Beziehung zu einer Entity vom Typ B haben oder genau eine Beziehung zu einer Entity vom Typ C haben.

- Eine Entity vom Typ B kann zu einer oder mehreren Entities vom Typ A eine Beziehung haben.
- Eine Entity vom Typ C kann zu einer oder mehreren Entities vom Typ A eine Beziehung haben.

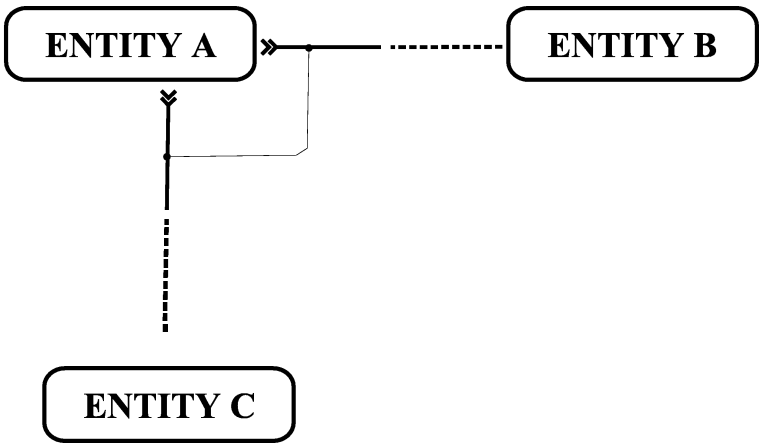


Abbildung B.2: „Exklusiv-Oder“-Beziehung

Eine häufig in der Praxis verwendete Beziehung, ist die „Many-to-Many“-Beziehung. Aufgrund von Normalisierungsbedingungen muß dieser Typ von Beziehung später jedoch aufgelöst werden. In den ER-Modellen der Lexikalischen Wissensbasis haben wir von Beginn an auf diese Art von Beziehung verzichtet, bzw. sofort diese wie in Abbildung B.3 durch die Einführung einer Verbindungsentität aufgelöst.

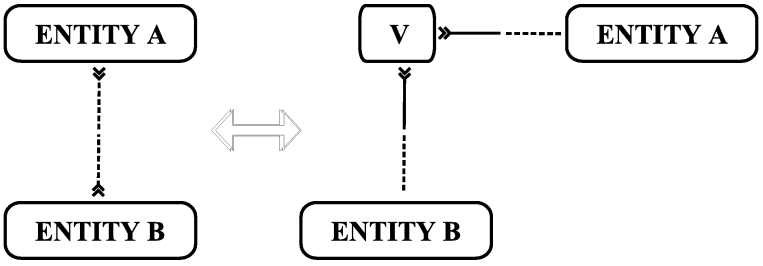


Abbildung B.3: „Many-to-Many“-Beziehung und ihre Auflösung

# Anhang C

# Systemvoraussetzungen und Implementierung

Beide Systeme laufen unter den Betriebssystemen Windows 95 oder Windows NT auf Pentium-Rechnern mit mindestens 32 MB Hauptspeicher. Die Entscheidung für diese Systemumgebung wurde durch die intendierte Benutzergruppe der Sprachwissenschaftler motiviert. Die Implementierung erfolgte unter Delphi 3 der Firma Borland[1]. In dieser objektorientierten visuellen Programmierumgebung (Sprache Object Pascal) ist mit der Borland Database Engine (BDE) eine Komponente integriert, über deren Schnittstelle die verschiedensten Datenbanken angesprochen werden können.

Bei der lexikalischen Wissensbasis von $\mathcal{PARLEX}$ wurde ein relationales Datenbankmodell verwendet, dessen Implementierung auf dem lokalen SQL-Server (Local InterBase Server) erfolgte. Diese kostengünstige Lösung kann durch die Verwendung der BDE ohne großen Aufwand in eine Client-Server Applikation überführt werden, bei der die lexikalische Wissensbasis zentral auf einem Server gepflegt wird.

Die Einbettung der Parser-Komponente konnte exemplarisch demonstriert wer-

[1] Seit April 1998 Inprise.

den. Der Parser wird von Paola Glavan[2] in der Programmiersprache C/C++ unter Unix implementiert. Wir haben einen rudimentären Demonstrator der Parser-Komponente unter Verwendung der GNU-WIN32 Tools [Cyg] recompiliert und in das System eingebettet. Die Kommunikation erfolgt wie bereits beschrieben file-basiert (siehe Abschnitt 5.6).

Beim Semantischen Inspektor ist die Abgleichtabelle zur Lemmatisierung als Desktop-Datenbank im Paradox-Format realisiert. Die externen Programme[3] zur Erzeugung der Belegsammlungen werden über die integrierte DOS-Shell des Betriebssystems angesteuert.

[2]Universität Rijeka

[3]Die beiden Programme Extract und Except wurden von Markus Ohlenroth (Universität Augsburg) in Assembler implementiert.

# Literaturverzeichnis


[Ant90] Evan L. Antworth. PC-KIMMO: A Two-Level Processor for Morphological Analysis. Technical report, Summer Institute of Linguistics, Dallas, Texas, 1990.

[AZ94] B. T. S. Atkins und A. Zampolli (Hrsg.). *Computational Approaches to the Lexicon.* Oxford University Press, 1994.

[Bac96] Klaus Backhaus. *Multivariate Analysemethoden.* Springer, Heidelberg, 1996.

[Bar92] Richard Barker. *CASE METHOD: Entity-Relatonship-Modellierung.* Addison-Wesley, 1992.

[BB89] Bran Boguraev und Ted Briscoe (Hrsg.). *Computational Lexicography for Natural Language Processing.* Longman, London and New York, 1. Auflage, 1989.

[BBR88] G. Edward Barton, Robert C. Berwick und Eric Sven Ristad. *Computational complexity and natural language.* MIT Press, 2. Auflage, 1988.

[Bre88] Joan Bresnan (Hrsg.). *The mental represenation of grammatical relations.* MIT Press, 3. Auflage, 1988.

[BS77] Henning Bergenholtz und Burkhard Schaeder (Hrsg.). *Die Wortarten des Deutschen: Versuch einer syntaktisch orientierten Klassifikation.* Klett, Stuttgart, 1. Auflage, 1977.

[Bü62] J. R. Büchi. On a decision method in restricted second-order arithmetics. In Nagel [Nag62], Seite 1–11. International Congress 1960.

[Bur96] Gerrit Burkert. *Repräsentation von lexikalisch-semantischem Wissen in einem System zur Verarbeitung natürlicher Sprache*, Band 141 von *Dissertationen zur künstlichen Intelligenz*. Infix, 1996.

[Che76] Peter P.S. Chen. The Entity-Relationship Model - Toward a Unified View of Data. *ACM TODS*, 1:9–36, 1976.

[Chu80] Kenneth Ward Church. On Memory Limitations in Natural Language Processing. Technical Report MIT/LCS/TR-245, MIT, Laboratory of Computer Science, 1980. zitiert in [Ing95].

[Chu88] K. W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, Seite 136–143, 1988.

[CHu95a] The Text Encoding Initiative: Background and Context. *Computers and the Humanities*, 29(1), 1995.

[CHu95b] The Text Encoding Initiative: Background and Context. *Computers and the Humanities*, 29(2), 1995.

[CHu95c] The Text Encoding Initiative: Background and Context. *Computers and the Humanities*, 29(3), 1995.

[CK91] Peter P.S. Chen und Heinz-Dieter Knöll. *Der Entity-Relationship-Ansatz zum logischen Systementwurf : Datenbank- und Programmentwurf* . BI-Wiss.-Verl., 1991.

[CKPS92] D. Cutting, J. Kupiec, J. Pedersen und P. Sibun. A practical part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, Seite 133–140, 1992.

[Cyg] Cygnus solution. The GNU-WIN32 Project. http://www.cygnus.com/misc/gnu-win32/.

[DLZ85] David R. Dowty, Karttunen Lauri und Arnold M. Zwicky (Hrsg.). *Natural Language Parsing*. Cambridge University Press, 1985.

[Eng91] Ulrich Engel. *Deutsche Grammatik*. Julius Groos Verlag Heidelberg, 2. Auflage, 1991.

[FBS95] Wolfgang Fleischer, Irmhild Barz und Marianne Schröder. *Wortbildung der deutschen Gegenwartssprache*. Niemeyer, Tübingen, 2.. Auflage, 1995.

[FH96] Helmut Feldweg und Erhard W. Hinrichs (Hrsg.). *Lexikon und Text - Wiederverwendbare Methoden und Ressourcen zur linguistischen Erschließung des Deutschen*. Max Niemeyer Verlag, 1996.

[Geb78] Heiko Gebauer. *Montague Grammatik: Eine Einführung mit Anwendungen auf das Deutsche*, Band 24 von *Germanistische Arbeitshefte*. Niemeyer, Tübingen, 1978.

[GKPS85] Gerald Gazdar, Ewan Klein, Geoffrey Pullum und Ivan Sag. *Generalized Phrase Structure Grammar*. Harvard Univ. Press, 1985.

[Gö95] Günther Görz (Hrsg.). *Einführung in die künstliche Intelligenz*. Addison-Wesley, 2. Auflage, 1995.

[GW95] Günther Görz und Wolfgang Wahlster. Sprachverarbeitung: Einleitung und Überblick. In Goerz [Gö95].

[HE95] Joachim Hartung und Bärbel Elpelt. *Multivariate Statistik: Lehr- und Handbuch der angewandten Statistik*. Oldenbourg Verlag GmbH, 5. Auflage, 1995.

[Hel92] G. Helbig. *Probleme der Valenz- und Kasustheorie*, Band 51 von *Konzepte der Sprach- und Literaturwissenschaft*. Max Niemeyer Verlag, 1992.

[Her81] Hans Jürgen Heringer. Die Unentscheidbarkeit der Ambiguität. de Gruyter, 1981.

[Her96] Hans Jürgen Heringer. *Deutsche Syntax Dependentiell.* Stauffenburg Verlag, 1996.

[HS91] Gerhard Helbig und Wolfgang Schenkel. *Wörterbuch zur Valenz und Distribution deutscher Verben.* Max Niemeyer Verlag, 8. Auflage, 1991.

[HSSS78] Gary G. Hendrix, Earl D. Sacerdoti, Danial Sagalowicz und Jonathan Slocum. Developing a Natural Language Interface to Complex Data. *ACM Transaction on Database Systems*, 3(2):105–147, 1978. Sekundärliteratur aus [Ing95].

[Ing95] Robert F. P. Ingria. Lexical Information for Parsing Systems: Points of Convergence and Divergence. In Walker et al. [WZC95], Seite 93–169.

[Jac94] Joachim Jacobs. *Kontra Valenz*, Band 12 von *Fokus: linguistisch-philologische Studien.* WVT, Trier, 1994.

[Jos87] A. Joshi. An Introduction to Tree Adjoining Grammars. In Alexis Manaster-Ramer (Hrsg.), *Mathematics of language*, Seite 87–114. Benjamins, 1987.

[Kar83] Lauri Karttunen. KIMMO: A general morphological processor. *Texas Linguistic Forum*, 22:165–186, 1983.

[Kar93] Lauri Karttunen. Finite-state lexicon compiler. Technical report, Xerox Palo Alto Research Center, 1993.

[Kas97] Samuel Kaski. *Data Exploration Using Self-Organizing Maps.* Doctor of technology, Helsinki University of Technology, Neurol Networks Research Centre, 1997.

[Kay84] M. Kay. Functional Unification Grammar: a formalism for machine translation. In *10th International Conference on Computational Linguistics, 22nd Annual Meeting of the Association for Computational Linguistics: proceddings of Coling 84*, 1984.

[KB88] Ronald M. Kaplan und Joan Bresnan. *Lexical Functional Grammar: A Formal System for Grammatical Representation*, Kapitel 4, Seite 173–281. In Bresnan [Bre88], 3. Auflage, 1988.

[Kem93] A. Kempe. A probabilistic tagger and an analysis of tagging errors. Technical report, Universität Stuttgart, Institut für maschinelle Sprachverarbeitung, 1993.

[KK85] Lauri Karttunen und Martin Kay. Parsing in a free word order language. In Dowty et al. [DLZ85].

[Koh97] Teuvo Kohonen. *Self-Organizing maps*. Springer, Heidelberg, 1997.

[Kos83] Kimmo Koskenniemi. *Two-level morphology : a general computational model for word-form recognition and production*. Publications of the Department of General linguistics, University of Helsinki ; 11, Univ. Helsinki, 1983.

[KR93] Reinhard Köhler und Burghard B. Rieger (Hrsg.). *Contributions to Quantitative Linguistics*. Kluwer Academic Publisher, 1993. Proceedings of the First International Conference on Quantitative Linguisticts, Trier 1991.

[Kru64a] J.B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric metrika. *Psychometrika*, 29:1–27, 1964.

[Kru64b] J.B. Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29:115–129, 1964.

[Lin98] Lingsoft. GERTWOL - German Morphological Analyser. http://www.lingsoft.fi/doc/gertwol/, 1998.

[Lud93] Petra Ludewig. *Inkrementelle wörterbuchbasierte Wortschatzerweiterung in sprachverarbeitenden Systemen*, Band 30 von *Dissertationen zur Künstlichen Intelligenz*. Infix, 1993.

[Meh93] Stephan Mehl. *Dynamische Semantische Netze*, Band 52 von *Dissertationen zur Künstlichen Intelligenz*. Infix, 1993.

[MM95] Petra Maier-Meyer. *Lexikon und automatische Lemmatisierung*. CIS-Bericht-95-84, Universität München, Centrum für Informations- und Sprachverarbeitung, 1995.

[Mon73] Richard Montague. The Proper Treatment of Quantification in Ordinary English. In J. Hintikka, J. Moravcsik und E. Suppes (Hrsg.), *Approaches to Natural Language*. Dordrecht, 1973.

[Mü98] Klaus Müller. Versuch einer Visualisierung polysemer Eigenschaften mit Hilfe selbstorganisierender Karten. Studienarbeit an Albert-Ludwigs-Universität Freiburg, Institut für Informatik und Gesellschaft, 1998.

[MWT80] W.D. Marslen-Wilson und L. Tyler. The Temporal Structure of Spoken Language Understanding. *Cognition*, 8:1–71, 1980.

[Nag62] Ernest Nagel (Hrsg.). *Logic, Methodology and Philosophy of Science*. Stanford University Press, 1962. International Congress 1960.

[Nir94] Sergei Nirenburg. Lexicon Acquisition for NLP: A Consumer Report. In Atkins und Zampolli [AZ94].

[Par84] Barbara H. Partee. Compositionality. In Fred Landman und Frank Veltman (Hrsg.), *Varieties of Formal Semantics*, Band 3 von *Studies in Semantics*, Seite 281–311. FORIS Publications, 1984.

[PB96] James Pustejovsky und Branimir Boguraev (Hrsg.). *Lexical Semantics - The Problem of Polysemy*. Clarendon Press, Oxford, 1996.

[Pin95] Manfred Pinkal. *Logic and Lexicon*, Band 56 von *Studies in Linguistics and Philosophy*. Kluwer Academic Publishers, 1995.

[Pus91] James Pustejovsky. The Generative Lexicon. *Computational Linguistics*, 17(4):409–441, 1991.

[Pus95] James Pustejovsky. *The Generative Lexicon.* MIT Press, 1995.

[Sch94] Anne Schiller. Deutsche Flexions- und Kompositionsmorphologie auf 2-Ebenen Basis. Technical report, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart, 1994.

[Sch95] Uka Maria Schmitt. Erweiterte Büchi-Automaten auf endlichen Wortmodellen: Theoretische Grundlagen, Automatengenerierung und Konzepte für das Parsen natürlicher Sprache. Technical Report 1/95, Universität Freiburg, Institut für Informatik und Gesellschaft (IIG), 1995.

[Sik97] Klaas Sikkel. *Parsing Schemata - A Framework for Specification and Analysis of Parsing Algorithms.* Texts in Theoretical Computer Science. Springer, 1997.

[SR90] Burkhard Schaeder und Burghard Rieger (Hrsg.). *Lexikon und Lexikographie*, Band 11 von *Sprache und Computer.* Georg Olms Verlag, 1990. Vorträge im Rahmen der Jahrestagung 1990 der GLDV.

[Ste83] M. Steedman. *Natural and unnatural Language Processing.* In [SW83], 1983.

[SW83] Jones K. Sparck und Y. Wilks. *Automatic Natural Language Processing.* Ellis Horwood, 1983.

[Tes59] Lucien Tesnière. *Eléments de syntaxe structurale.* Klincksieck, Paris, 1. Auflage, 1959.

[Tes80] Lucien Tesnière. *Grundzüge der strukturalen Syntax.* Klett-Cotta, 1. Auflage, 1980. Hrsg. und übers. von Ulrich Engel.

[Tho79] Richmond H. Thomason (Hrsg.). *Formal Philosophy: selected papers of Richard Montague.* Yale Univ. Press, New Haven, 3. Auflage, 1979.

[Tho90] Wolfgang Thomas. Automata on Infinite Objects. In Jan van Leeuwen (Hrsg.), *Handbook of Theoretical Computer Science*, Band B, Seite 133–191. Eslevier Science Publishers B.V., 1990.

[Tor58] W.S. Torgeson. *Theory and methods of scaling.* Wiley, 1958.

[TS96] Christine Thielen und Anne Schiller. *Ein kleines und erweitertes Tagset fürs Deutsche*, Seite 193–205. In Feldweg und Hinrichs [FH96], 1996.

[Wod70] W. A. Wodds. Transition Network Grammars for Natural Language Analysis. *Communications of the ACM*, 13, 1970.

[WZC95] Donald E. Walker, Antonio Zampolli und Nicoletta Calzolari (Hrsg.). *Automating the Lexicon: Research and Practice in a Multilingual Environment.* Oxford University Press, 1995.

[Zer91] Uri Zernik (Hrsg.). *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon.* Lawrence Erlbaum Associates, 1991. Seite 1-26.