Promis$D$ – Ein Analyseverfahren zur antizipationsfreien Erkennung und Erklärung von grammatischen Fehlern in Sprachlehrsystemen

D I S S E R T A T I O N

zur Erlangung des akademischen Grades
doctor philosophiae
(Dr. phil.)
im Fach Allgemeine Sprachwissenschaft

eingereicht an der
Philosophischen Fakultät II
Humboldt-Universität zu Berlin

von
Herrn Veit Reuer
geboren am 27.1.1969 in Leer/Ostfriesland

Präsident der Humboldt-Universität zu Berlin:
Prof. Dr. Jürgen Mlynek

Dekanin der Philosophischen Fakultät II:
Prof. Dr. Verena Lobsien

Gutachter:

1. Prof. Dr. em. Jürgen Kunze
2. Prof. Dr.-Ing. Wolfgang Menzel

Tag der mündlichen Prüfung: 26. Juni 2003

Abstract

The dissertation starts with an analysis of the requirements for Intelligent Computer-Assisted Language Learning systems (ICALL), which partially depend on didactic aspects of foreign language teaching. Based on this a type of exercise can be identified, that on the one hand allows the learner to enter free formed input supporting the so called communicative competence as a major didactic goal and on the other hand may be realised with advanced computational linguistics' methods. In the following chapter a look at grammar theories and especially Lexical Functional Grammar (LFG) is taken. The grammar theory needs to be tractable in an implementation and it is of a further advantage if the concepts of the theory are similar to the concepts in learner grammars in order to simplify the generation of feedback. Subsequently the user interface of the actual program is presented with a focus on error messages. The implementation is named Promis*D*, which stands for 'Projekt mediengestütztes interaktives Sprachenlernen - Deutsch'. Finally an anticipation-free parsing method is developed using neither information from the lexicon nor the grammar in order to identify grammar errors. The recognition is restricted to those areas where errors occur frequently in a learner corpus in order to allow for a greater efficiency parsing authentic data. Along the two structural levels in LFG the presentation of the algorithm follows: the constituent-structure with a modified Early-algorithm integrating error hypotheses into the chart and the feature-structure with a new unification-strategie storing information about clashing values in the f-structure. The dissertation closes with an evaluation and an outlook on the generation of error messages.

Keywords:
Parsing, Error Recognition, Intelligent Computer Assisted Language Learning, Lexical Functional Grammar

Zusammenfassung

Gegenstand der Arbeit ist zunächst eine Analyse der didaktischen Anforderungen an Sprachlernsysteme, die sich zum Teil aus dem Fremdsprachenunterricht ergeben. Daraus ergibt sich ein Übungstyp, der vom Lerner eine frei gestaltete Eingabe erfordert und damit insbesondere die kommunikative Kompetenz fördert, der aber auch mit Hilfe computerlinguistischer Methoden realisiert werden kann. Anschließend wird zur Auswahl einer geeigneten Grammatiktheorie insbesondere die Lexical Functional Grammar (LFG) näher betrachtet. Die Theorie muss sich aus computerlinguistischer Sicht für eine Implementierung im Rahmen eines Sprachlernprogramms eignen und es ist von zusätzlichem Vorteil, wenn die verwendeten Konzepte denen in Lernergrammatiken ähneln, um so die Generierung von Rückmeldungen zu vereinfachen. Im darauf folgenden Abschnitt wird kurz das eigentliche Programm Promis*D* (Projekt mediengestütztes interaktives Sprachenlernen - Deutsch) vorgestellt, wie es sich auch dem Nutzer präsentiert. Schließlich wird ein so genanntes antizipationsfreies Verfahren entwickelt, bei dem weder in der Grammatik noch im Lexikon Informationen zur Fehleridentifizierung enthalten sind. Die Fehlererkennung wird dabei auf die Bereiche eingeschränkt, in denen sich in einem Lernerkorpus häufig Fehler zeigen, um einerseits wesentliche Fehlertypen abzudecken und andererseits eine größere Effizienz bei der Analyse von realen Eingaben zu erreichen. Die Vorstellung des Verfahrens unterteilt sich entsprechend den grundlegenden Struktureinheiten der LFG in zwei Bereiche: die Konstituentenstruktur mit einer modifizierten Form des Earley-Algorithmus zur Integration von Fehlerhypothesen in die Chart und die Feature-Struktur mit einer veränderten Unifikationstrategie zur Behandlung und Speicherung von sich widersprechenden Werten in F-Strukturen. Zum Abschluss erfolgt die Evaluation und es werden die Möglichkeiten zur Gestaltung einer Rückmeldung an den Lerner diskutiert.

Schlagwörter:
Parsing, Fehlererkennung, Intelligentes Computergestütztes Sprachenlernen, Lexical Functional Grammar

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

In dieser Arbeit soll der Frage nachgegangen werden, welche Möglichkeiten es zur automatischen, das heißt maschinellen Analyse von Eingaben mit grammatischen oder auch morphosyntaktischen Fehlern in einem Sprachlernsystem[1] gibt. Besondere Beachtung finden dabei außerdem die Aufgabenstellung sowie die Form und der Inhalt der aus der Analyse erzeugten Rückmeldungen. Der folgende Satz soll das Problem verdeutlichen.

(1.1) *Ein Auto ist in den Ampel gestoßen.[2]

Ein Lehrer[3] würde bei diesem Satz in einer schriftlichen Arbeit sicherlich vermuten, dass der Lerner das korrekte Genus des Substantivs *Ampel* nicht kennt, da der Artikel *den* im Prinzip den korrekten Kasus Akkusativ anzeigt, der von der Präposition *in* in diesem Fall gefordert wird. Der Artikel *den* muss also zu *die* geändert werden. Desweiteren könnte ein Lehrer anmerken, dass das Verb *stoßen* im Kontext eines Unfalls üblicherweise mit der Präposition *gegen* verwendet wird, wobei der Bereich der Morphosyntax schon verlassen ist, da es sich hierbei eher um einen „Ausdrucksfehler“ handeln könnte.
In einer mündlichen Unterrichtssituation dagegen würde der Lehrer diese

[1] In dieser Arbeit werden die Begriffe *System*, *Programm* und *Anwendung* synonym verwendet. Das gleiche gilt für *Rückmeldung* und *Feedback*. Von Zeit zu Zeit werden auch englische Begriffe verwendet, wenn es sich hierbei um die gängige Terminologie handelt.

[2] Bei diesem Satz handelt es sich um eine authentische Eingabe, die eine Studierende aus einem Mittelstufenkurs „Deutsch als Fremdsprache“ des Sprachenzentrums der Humboldt-Universität zu Berlin bei der Benutzung des hier beschriebenen Programms gemacht hat. Der Stern „*“ markiert einen morphosyntaktisch zweifelhaften, wenn nicht gar falschen Satz. Gelegentlich wird auch das Zeichen „?“ verwendet, um einen meines Erachtens zweifelhaften Satz zu markieren, der aber möglicherweise in sehr speziellen Kontexten verwendet werden kann.

[3] Wenn es meiner Ansicht nach vertretbar ist, geschlechtlich unspezifizierte Formen oder beide Varianten zu benutzen, so werde ich dies tun. Ansonsten ist bei der Verwendung einer geschlechtlich spezifizierten Form die jeweils andere immer mitzudenken.

Fehler[4] vielleicht komplett ignorieren, wenn er das Gefühl hat, dass der Hörer die Intention des Sprechers verstanden haben und der Kommunikationsfluss weiterlaufen kann. Eine weitere plausible Reaktion bestünde in der Formulierung einer korrigierten Echofrage wie zum Beispiel „Ein Auto ist gegen die Ampel gestoßen?“, die nur einen indirekten Hinweis auf den Fehler beinhaltet, ansonsten aber das Ziel der kommunikativen Gesamtsituation weiterführt.

Dieses Beispiel demonstriert die vielfältigen Möglichkeiten zur Analyse einer fehlerhaften Äußerung, die auf unterschiedliche Art und Weise erklärt und korrigiert werden kann. Idealerweise sollte die Rückmeldung auf eine fehlerhafte Eingabe in einem Sprachlernsystem[5] genauso lerner- und situationsorientiert sein, wie sie im Unterricht möglich ist. Sprachdidaktiker, die Computer im Sprachunterricht einsetzen, bemängeln aber als eine der wesentlichen Unzulänglichkeiten in aktuellen kommerziellen CALL-Systemen genau diesen Aspekt der völlig unzureichenden Fehleranalysen und Rückmeldungen. Der Begriff „Fehleranalyse“ soll hier so verstanden werden, dass er einen nicht näher spezifizierten Oberbegriff zu Termini wie „Fehlerlokalisierung“, „Fehleridentifizierung“ und „Fehlererklärung“ etc. darstellt, der nicht nur den computerlinguistischen Teil umfasst, sondern darüber hinaus geht (siehe Abschnitt 2.3).

Gegenstand der vorliegenden Arbeit ist zunächst eine Analyse der Anforderungen an Sprachlernsysteme, die sich zum Teil aus der Didaktik des Fremdsprachenunterrichts (FU) ergeben und zum Teil durch die computerlinguistischen Möglichkeiten bestimmt werden. Darauf aufbauend wurde ein computerlinguistisches Parsingverfahren zur Erkennung von morphosyntaktischen Fehlern entwickelt, das eingebettet in ein Sprachlernprogramm die Eingaben von Sprachenlernern analysiert und bewertet. „Erkennung“ oder auch „Identifizierung“ von Fehlern soll sich in dieser Arbeit immer auf den computerlinguistischen Prozess beziehen, der als Ergebnis eine Struktur liefert, die Hinweise auf mögliche Fehlerpositionen und -typen sowie manchmal auch eine Korrektur beinhaltet. Drei Aspekte stehen dabei im Vordergrund.

Erstens wird ein sowohl im computerlinguistischen als auch im sprachwissenschaftlichen Bereich anerkannter Grammatikformalismus zur syntaktischen Analyse verwendet. Es handelt sich um die Lexical Functional Grammar (LFG; Kaplan und Bresnan, 1982). Wie gezeigt wird, hat die LFG zusätzlich zur computerlinguistischen Eignung den entscheidenden Vorteil,

[4]Wenn in dieser Arbeit der Begriff „Fehler“ verwendet wird, kann er in einigen Fällen auch die „fehlerhafte Äußerung“ mit umfassen, beispielsweise wenn von der „Korrektur eines Fehlers“ gesprochen wird, lässt sich das meines Erachtens nicht von der „Korrektur einer fehlerhaften Äußerung“ trennen.

[5]Für den Bereich des computergestützten Sprachenlernens hat sich die Abkürzung CALL (Computer-Assisted Language Learning) durchgesetzt. Folglich werde ich auch die Begriffe „CALL-System“ und die Abkürzung ICALL für „Intelligent Computer-Assisted Language Learning“ verwenden.

Konzepte zu verwenden, die denen so genannter Lernergrammatiken oder pädagogischer Grammatiken ähneln und die damit die Generierung von Rückmeldungen deutlich vereinfachen. Die Verwendung der LFG hat also in verschiedener Hinsicht Vorteile, die sich sowohl auf den Analyseprozess als auch auf die Feedbackmöglichkeiten auswirken.

Zweitens wurde ein Übungstyp entwickelt, der vom Lerner eine frei gestaltete Eingabe fordert und damit insbesondere die kommunikative Kompetenz als eines der herausragenden didaktischen Ziele fördern kann. Mit Hilfe der tiefen morphosyntaktischen Analyse und weiteren Modulen zur Abwicklung eines Dialogs ist das hier vorgestellte System in der Lage, einen Übungstyp zur Verfügung zu stellen, in dem kleine, aus dem Unterricht beziehungsweise aus Lehrbüchern bekannte Frage-Antwort-Dialoge simuliert werden. Beispielhaft sind drei Dialogübungen modelliert worden, in denen der Nutzer auf Fragen des Systems antworten muss: eine persönliche Vorstellung, ein Unfallbericht sowie eine Wegbeschreibung. Realisiert wurde die Implementation unter dem Namen Promis*D*, der für „*Pro*jekt *m*ediengestütztes *i*nteraktives *S*prachenlernen - *D*eutsch“ steht und in Anlehnung an das Projekt PROMISE (siehe Seite 5) gegeben wurde.[6]

Drittens schließlich arbeitet das von mir entwickelte Verfahren zur Fehlererkennung in dem Sinne antizipationsfrei, dass weder in der Grammatik noch im Lexikon Informationen zur Identifizierung von Fehlerpositionen und -typen enthalten sind. Allerdings wird die Fehlererkennung auf die strukturellen Bereiche und Features, an denen sich in einem Lernerkorpus häufig Fehler zeigen, eingeschränkt, um einerseits wesentliche Fehlertypen abzudecken und andererseits eine größere Effizienz bei der Analyse von authentischen Eingaben zu erreichen. Auf der Basis der Parse-Ergebnisse können insbesondere für den Bereich der Kongruenz- und Rektionsfehler präzise und umfassende Fehlermeldungen generiert werden.

1.1 Allgemeine Vorbemerkungen

Obwohl es seit Anfang der neunziger Jahre verschiedene internationale Workshops zum Thema ICALL gegeben hat, aus denen auch vielzitierte Publikationen entstanden sind (z.B. Swartz und Yazdani, 1992; Appelo und de Jong, 1994; Holland et al., 1995; Jager et al., 1998; Schulze et al., 1999) und auch intensiv an der Nutzung von computerlinguistischen Verfahren für den Bereich Sprachenlernen geforscht worden ist, sind tatsächlich sehr wenig kommerzielle Produkte aus diesen Entwicklungen entstanden. Die wenigen mir bekannten Programme, die käuflich zu erwerben gewesen sind, waren zum Beispiel WIZDOM (Handke, 1992) und „Herr Kommissar“ (deS-

[6] Dazu wurde SWI-Prolog und die grafische Erweiterung XPCE verwendet. Zum gegenwärtigen Zeitpunkt muss mindestens die SWI-Prolog-Version 5.0.5 und die XPCE-Version 6.0.5 verwendet werden.

medt, 1995). Als Antwort auf die Frage, warum relativ wenig Entwicklungen aus der Forschung zur Computerlinguistik und Künstlichen Intelligenz (CL&KI) in CALL-Programme Eingang gefunden haben, können meines Erachtens mindestens drei Gründe angeführt werden. Erstens ist die Entwicklung eines Systems, das computerlinguistische Methoden integriert, ein relativ aufwändiges Unterfangen, das bisher nur in wissenschaftlichen Projekten realisiert werden konnte. Diese Situation hat sich meines Erachtens in den letzten Jahren grundlegend geändert, da für bestimmte Bereiche Verfahren und Werkzeuge aus der Computerlinguistik zur Standardtechnologie gehören (cf. Nerbonne, 2002).

In Nerbonne et al. (1998, S. 2) wird Salaberry (1996) zitiert, bei dem noch ein zweiter Grund angeführt wird, der die Skepsis eines Sprachlehrers widerspiegelt. Da es die Linguistik und die Computerlinguistik bisher nicht geschafft haben, die ganze Komplexität der Sprache aufzubereiten und zur Verfügung zu stellen, müssen auch ICALL-Programme scheitern. Dieses Argument ist allerdings in dieser Form meines Erachtens nicht akzeptabel, da erstens in vielen linguistischen Bereichen umfassende Beschreibungen zur Verfügung stehen und zweitens auch in der Computerlinguistik wie erwähnt für bestimmte linguistische Bereiche wie beispielsweise die morphologische Analyse oder Tagging bereits umfassende Anwendungen entwickelt worden sind. Hier scheint sich die in Atwell (1999) hervorgehobene, überzogene Erwartungshaltung nach einer „Konversations-Übungsmaschine“ zu zeigen, die sowohl eine freie Konversation führen kann als auch nach didaktischen Gesichtspunkten auf fehlerhafte Äußerungen eingehen kann.

Salaberry (2000) hebt noch einen weiteren Punkt hervor: Obwohl viele Konzepte zur Nutzung neuer computerlinguistischer Verfahren entwickelt worden sind, haben sich daraus selten neue Bereicherungen für die Didaktik und Pädogogik des Sprachenlernens ergeben. Im Vordergrund stand in den meisten Fällen der Einsatz der Technologie und erst in zweiter Linie der Beitrag zum Sprachenlernen. Dieser Vorwurf kann auch der vorliegenden Arbeit gemacht werden: Eine zusätzliche Evaluierung des hier entwickelten Programms mit längerem Einsatz im Unterricht würde den Rahmen der Arbeit sprengen. Zusätzlich kann angemerkt werden, dass das Ziel der computerlinguistischen Forschung meines Erachtens vor allem in der Entwicklung neuer Verfahren und Anwendungen liegt und nicht notwendigerweise in der Entwicklung eines Systems zur vollen Marktreife und damit zur Anwendung im Unterricht. In dieser Arbeit wird also ein Programm entwickelt, das einerseits didaktische Anforderungen an CALL-Programmen zu berücksichtigen sucht und neue computerlinguistische Verfahren in ein ICALL-System integriert, andererseits aber ein Prototyp bleibt, für den nicht umfassend der Beitrag zur Sprachvermittlung untersucht wird.

Der Anstoß zur wissenschaftlichen Beschäftigung mit ICALL und zur Entwicklung des vorliegenden Programms hatte seinen Ursprung in einem Studien-Projekt mit dem Namen „Projekt mediengestütztes interaktives

Sprachenlernen Englisch“ (PROMISE) am ehemaligen Institut für semantische Informationsverarbeitung[7] der Universität Osnabrück (Bauer et al., 1994; John, 1994). In diesem Projekt entwickelte eine Gruppe von acht Studierenden, unter anderem auch mir, unter der Leitung und Mithilfe von drei Dozenten ein Programm zum Üben der kommunikativen Kompetenz in Englisch. Der in PROMISE präsentierte Übungstyp stellt das Vorbild für den hier gewählten Typ dar. Zur Analyse der Eingaben der Lerner in einem Frage-Antwort-Dialog wurde ein am Institut entwickelter Parser verwendet, der wie der hier vorgestellte auch eine Art LFG verarbeitete. Die Erkennung von Konstituentenfehlern[8] wurde mit Hilfe von Prolog-Code innerhalb von Fehlerregeln erreicht. Wenn ein solcher Fehler mit Hilfe der „falschen“ Phrasenstrukturregel (PS-Regel) erkannt wurde, wurde der enthaltene Prolog-Code ausgeführt und der Lerner über seinen Fehler informiert. Auf eine identische, streng antizipationsbasierte[9] Weise wurden Fehler in der Kongruenz etc. identifiziert. In einigen Tests mit Lernern ergab sich, dass das Programm nur auf eine sehr geringe Zahl von Fehlern reagieren konnte. Zusätzlich waren einige andere Module mit äußert wenig Umfang und Flexibilität implementiert, sodass die dialogorientierten Übungen nur sehr starr vom Lerner abgearbeitet werden konnten. Aus diesen Unzulänglichkeiten heraus ergab sich die Motivation, ein System zu entwickeln, das zwar einen ähnlichen Übungstyp realisiert, aber deutlich flexibler und umfassender auf die Eingaben eines Lerners reagieren kann.

1.2 Übersicht über die folgenden Kapitel

Zunächst wird in Kapitel 2 ein Blick auf die Didaktik des Computereinsatzes im Sprachunterricht geworfen, um daraus Anforderungen an Software herzuleiten, die mit Hilfe computerlinguistischer Methoden diesen Anforderungen zumindest zum Teil entgegenkommen kann. Im zweiten Teil wird die Behandlung von fehlerhaften Äußerungen im Unterricht beleuchtet, um daraus Eigenschaften einer geeigneten Behandlung von fehlerhaften Eingaben in dialogorientierten Übungen zu bestimmen.

Anschließend wird in Kapitel 3 eruiert, welche Beschreibungssysteme zur Präsentation von grammatischen Sachverhalten im Sprachunterricht gewählt werden. Mit Hilfe dieser Analyse soll gezeigt werden, dass die LFG eine

[7] Das Institut wurde 2002 in Institut für Kognitionswissenschaft umbenannt.

[8] Zu Konstituentenfehlern möchte ich nicht nur Wortstellungsfehler im engeren Sinne zählen, das heißt Sätze, bei denen die Reihenfolge der Wörter vertauscht ist, aber das gesamte lexikalische Material vorhanden ist. Zusätzlich möchte ich zu diesem Fehlertyp, den ich Verschiebungsfehler nenne, auch Auslassungs-, Einfügungs- und Ersetzungsfehler zählen. Nähere Erläuterungen folgen in Abschnitt 4.2.

[9] „Antizipationsbasiert“ soll hier das Gegenteil von „antizipationsfrei“ bedeuten, das heißt, Informationen zur Identifizierung von Fehlern sind in der Grammatik und/oder im Lexikon enthalten und damit vom Entwickler der linguistischen Datenbasis vorhergesehen worden.

geeignete Wahl der Grammatiktheorie darstellen kann, um für einen Lerner verständliches Feedback zu generieren.

Im darauf folgenden Kapitel 4 wird das eigentliche Programm so dargestellt, wie es sich auch dem Lerner des Deutschen beziehungsweise dem Benutzer präsentiert. Der Ablauf der Dialogübungen wird erläutert und es wird gezeigt, welche wesentlichen Systemreaktionen erfolgen können. Im zweiten Teil dieses Kapitels wird dann auf die Meldungen des Systems auf fehlerhafte Eingaben eingegangen. Zum einen wird dort erläutert, welche Fehler das System überhaupt in der Lage ist zu erkennen, und zum anderen wird demonstriert, wie sich die Erkennung der Fehler in Fehlermeldungen äußert. Der dritte und letzte Teil präsentiert die allgemeine Systemarchitektur. Anhand zweier Grafiken wird im Überblick gezeigt, welche Module welche Aufgaben übernehmen und welche Verarbeitungschritte zur Analyse einer Eingabe erfolgen.

Das entscheidende Kapitel 5 stellt nach einem Abschnitt über den „Stand der Technik“ im zweiten und dritten Abschnitt die Methoden zur Identifizierung von Fehlern in Lernereingaben vor. Die Vorstellung des Verfahrens unterteilt sich entsprechend den grundlegenden Struktureinheiten der LFG in zwei Bereiche: der so genannten K-Struktur (Konstituenten-Struktur) und der F-Struktur (Feature- oder auch funktionale Struktur). Im Rahmen eines Earley-basierten Chart-Parsers wird für die Analyse von Konstituentenfehlern eine Modifikation der so genannten Shift-Operation vorgenommen, sodass bei bestimmten Chart-Konfigurationen Fehlerhypothesen in Form von zusätzlichen, gewichteten Kanten zur Chart hinzugefügt werden, um darauf aufbauend möglicherweise eine Kante zu generieren, die den gesamten Satz überspannt. Für die Fehlererkennung in der F-Struktur werden die Definitionen für die Subsumption und darauf aufbauend die Unifikation derart verändert, dass ein Widerspruch zwischen atomaren Werten zu einem Feature in die F-Struktur integriert werden. Im Anschluss erfolgt die Evaluation mit Hilfe eines Lernerkorpus Deutsch als Fremdsprache und es werden die Möglichkeiten zur Gestaltung einer Rückmeldung an der Lerner diskutiert.

Im vorletzten Kapitel 6 wird kurz auf die Module eingegangen, die zwar für das Funktionieren des Systems von Bedeutung sind und auch zum Teil weitere Fehlermeldungen produzieren, die aber nicht den Schwerpunkt der Arbeit bildeten. Hierbei ist insbesondere das Modul zur Dialogsteuerung hervorzuheben, welches daher in einem separaten Abschnitt beschrieben wird. Der zweite und letzte Teil geht kurz auf weitere Module ein, die ich für erwähnenswert halte. Dabei handelt es sich insbesondere um die Module zur Orthografie- und zur Semantikkontrolle.

Kapitel 2

Computereinsatz und Fehlerbehandlung im Fremdsprachenunterricht

Dieses Kapitel beschäftigt sich in zweierlei Hinsicht mit der Didaktik des Fremdsprachenunterrichts (FU). Erstens soll die Didaktik in Hinblick auf die Verwendung von Computern im FU betrachtet werden. Zweitens stellt die didaktisch sinnvolle Behandlung von sprachlichen Fehlleistungen im FU einen weiteren Aspekt dar. Zusammengenommen werden dann die Konsequenzen für die Entwicklung eines Sprachlernsystems untersucht, das in der Lage ist, informative Rückmeldungen zu morphosyntaktisch fehlerhaften Eingaben der Lerner in dialogorientierten Übungen zu geben. Daraus ergibt sich einerseits die allgemeine Fragestellung, welche Vorteile sich aus dem Einsatz von computerlinguistischen Methoden in einem Sprachlernprogramm für den Lerner einer Fremdsprache ergeben. Andererseits werden so die speziellen Anforderungen für die Entwicklung des hier vorzustellenden Systems deutlich.

Mit der rasanten Entwicklung der Computertechnologie, der Telekommunikation und darin insbesondere des WWW („Neue Technologien“) hat sich auch die Nutzung von computergestützten Medien im Sprachunterricht stark ausgebreitet. Dabei geht es hier um den Einsatz von Computern im fremdsprachlichen Unterricht, obwohl auch einige Verlage Programme für den Lese- beziehungsweise Schriftspracherwerb der Muttersprache anbieten. Die folgende Liste nennt einige der allgemeinen Vorteile der Nutzung des Computers in einem Lernkontext.

- sofortiges Feedback
- unterschiedliche Formen der Interaktion
- potenzielle Unabhängigkeit von Zeit und Ort
- zugang zu unterschiedlichen Medienformaten mit einem Gerät, das heißt mit dem Computer

- netzwerkartige Strukturierung des Lehrmaterials für den individuellen und autonomen Einsatz

Nicht alle diese Aspekte treffen auf jedes Programm beziehungsweise didaktische Konzept zur Nutzung des Computers im FU zu; eher handelt es sich um Vorteile, die sich prinzipiell aus CALL ergeben können. Im folgenden Abschnitt liegt der Schwerpunkt auf den allgemeinen didaktischen Konzepten zur Integration des Computers im FU und im besonderen auf den Möglichkeiten der Kommunikation beziehungsweise Interaktion mit dem Computer. Von besonderer Relevanz ist die Ansicht, dass sich didaktische Konzepte wie zum Beispiel die Förderung der kommunikativen Kompetenz sowie die Erzeugung der Sprachbewusstheit mit Hilfe des Computers womöglich in besonderer Weise umsetzen lassen. Die Methoden der Computerlinguistik sind, wie sich zeigen wird, gerade hier sinnvoll integrierbar.

Für den Bereich CALL gibt es inzwischen eine fast unüberschaubare Menge an Publikationen mit didaktischen Ansätzen, Erfahrungsberichten und Bibliographien, zum Beispiel Chapelle (2001); Wolff (1998); Levy (1997); Higgins (1995). Ausgehend von so genannten konstruktivistischen Ansätzen (insbesondere Rüschoff und Wolff, 1999), die seit einiger Zeit im Mittelpunkt der Diskussion zum Einsatz von Computern im FU stehen, sollen in Abschnitt 2.1 die Kritikpunkte an diesen Ansätzen beleuchtet werden. Hier ist die Frage zu beantworten, welche neuen Aspekte die konstruktivistische Verwendung des Computers tatsächlich außer einer weiteren Motivation für das Fremdsprachenlernen gebracht hat. Beispielsweise zieht Götze (2004) den Vergleich mit den Sprachlaboren der 80er Jahre heran, die in ihrer Zeit ein ungeahntes Interesse am Medieneinsatz provoziert hätten. Trotz der vermeintlich revolutionären Technologie sei allerdings eine große Ernüchterung gefolgt, die nun womöglich auch dem Bereich CALL drohe. Außerdem soll in diesem Abschnitt eine Kategorisierung aufgestellt werden, die eine Einordnung des in Promis*D* verfolgten Ansatzes in das Spektrum der CALL-Systeme ermöglicht.

Im zweiten Abschnitt liegt der Fokus auf der Behandlung[1] von Fehlleistungen im Unterricht und den sich daraus ergebenden Hinweisen auf didaktische Konsequenzen für den Unterricht. Hier stehen die Möglichkeiten der Rückmeldung, welche insbesondere die Fehlerkorrektur umfasst, im Vordergrund. Abhängig von der jeweiligen Lehrsituation, die unter anderem von den Faktoren Mündlichkeit/Schriftlichkeit, Kommunikation/Vermittlung etc. abhängt, wird im Unterricht eine andere Behandlung der sprachlichen Fehlleistungen vorgenommen.

Im dritten Abschnitt geht es um die didaktischen Konsequenzen für ein Sprachlernprogramm, dessen Architektur computerlinguistische Verfahren zur Analyse von Lernereingaben beinhaltet. Idealerweise sollte dabei wie

[1] Unter dem Begriff „Fehlerbehandlung“, der im Folgenden öfter verwendet wird, soll sowohl die Behandlung der fehlerhaften Äußerung als auch des Fehlers selbst gemeint sein.

erwähnt nach neuesten didaktischen Konzepten die Förderung der kommunikativen Kompetenz und der Sprachbewusstheit an erster Stelle stehen. Es wird sich zeigen, dass insbesondere in dieser Hinsicht bisherige Übungsformate in Computerprogrammen in keiner Weise relevante Unterstützung geleistet haben. Dagegen kann der Übungstyp des simulierten Dialogs meines Erachtens die Forderungen der Didaktik nach kommunikativen Lernsituationen ergänzen.
Als weitere Konsequenz aus den Betrachtungen zur Fehlerbehandlung im Unterricht stellt sich heraus, dass in dem implementierten Übungstyp eine nach didaktischen Konzepten angemessene Fehlerbehandlung einerseits in Bezug auf den Kommunikationsfluss sowie der Fehlerbehandlung im mündlichen Bereich und andererseits in Bezug auf den Rückmeldungsinhalt der Fehlerbehandlung im schriftlichen Bereich entsprechen kann. Schließlich sollte ein Programm in Hinblick auf die inhaltliche Gestaltung der Rückmeldungen insbesondere auch die Möglichkeit der Anpassung an Lernerkenntnisse bieten.

2.1 Didaktik und Computer im Fremdsprachenunterricht

An dieser Stelle soll ein kurzer Überblick über die vorherrschenden didaktischen Tendenzen im Bereich CALL unter besonderer Berücksichtigung eines so genannten konstruktivistischen Ansatzes erfolgen. Charakterisieren lässt sich dieser Ansatz besonders in Abgrenzung gegenüber einem instruktionalen Ansatz: In Ersterem soll das Lernen primär durch die Konstruktion des sprachlichen Wissens und des Umgangs mit diesem Wissen durch den Lerner erfolgen, während in Zweiterem Lernen durch Instruktion im Vordergrund steht. Eine zeitgemäße Fremdsprachendidaktik in Bezug auf die Nutzung von Computern im FU sollte nach Ansicht von Rüschoff und Wolff (1999, S. 54ff) die drei folgenden „Formen des Sprachgebrauchs“ (oder auch Ziele des FU) umfassen. Ähnliches wird auch schon in Ritter (1995) hervorgehoben.

- Zum einen soll, wie auch in älteren Modellen immer wieder gefordert, die *kommunikative Kompetenz* als Ziel berücksichtigt werden. Diese Kompetenz, zur der ein großer Teil Sprachproduktion gehört, umfasst allgemein das Zurechtfinden eines Lerners in einer so genannten kommunikativen Situation.

- Zum zweiten soll *Sprachbewusstheit* (“‘Sprachbewusstsein”’ nach Bußmann, 1990, S. 697) gefördert werden. Der Lerner soll also nicht nur die Verwendung der Sprache in kommunikativen Situationen beherrschen, sondern er sollte sich bei der Verwendung auch der sprachlichen Strukturen seiner Äußerungen bewusst sein. Nach Bußmann geht es um die „Fähigkeit zu metasprachlichen Urteilen über sprachliche Ausdrücke“.

- Schließlich stellt die *Bewusstheit über das Lernen der Sprache* („Lernbewusstheit“ bei Rüschoff und Wolff) den dritten Aspekt dar. Der Lerner soll sich im FU auch über die Lernprozesse, die beim Lernen einer Fremdsprache ablaufen, bewusst werden, wobei sich möglicherweise zusätzlich der Aspekt des Erwerbs der Lernstrategien von der Lernbewusstheit abgrenzen lässt. Dieser Bereich soll aber in der vorliegenden Arbeit nicht weiter betrachtet werden.

Mit der Festlegung auf diese Bereiche hat sich eine gewisse Abkehr vom alleinigen Paradigma des Erreichens der kommunikativen Kompetenz ergeben. Sie steht nun nicht mehr als einziges Ziel im Mittelpunkt, sondern wird neben die Förderung der Sprachbewusstheit und der Bewusstheit über Lernprozesse gestellt. Ob sich aus dieser Reorientierung ein ganz und gar neues Paradigma in der Didaktik des Fremdsprachenlehrens und -lernens ergibt, ist umstritten. Außerdem muss hinzugefügt werden, dass es sich hier um Ideen aus der stark konstruktivistischen Lerntheorie nach Rüschoff und Wolff (1999) handelt. Diese Theorie wird aber von anderen Didaktikern mit der Begründung kritisiert, dass damit falsche Schwerpunkte für eine allgemeine Didaktik des Fremdsprachenlehrens gesetzt werden (Rösler und Tschirner, 2002). Es besteht vor allem die Gefahr, dass der Lerner in einem zu offenen Unterricht allein gelassen und der „didaktische Schutzraum“ ignoriert wird. Auf diesen Aspekt wird unten zurückzukommen sein. Umgekehrt muss hier allerdings betont werden, dass aus dem Bereich der eher traditionellen Didaktik zwar Anregungen für die Nutzung des Computers im FU kommen, diese sich dem Thema aber eher vorsichtig nähert und insofern die radikaleren Ideen prominenter erscheinen.

Alle drei Ziele erfordern nach Rüschoff und Wolff (1999, S. 58ff) verschiedene *Bausteine des modernen FU*. Zu diesen Bausteinen gehören dann unterschiedliche Gestaltungsparameter beziehungsweise -prinzipien:

- Inhaltsbezogenes/aufgabenorientiertes Lernen
  mit authentischen Materialien und Aufgabenstellungen
- Projektbasiertes/prozessorientiertes Lernen
  mit authentischen Interaktionsformen
- Kognitiv-konstruktivistisches Lernen
  mit eigenständigem/eigenverantwortlichem Wissenserwerb
- Lernen in einem offenen und multimodalen Umfeld

Alle diese Bausteine können in einem Unterricht, der von „Neuen Technologien“ unterstützt wird, nach Meinung der Autoren ausgezeichnet gefördert werden. Hier ist nicht der Platz, die einzelnen Bausteine an sich ausführlicher zu beleuchten und zu bewerten. Aber neben Rüschoff und Wolff misst beispielsweise auch Storch (1999, S. 12) diesen Bausteinen positives Gewicht

bei:„Entwicklungen wie [...] Lernerautonomie, Prozessorientierung oder die Reflexion über Lern- und Kommunikationsstrategien stellen keine Ablösung des kommunikativen [FUs] dar, sondern sinnvolle Erweiterungen und Fortentwicklungen.“ In dieser Arbeit soll dagegen die Verknüpfung dieser Elemente mit dem Einsatz von Computern im FU näher betrachtet werden.

2.1.1 Typologie

Da verschiedene Programmtypen für bestimmte Bereiche beziehungsweise Bausteine besonders geeignet sind, ist eine Kategorisierung von Software notwendig. In der Literatur hat es verschiedene Vorschläge zur Kategorisierung von Programmen gegeben, die in irgendeiner Form den Fremdsprachenerwerbsprozess unterstützen können. Wolff (1998) unterscheidet zum Beispiel die folgenden drei Oberkategorien, die in Rüschoff und Wolff (1999) im Wesentlichen beibehalten werden und die im Folgenden verfeinert werden sollen:

Tutoriell orientierte Anwendungen: geschlossene Systeme, die vor allem dazu dienen, Lexeme und grammatische Strukturen zu *üben* (Rüschoff und Wolff, 1999, S. 68). Die besondere Eigenschaft besteht darin, dass diese Systeme speziell für das Sprachenlernen entwickelt worden sind und der Lerner auf vorherbestimmten Pfaden durch das Programm geführt wird.

Werkzeuganwendungen: Werkzeuge, wie sie zum Teil aus dem beruflichen Alltag bekannt sind. Diese umfassen Textverarbeitung und Datenmanipulationsprogramme sowie aus dem sprachlernbezogenen Bereich Autorenwerkzeuge und „offene multimediale Anwendungen“, wobei letztere sich dadurch auszeichnen, dass eine bestimmte Arbeitsweise, das heißt Nutzerführung nicht durch das Programm festgelegt wird und sie daher „offen“ genannt werden.

Telekommunikation: E-Mail, Chat und WWW. Telekommunikation wird vor allem für die Beschaffung von multimedialen Materialien, zur Kommunikation mit Muttersprachlern und im Rahmen des „distance-learning“ eingesetzt.

Neu in Rüschoff und Wolff (1999) ist eine vierte Oberkategorie:

Anwendungen als Ressourcen: „Anwendungen, in welchen die Neuen Technologien als Ressourcen für das fremdsprachliche Lernen genutzt werden“ (Rüschoff und Wolff, 1999, S. 67). Zu diesem Bereich zählen elektronische Datenbanken, zum Beispiel Wörterbücher, aber auch Datenbanken mit Audio- und Videoarchiven und über das WWW erreichbare Textarchive.

Einige Jahre früher unterscheidet Ritter (1995) auch in drei Bereiche, allerdings mit einem etwas anderen Schwerpunkt, da das Internet zum damaligen Zeitpunkt noch nicht sehr weit verbreitet war: Tutorielle Programme, Spiele und Simulationen sowie Werkzeuge. Insbesondere den Spielen wurde 1995 noch eine wichtige Rolle zugebilligt, vermutlich, da viele Spiele noch textbasiert zu bearbeiten waren. Mit den zunehmenden Grafikfähigkeiten der Computer spielt Text bei Spielen und Simulationen derzeit nur noch eine sehr untergeordnete Rolle, was dazu führt, dass Spiele nicht mehr als eine für den FU relevante Kategorie wahrgenommen werden.[2]

Eine genauere Definition der genannten Kategorien in der Typologie wird nicht vorgenommen, sodass eine eindeutige Zuordnung einzelner, konkreter Programme manchmal schwierig ist und diese daher auch von den Autoren zum Teil in mehreren Kategorien aufgeführt werden. Diese nach funktionalen und didaktischen Gesichtspunkten gestalteten Kategorisierungen korrespondieren zum Teil mit anderen Aspekten des fremdsprachlichen Lernens (Rüschoff und Wolff, 1999, ebd.), „wie dem Erlernen fremdsprachlicher Strukturen und Lexeme [→tutorielle Systeme], der Arbeit mit authentischen Materialien [→Ressourcen]“ usw.

2.1.2 Typologie und didaktischen Gestaltungsprinzipien

Die Kategorisierung von Software und die oben angeführten Bausteine lassen in einer bestimmten Betrachtungsweise die Bewertung von Software hinsichtlich ihrer Eignung im FU zu. Aus der Bewertung ergibt sich, dass vor allem Programme geeignet sind, die entweder einen sehr offenen Charakter haben, das heißt die Informationen zusammengestellt nach unterschiedlichsten Kriterien zur Verfügung stellen können, oder die mit Hilfe des Internets die Bearbeitung authentischer Texte, das heißt Texte, die nicht für eine Lernsituation erstellt worden sind, sowie die „authentische“ Kommunikation mit Personen im Sinne einer Kommunikation mit Muttersprachlern (Email, Chat) erlauben.

Authentische Materialien/Texte können beispielsweise ohne größere Probleme aus dem WWW bezogen werden. Darauf aufbauend werden diese Texte dann zur Grundlage von *Projektarbeit* gemacht und mit Hilfe der Projektarbeit wird wiederum das *eigenständige/eigenverantwortliche Lernen* gefördert. Das *offene* und *multimodale Lernumfeld* ergibt sich aus dem Einsatz beziehungsweise der Kombination von Computern, Büchern und anderen Materialien, die den Lernern zur Projektarbeit zur Verfügung gestellt wer-

[2] Offensichtlich könnten Spiele auch bewusst so aufgebaut werden, dass die Benutzung von Sprache – in den meisten Fällen wohl Text – notwendig zur Bewältigung der Spiele wäre. Aber gerade die Analyse komplexer Eingaben benötigt computerlinguistische Methoden, für deren robusten Einsatz eine aufwendige Entwicklung notwendig ist und deshalb meines Wissens selten realisiert worden ist.

den. Als weiteres Beispiel wird auf der Website `http://www.ict4lt.org`[3] in der Einführung (Abschnitt 1.1.4.2 „ICT as a teaching and learning medium“) unter anderem darauf hingewiesen, welche Möglichkeiten dem Computernutzer in Bezug auf Lernen zur Verfügung stehen (meine Hervorhebungen):

- „create, edit and distribute multimedia documents *worldwide*;
- communicate with equal ease on a one-to-one or a one-to-many basis with contacts *anywhere in the world*;
- pursue their education by means of multimedia software, perhaps linked to a *website* and involving *on-line communications* with a tutor and fellow students.“

Hier zeigt sich verstärkt, als wie wichtig einerseits die Möglichkeiten der Kommunikationstechnologien und andererseits die darauf aufbauende „authentische“ Sprachverwendung, das heißt mit Muttersprachlern, gesehen wird.

Ritter (1995) hebt die Relevanz des handlungsorientierten Unterrichts hervor, der sich seiner Meinung nach gut in einen computergestützten Unterricht integrieren lässt. Mit einer etwas anderen Terminologie wird auch bei Ritter betont, dass es drei wesentliche Säulen im Konzept der modernen Fremdsprachendidaktik gibt, die mit Hilfe des Computereinsatzes verwirklicht werden können: Handlungsorientierung, Sprachbewusstheit sowie Lern- und Arbeitstechniken. Hier scheint im Vergleich kein wesentlicher Unterschied zu Rüschoff und Wolff zu liegen. Als Ziel des FUs gilt immer, „die Kommunikations- und Handlungsfähigkeit in den verschiedenen fremdsprachlichen Verwendungskontexten“ zu erreichen. „Dazu müssen eine Vielzahl schülerrelevanter Handlungssituationen geschaffen werden, die neben einer Orientierung an den Mitteilungsbedürfnissen der Lernenden vor allem auf einen interkulturell bedeutsamen und authentischen Sprachgebrauch abzielen.“ (Ritter, 1995, S. 255) Was nicht erwähnt wird, ist die selbständige Konstruktion von sprachlichen Wissen, die eng mit der so genannten Lernerautonomie verbunden ist. Im Gegensatz zu Rüschoff und Wolff lässt Ritter aber auch Programmen ihren Raum, die die oben erwähnten Ziele nicht direkt unterstützen können. Während Erstere bei der Kategorie „Traditionelle Lern- und Übungsprogramme“ (siehe Abschnitt 2.1.4; eine Unterkategorie der schon erwähnten „tutoriell orientierten Anwendungen“) die mangelhafte Eingabeanalyse sowie die „partikularistische und atomistische Ausrichtung“ bemängeln, äußert sich Ritter insbesondere nicht prinzipiell gegen „drill&practice“-Programme, die als Beispiele für die Bereiche „Übung und Anwendung eingegrenzter grammatischer Strukturen“ und „Wortschatzarbeit“ erwähnt werden.[4]

[3] Stand 2.2005 – ict4lt steht für „Information and Communication Technology For Language Teachers“

[4] Im Erscheinungsjahr 1995 hatte der Bereich der computergestützten Telekommunikation noch nicht in dem Umfang wie heute in den Alltag Einzug gehalten. Daher wurden

2.1.3 Bewertung des konstruktivistisch orientierten Konzepts

Nach dieser Vorstellung einiger grundlegenden Punkte soll nun auf einige Kritikpunkte am konstruktivistischen Konzept eingegangen werden. Aus dieser Kritik wird sich insbesondere ein geeignetes, didaktisches Anwendungsszenario für Promis*D* ergeben, das nach den bisher angeführten Aspekten eines didaktischen Konzepts eher ungeeignet erscheinen muss. Die wichtigsten Kritikpunkte am konstruktivistischen Ansatz bestehen unter anderem aus (cf. Rösler und Tschirner, 2002):

- Ablehnung eines strukturierten Unterrichts: Konstruktivistische Ansätze scheinen einen strukturierten Unterricht aus prinzipiellen Gründen abzulehnen, obwohl deutlich zwischen diesem und einer simplen Form des Instruktionismus unterschieden werden muss.
- Ablehnung einer „starken“ Lehrerrolle: Konstruktivistische Ansätze schließen einen starken Lehrer aus, obwohl gerade Aspekte wie zum Beispiel Feedback und Motivation durch den Lehrenden sich vorteilhaft auf Lernprozesse auswirken können.
- Vernachlässigung von unbewussten Spracherwerbsprozessen: Neben den wichtigen bewussten Spracherwerbsprozessen werden in konstruktivistischen Ansätzen die unbewussten Sequenzen des Spracherwerbs nicht berücksichtigt, obwohl diese für die Entwicklung von funktionalen Kompetenzen gleicher Maßen beteiligt sind.
- Unklarheiten bei der Beschreibung eines „autonomen Lerners“: Oft wird unsauber zwischen einem bereits autonomen Lerner und einem Lerner unterschieden, der zu einem autonomen Lerner werden soll, obwohl sich daraus relevante Unterschiede für die Planung des Unterrichts ergeben sollten.
- Fehlende Integration von Medienkompetenz: Neben der Nutzung des WWW zur Integration von authentischen Materialien, das heißt Materialien, die nicht für eine Lernsituation produziert worden sind, in den Lernprozess erscheint die damit einhergehende Notwendigkeit für eine starke Medienkompetenz in konstruktivistischen Ansätzen vernachlässigt zu werden, obwohl zum Beispiel die Beurteilung der Authentizität eines Dokuments im WWW von entscheidender Bedeutung ist.

Im Rahmen der Einordnung von Promis*D* sollen im Folgenden drei Aspekte herausgegriffen werden, um zu zeigen, dass eine tutoriell orientierte Anwendung auch in einem Szenario Sinn macht, das einerseits die „Bausteine des modernen FUs“ und andererseits aber die Kritik an diesem Konzept berücksichtigt und zu integrieren versucht.

---

im Wesentlichen nur Programme berücksichtigt, die „stand alone“ auf einem PC installiert werden konnten.

1. Notwendigkeit der Instruktion

Wenn die ersten beiden der oben angeführten Kritikpunkte zusammengefasst werden, kann man davon sprechen, dass eine wesentliche Voraussetzung eigentlich die Integration in eine Unterrichtssituation sein müsste. Die Verwendung von Ressourcen und Werkzeugen zur explorativen Erkundung einer Sprache setzt einen Lehrer voraus, der neben den angeführten Aufgaben wie Feedback und Motivation auch den Rahmen der Exploration festlegt und eine Aufgabenstellung vorgibt. Die angestrebte Erlangung der Sprachbewusstheit ist somit eng mit vermittelndem Unterricht verbunden. Es ist vermutlich davon auszugehen, dass nur, wenn ein Lehrer anleitet beziehungsweise erklärt, welche Formen in einer Sprache relevant und interessant sind, diese vom Lerner zur eigenständigen Konstruktion von zusätzlichem Wissen über die Sprache genutzt werden können.

Wenn aber die Instruktion in einer modernen Form durchaus ihren Platz in einem didaktischen Konzept hat, ist auch die Verurteilung eines tutoriellen Systems unangebracht. Schließlich kann ein tutorielles System in diesem Bereich Unterstützung leisten, wenn es nicht nur einfache Drill-Übungen anbietet, sondern zusätzlich eine „Verknüpfung [...] mit einer Art Lernenzyklopädie [bereitstellt], die Lernende zur eigenständigen Erarbeitung von Lösungen über die Recherche in entsprechenden Referenzmaterialien motiviert“, wie Rüschoff und Wolff (1999, S. 81) zugestehen. Diese Möglichkeit besteht nach Erkenntnissen der Autoren aber zur Zeit nur bei so genannten Autorenprogrammen, in denen ein solche Verknüpfung vom Autor/Lehrer integriert werden kann.

2. Vernachlässigung der kommunikativen Kompetenz

Obwohl die Kommunikation beziehungsweise die Erlangung der kommunikativen Kompetenz weiterhin als wesentliche Säule gesehen wird, beschränkt sie sich in den angestrebten Szenarien auf die mündliche Kommunikation über die Sprache mit dem Computer als Werkzeug. In erster Linie wird Sprache, die mit Hilfe von Werkzeugen in projektorientierter Arbeit untersucht wird, rezipiert, während sich die Produktion (zur Zeit) schwerpunktmäßig auf geschriebene Sprache beschränkt (siehe die erwähnte Website `www.ict4lt.org`: „create, edit and distribute multimedia documents worldwide“).

Wenn der Computer als Instrument zur Telekommunikation eingesetzt wird, entsteht zwar eine gewünschte Form der authentischen Kommunikation. Diese beschränkt sich aber wiederum (zur Zeit) auf die schriftliche Form, nämlich den E-Mail- beziehungsweise Chat-Kontakt, der zusätzlich sehr speziellen Beschränkungen und Konventionen unterliegt. Notwendig sind also zusätzlich Elemente, die direkte Interaktionformen wie zum Beispiel Dialoge trainieren. Auch in einem simulierten Dialog wie in Promis*D* finden

sich bestimmte Aspekte der mündlichen Kommunikation, die für den Erwerb der kommunikativen Kompetenz von Bedeutung sein können, wie in Abschnitt 2.2 erläutert wird.

3. Fehlendes Feedback in der Hypothesenbildung

Eine entscheidende Funktion muss in all diesen Szenarien vom Lehrer übernommen werden: er muss immer noch das Feedback zur Qualität des von den Lernern produzierten Materials übernehmen. Weder liefert der alleinige Umgang mit authentischen Materialien Feedback zu sprachlichen Strukturen, noch wird bei der Produktion von sprachlichem Material Feedback durch den Computer geliefert. Diese Tatsache bedeutet auch, dass die Prinzipien des konstruktivistischen Ansatzes grundsätzlich nur sehr beschränkt für einen Selbstlerner geeignet sind. Hierfür bedarf es einer Anleitung zur Erkennung der relevanten sprachlichen Strukturen und es bedarf eines Feedbacks auf vom Lerner produziertes Material.

Allgemeiner gesehen entsteht der Eindruck, dass die Verbreitung des Computers und die Entwicklung neuer didaktischer Konzepte miteinander einhergehen (siehe zum Beispiel Levy, 1997, S. 221ff). Der Computer bietet erweiterte Möglichkeiten, mit sprachlichem Material umzugehen, die dann für die kognitive Konstruktion von Sprachwissen eingesetzt werden können. So lassen sich bestimmte Arbeitsschritte, wie zum Beispiel die Arbeit mit Konkordanzen, erst durch den Einsatz eines Computers effektiv und für den Unterricht geeignet realisieren. Diese Möglichkeiten bedeuten aber nicht, dass ähnliche didaktische Konzepte wie beispielsweise die Kommunikation mit Muttersprachlern oder der Einsatz von authentischen Materialien nicht auch ohne den Einsatz von Computern realisiert werden könnten. Auf diesen Aspekt wird auch in der Kritik von Rösler und Tschirner (2002) hingewiesen, in der die positiven Elemete des konstruktiven Ansatzes eher als eine Anregung an die traditionelle Didaktik gesehen werden, bestimmte Methoden der Fremdsprachendidaktik stärker in der Vordergrund zu holen, um damit eine Korrektur der Schwerpunkte zu bewirken. Beispielsweise ist nicht erst seit der Entwicklung konstruktivistischer Konzepte für den FU bekannt, "dass Sprachgebrauch und fremdsprachliches Lernen eine emotionale und leibliche Komponente haben und in der Lebenswelt situiert sind [...]"(Rösler und Tschirner, 2002, S. 146)

Eine konkretere Fragestellung, die hier aber nicht beantwortet werden kann, ist, ob die Entwicklung im Bereich CALL eher technologiegetrieben ist, oder ob die Innovation wie in erwünschter Weise aus dem Bereich des Sprachenlehrens beziehungsweise -lernens stammt (Levy, 1997, S. 215f). Eine sehr weitreichende Schlussfolgerung wäre es, die Technologie als den entscheidenden Faktor einzustufen und den Computer ausschließlich als Mittel zur verstärkten Motivation zu sehen.

Nach diesen allgemeinen Anmerkungen zum generellen Konzept des Ein-

satzes von Computern im FU unter besonderer Berücksichtigung des konstruktivistischen Ansatzes von Rüschoff und Wolff soll nun auf die spezielle Kategorie der „tutoriellen Anwendungen“ eingegangen werden, da das System Promis*D* dieser Kategorie zuzuordnen ist.

2.1.4 Präzisierung der Typologie zu tutoriellen Anwendungen

Die tutoriellen Systeme (Abschnitt 2.1.1) lassen sich weiter in „traditionelle Lern- und Übungsprogramme“, „Lernprogramme auf der Basis der KI-Forschung“ und „tutoriell orientierte, geschlossene Multimedia-Anwendungen“ unterteilen (Rüschoff und Wolff, 1999, S. 68ff).

> **Traditionelle Lern- und Übungsprogramme:** Obwohl keine eigentliche Definition gegeben wird, handelt es sich offensichtlich um Übungsprogramme, die ohne multimediale Unterstützung lediglich Übungen wie Multiple Choice und Lückentextaufgaben anbieten. Wesentliche Vorteile sind die sofortige Rückmeldung, die Möglichkeit zur zeitlich individuellen Arbeitsgestaltung und manchmal die Anpassung an den Unterricht mit Hilfe von Autorensystemen.
>
> **Lernprogramme auf der Basis der KI-Forschung (Künstliche Intelligenz):** Hierbei handelt es sich nach Meinung der Autoren um Systeme, die mit einer intelligenten Fehleranalyse und Fehlerkorrekturfunktion ausgestattet sind, sodass im Folgenden insbesondere die Methoden der Computerlinguistik (CL) mit einbezogen werden, da es sich bei der Fehleranalyse meines Erachtens eher um einen Bereich der CL und nicht um einen der KI handelt. Auf diese Kategorie wird in Abschnitt 2.1.6 speziell eingegangen.
>
> **Tutoriell orientierte, geschlossene Multimedia-Anwendungen:** Diese Systeme zeichnen sich dadurch aus, dass sie zwar alle Arten von multimedialen Elementen integrieren, vor allem aber den Lerner „von Anfang bis Ende an die Hand [nehmen]“. Üblicherweise werden Texte oder Dialoge zum Teil auch visuell und akustisch präsentiert, zu denen dann der Lerner Aufgaben bearbeiten muss.

Bevor später KI-basierte Systeme betrachtet werden (S. 19), soll zunächst auf die Kategorie der Multimedia-Anwendungen eingegangen werden, da traditionelle Übungsprogramme nicht mehr dem aktuellen Stand der Technik und der Didaktik entsprechen und wohl kaum noch im Einsatz sind.

2.1.5 Tutoriell orientierte Multimedia-Anwendungen

Neben der multimedialen Präsentation und den schon erwähnten Vorteilen tutorieller Anwendungen wie beispielsweise dem sofortigen Feedback auf Ler-

nereingaben besitzen diese Programme eine weitere wesentliche Eigenschaft, die der strukturierten Einführung von Lexemen und grammatischen Phänomenen. Damit wird es einem Lerner ermöglicht, einen bestimmten Bereich einer Sprache durch das Programm geleitet kennenzulernen. Die Einbettung des zu lernenden Materials in Dialoge und Situationen liefert außerdem einen Kontext, der unter Umständen die Memorisierung erleichtert.

Tutoriell orientierte Programme, die das zu vermittelnde Wissen vorgeben und auch die Art der Vermittlung stark festlegen, lassen sich allerdings nur begrenzt im konstruktivistisch orientierten Unterricht einsetzen. Kaum eines der geforderten Merkmale wie zum Beispiel „der eigenständige/ eigenverantwortliche Wissenserwerb“ mit Hilfe von kognitiv-konstruktivistischem Lernen lassen sich damit realisieren, da diese Programme den Umgang mit sprachlichem Material stark einschränken beziehungsweise vorschreiben. „Geschlossene Multimedia-Anwendungen [...] führen wahrscheinlich nur in geringem Maße dazu, dass die Lernziele Kommunikationsfähigkeit, Sprachbewusstheit und Lernbewusstheit realisiert werden.“ (Rüschoff und Wolff, 1999, S. 71) Anzumerken ist hier allerdings, dass gerade eine strukturierte Einführung die Bewusstmachung der wesentlichen Eigenschaften einer Sprache vermutlich fördern kann. Anstelle einer Aufgabenstellung zur Identifizierung bestimmter Eigenschaften beispielsweise in einer *umfangreichen* Materialsammlung, können tutorielle Anwendungen eine „angeleitete“ Einführung in diese Eingeschaften mit *wenigen, deutlichen* Beispielen bieten und damit auch die Sprachbewusstheit fördern. Auch die umfangreiche Diskussion zum Stellenwert des Grammatikunterrichts deutet darauf hin, dass eine Bewusstmachung sprachlicher Strukturen zu einem eher unbewussten Wissen führt, das dann in kommunikativen Situationen eingesetzt werden kann. Die Grammatikvermittlung bekommt so die Aufgabe, „den Trial-and-error-Prozess des Bildens, Überprüfens und Revidierens von Hypothesen über die zu erlernende Sprache zu unterstützen.“(Storch, 1999, S.75)

Obwohl die tutoriell orientierten Anwendungen von den Produzenten beziehungsweise Verlagen oft als innovativ gepriesen werden, sind „viele dieser Materialien eigentlich nur alte Zöpfe mit neuen und etwas bunteren Schleifen [...]. Genau wie bei CALL-Materialien der ersten Stunde verbergen sich hinter den bunten und tönenden Bildschirmen zumeist nur einfache Drill-Programme und traditionelle Übungsformate“, bemängeln Rüschoff und Wolff (1999, S. 82). Obwohl also das zu lernende Material mit Hilfe von Multimedia für alle Wahrnehmungskanäle aufbereitet wird, liegt ein wesentliches Problem in den Übungsformaten.

Damit solch ein Programm überhaupt in der Lage ist, die Eingaben der Lerner zu bewerten, bieten sich zwei Strategien an. Entweder werden die Möglichkeiten zur Lösung der Aufgabe in irgendeiner Form vorgegeben (Auswahlliste oder Multiple-Choice), oder der Kontext beziehungsweise die Aufgabenstellung wird so eng gefasst, dass nur sehr wenige (Fehl-) Eingaben möglich sind. In beiden Fällen wird die Menge der Möglichkeiten allerdings

so weit eingeschränkt, dass vom Lerner praktisch keine Reflexion über die Eingabe notwendig ist. Der alte Vergleich, der auch bei den frühen Programmen ohne Multimedia („traditionelle Lern- und Übungsprogramme“) herangezogen wurde, gilt also immer noch: Die Übungen entsprechen einer nahezu exakten Umsetzung der Übungen aus dem so genannten Arbeitsbuch, mit Ausnahme der Tatsache, dass der Lerner bei der Benutzung des Computers eine sofortige Rückmeldung erhält und nicht auf die Beurteilung des Lehrers warten oder in einem Lösungsheft nachschlagen muss. Auch wenn tutorielle Anwendungen unter Umständen Sprachbewusstheit fördern können, bleibt die erstrebenswerte Förderung der kommunikativen Kompetenz natürlich aus, da vom Lerner keinerlei kreativer Sprachgebrauch gefordert wird.

Die Behauptung verschiedener Lernsoftware-Hersteller, dass ihre tutoriellen Programme *inter*aktiv seien, muss aus den genannten Gründen abgelehnt werden. Vermutlich trifft eher die Bezeichnung *reaktiv* zu, da das Lernen mit diesen Programmen überwiegend aus Rezeption von Sprache besteht, was aber nur die eine Hälfte der Kommunikation ausmacht. Die andere Hälfte, nämlich die Sprachproduktion in kommunikativen Situationen, kann wie erwähnt nicht mit (traditionellen) tutoriell orientierten Systemen geübt werden.[5]

2.1.6 KI-basierte Systeme

Rüschoff und Wolff (1999, S. 69f) setzen sich in einem Unterkapitel mit „Lernprogrammen auf der Basis der KI-Forschung“ auseinander. Aus zwei Gründen haben sich speziell Programme, die Methoden der CL&KI verwenden, nach Meinung der Autoren bisher nicht durchsetzen können.

- Erstens gelang es ihrer Ansicht nach nicht, neue „Übungsformate zu entwickeln, die über primitive *multiple-choice* und Lückentextübungen hinausgehen“ (Rüschoff und Wolff, 1999, S. 70).
- Zweitens sind die Programme immer noch langsam und ungenau, sodass ein tatsächlicher Einsatz im FU nicht sinnvoll erscheint.
- Ein dritter Grund wird von den Autoren gegen „Lernprogramme auf der Basis der KI-Forschung“ angeführt, der sich aber genau genommen allgemein gegen tutoriell orientierte Anwendungen richtet: Die Bausteine des modernen FUs wie inhaltsbezogenes/aufgabenorientiertes Lernen, projektbasiertes/prozessorientiertes Lernen und kognitiv-konstruktivistisches Lernen werden in einem Konzept mit tutoriell orientierten Programmen nicht unterstützt (siehe Seite 10).

[5] Brian Farrington hat die tutoriellen Systeme auf der EuroCALL 2000 als „stone-deaf tutors“ bezeichnet, das heißt, die Programme werden zwar als „Tutoren“ bezeichnet, verhalten sich aber, als wären sie stock-taub. „Consequently, a session with one of the currently fashionable multimedia packages is like trying to learn to speak a foreign language from a tutor who is stone deaf ...“

Die Autoren lassen mit dieser Kritik zwei entscheidende Aspekte außer Acht, wobei sich die Situation in den letzten Jahren nicht zuletzt auch aufgrund der gestiegenen Rechenleistung grundlegend geändert hat. Erstens wurde nicht nur versucht, mit Hilfe der CL-Technologie klassische Übungsformate, also tutoriell orientierte Systeme zu verbessern, sondern es wurden auch neue Ideen entwickelt, die sehr wohl kognitiv-konstruktivistisches Lernen unterstützen (zum Beispiel Zock (1992) und Hamburger (1994, 1995), aber auch zur Orthografie Thelen (2002)). Als Beispiele sollen hier auch die Systeme Glosser–RuG (Dokter und Nerbonne, 1998) und LogoTax (Ludewig, 2003) angeführt werden, die beide nicht Module zur Fehleranalyse von Lernereingaben beinhalten, sondern auf unterschiedliche Weise die Exploration von Sprachdaten unterstützen und damit konstruktivistisch-didaktische Konzepte umsetzen. Zweitens existieren Systeme, die Methoden der Computerlinguistik integrieren und im FU auch eingesetzt werden. Ein Beispiel ist wiederum das System Glosser–RuG (Dokter et al., 1998), bei dem zum Beispiel ausführliche Nutzerstudien durchgeführt wurden. Auch das von Heift (2003) beschriebene System, das zwar „klassische“ Übungformate enthält, wird im Sprachunterricht eingesetzt und ist ausführlich evaluiert worden.[6]

Damit sind alle drei Kritikpunkte an Programmen auf der Basis von Methoden der CL&KI entkräftet. Um in diesem Bereich ein präziseres Bild von den unterschiedlichen Ausformungen der Programme zu erhalten, kann eine weitere Unterscheidung von Systemen mit CL-Methoden (Lernprogramme auf der Basis der KI-Forschung) nach funktionalen Gesichtspunkten getroffen werden, die den Rahmen der Strukturierung nach Rüschoff und Wolff (1999) aber verlassen muss.

> **Toolartige, CL-basierte Systeme:** Zu dieser Kategorie gehören Anwendungen, die auf (computer-) linguistischer Grundlage authentische Sprachdaten analysieren und die Ergebnisse in meist modifizierter Form dem Lerner präsentieren, um damit unter anderem die erwähnten konstruktivistisch-didaktischen Konzepte umzusetzen.[7] Eine weitere Unterteilung dieser Kategorie kann insofern vorgenommen werden, als dass bestimmte Systeme eher dynamisch angelegt sind und (nahezu) beliebige Texte analysieren, während andere System in dieser Hinsicht eher statisch aufgebaut sind und nur die Untersuchung der bereits enthaltenen Texte erlauben.
>
> **Tutorielle, CL-basierte Systeme:** Diese Systeme leisten in erster Linie eine Fehleranalyse der Lernereingabe. Dabei kann zusätzlich zwi-

[6] Auch in Chapelle (2001) wird lediglich Literatur aufgeführt, die auf Forschungsarbeiten von vor 1995 beruht und damit entscheidende Entwicklungen nicht berücksichtigt beziehungsweise misachtet.

[7] Zu beachten ist, dass diese Kategorie streng genommen keine Unterkategorie der KI-basierten Lernprogramme im Sinne von Rüschoff und Wolff (1999) ist, da es sich nicht um tutorielle Anwendungen handelt.

schen Programmen zur Analyse von geschriebener und von gesprochener Sprache unterschieden werden. Programme zur Analyse von schriftlichen Lernereingaben lassen sich wiederum nach der Aufgabenstellung klassifizieren:

CL-basierte Systeme zur Förderung von grammatischen Strukturen: In diese Kategorie lassen sich Systeme einordnen, die traditionelle Satzkonstruktionsaufgaben beinhalten, zum Beispiel auf der Grundlage einer Liste mit unflektierten Wörtern.

CL-basierte Systeme zur Förderung der kommunikativen Kompetenz: Zu diesen Systemen gehören Programme, die zwar eine Analyse der Lernereingaben vornehmen, diese aber in kommunikativen Übungen verwenden, um die Reaktion des Systems als Interaktionspartner zu planen.

Jede dieser Kategorien lässt sich außerdem in Hinsicht auf eine möglicherweise enthaltene Lerner-Modellierung unterscheiden, womit meines Erachtens erst die Methoden der KI berücksichtigt werden. Dieser Bereich des Usermodelings sei hier aber nur der Vollständigkeit halber angemerkt und wird im Folgenden nicht weiter berücksichtigt. Als Beispiel sei das System ELDIT (Gamper und Knapp, 2001) angeführt, das im Wesentlichen aus einem adaptiven Online-Lexikon besteht, in das Vokabelübungen integriert sind.

Hervorgehoben werden muss außerdem, dass tutorielle, CL-basierte Systeme entwickelt worden sind, um insbesondere das Feedback zu verbessern, womit ein weiterer, entscheidender Kritikpunkt an tutoriell orientierten Anwendungen entkräftet wird. Beispielsweise ist das System E-Tutor (Heift, 2003) in der Lage, für eine Reihe von insbesondere morphosyntaktischen Fehlern präzise Rückmeldungen zu geben, die sich zusätzlich an das Vorwissen des Lerners anpassen lassen.

Wesentlich in Bezug auf ein Konzept wie Promis*D* ist hier, dass es auf der Basis einer umfassenden morphosyntaktischen Analyse dem Lerner einerseits ermöglicht wird, Übungen mit einer erweiterten Aufgabenstellung zu bearbeiten und andererseits Feedback zu Orthografie und Morphosyntax auf Grundlage von CL-Methoden zu erhalten. Die erweiterte Aufgabenstellung in einer kommunikativen Übung wie beispielsweise in einem simulierten Frage-Antwort-Dialog sollte vom Lerner insbesondere die Eingabe von Sätzen erfordern, die lediglich durch die Darstellung eines Situationsrahmens vorstrukturiert sind. Damit wird der Lerner in die Lage versetzt, ununterstützt Sprache produzieren zu müssen und trotzdem relativ umfassende Rückmeldungen erhalten zu können. Neben der Möglichkeit zur erweiterten Sprachproduktion, die die kommunikative Kompetenz verbessern kann, wird offensichtlich insbesondere durch die Möglichkeit der Rückmeldungen die Sprachbewusstheit gefördert.

Nach dieser Kategorisierung muss das hier vorzustellende System Promis*D* als tutorielles, CL-basiertes Programm zur Förderung der kommunikativen Kompetenz charakterisiert werden. Wie sich zeigen wird, stehen Lexeme und grammatische Strukturen zwar nicht unmittelbar im Vordergrund der Übungen, dennoch besteht ein Schwerpunkt in den Rückmeldungen zu morphosyntaktischen Fehlern. Ein zweiter Gesichtspunkt, der zu dieser Einteilung führt, ist die Geschlossenheit des Systems. Das Design des Programms verhindert eine Erweiterung der Inhalte oder eine Veränderung der Übungstypen durch Lehrer oder gar Lerner. In Abschnitt 2.3 werden insbesondere die zu Beginn angeführten „Bausteine des modernen FUs“ erneut aufgegriffen, um dann zu zeigen, in welcher Form das Konzept von Promis*D* die Realisierung einige dieser Bausteine unterstützen kann.

2.1.7 Zusammenfassung

Abschließend lässt sich festhalten, dass der Computer im Sprachunterricht eine wesentliche Bereicherung darstellen kann, wenn das Lernen mit bestimmten didaktischen Konzepten verknüpft wird. Insbesondere kann das Internet mit Hilfe des WWW und seinen Kommunikationsmöglichkeiten dazu beitragen, mit authentischen Texten und authentischer Kommunikation projektorientierte Lernarbeit zu unterstützen. Außerdem erweitern Datenbanken wie zum Beipiel elektronische Enzyklopädien die Möglichkeiten des eigenverantwortlichen, konstruktiven Lernens. Allerdings werden mit Übungstypen, die diese Ressourcen nutzen, üblicherweise nur bestimmte Formen der Kommunikation trainiert und der Umgang mit Sprache auf begrenzte Bereiche eingeschränkt. Darüber hinaus scheinen didaktische Konzepte, die auf dem konstruktivistischen Ansatz beruhen, den Nutzen eines stärker strukturierten Unterrichts und die Funktion des Lehrenden für Anregung und Feedback zu ignorieren, sodass der Eindruck entsteht, der Lerner wird mit der Aufgabenstellung und umfangreichen Materialien mehr oder weniger allein gelassen.

Tutoriell orientierte Anwendungen ohne CL-Methoden sind zwar üblicherweise in der Lage, lexikalische und strukturelle Eigenschaften einer Sprache zu vermitteln beziehungsweise zu trainieren, lassen aber weder projektorientiertes noch konstruktives Arbeiten mit der Sprache zu, obwohl diese Übungsformen als essentielle Bestandteile des Sprachenlernens gesehen werden. Schwerwiegender noch wiegt ein weiterer Nachteil bei der Benutzung von herkömmlichen, tutoriell orientierten Anwendungen in Selbstlernsituationen: Die Komponente der eigenständigen Sprachproduktion in kommunikativen Situationen wird nicht annähernd berücksichtigt. Dagegen können Anwendungen, die Methoden der CL&KI integrieren, sowohl den Umgang mit authentischen Sprachdaten verbessern als auch erweiterte Übungsformate beinhalten. Auf diese Weise können einerseits bestimmte Elemente einer konstruktivistischen Didaktik mit linguistischen Informationen zur Verbes-

serung der Sprachbewusstheit angereichert werden und andererseits mehrere Kritikpunkte an tutoriellen Systemen wie zum Beispiel das Fehlen „echter Interaktion“ überwunden werden.

Nach dieser Bewertung insbesondere der tutoriell orientierten Systeme geht es im folgenden Abschnitt um die Behandlung von fehlerhaften Äußerungen der Lerner im Unterricht. In Abschnitt 2.3 folgt dann eine Aufstellung und Diskussion von Anforderungen an ein ICALL-Programm, das den Leitlinien der Sprachlerndidaktik in bestimmten Szenarien genügen kann.

2.2 Didaktik und Fehlerbehandlung im Fremdsprachenunterricht

Wenn einer der wesentlichen Aspekte von Promis*D* die Analyse von und Rückmeldungen zu fehlerhaften Eingaben ist, muss auch eine Betrachtung der didaktisch sinnvollen Behandlung von fehlerhaften Äußerungen und den enthaltenen Fehlern im FU erfolgen, um auf diese Weise zumindest ein Idealkonzept für ein ICALL-Programm entwickeln zu können. Der implementierte Übungstyp ähnelt insofern einer mündlichen (Unterrichts-) Situation, als dass auch hier die Bewältigung des Dialogs und damit der kommunikativen Situation, und nicht die Produktion von korrekter Sprache im Vordergrund steht. Neben der „reinen“ Aufgabenstellung der Dialogsimulation spielt hier vor allem die Aufgabe der (nahezu) freien Sprachproduktion die entscheidende Rolle, die den Lerner zwingt, ganze Sätze ohne eine wesentliche Hilfe zu produzieren. Aus noch ausführlicher zu erläuternden Gründen liegt daher der Schwerpunkt im Folgenden auf dem Umgang mit fehlerhaften Äußerungen in mündlichen Unterrichtssituationen. Schließlich ist der Fehler ein wesentlicher Bestandteil der Sprachstandsanalyse und kann außerdem dem Lerner helfen, die speziellen, persönlichen Problembereiche bei der Beherrschung einer Sprache zu erkennen.

2.2.1 Exkurs Spracherwerbsforschung

Zu Beginn dieses Abschnitts folgt ein kurzer Exkurs zu sprachlichen Fehlleistungen von Fremdsprachenlernern aus der Sicht der Psycholinguistik beziehungsweise Spracherwerbsforschung, da dieser Bereich auch hier immer wieder Thema in Publikationen ist. Die Forschung zum Zweitspracherwerb beschäftigte sich insbesondere in den 70er und 80er Jahren mit der so genannten „Error Analysis“ (EA), die die „Contrastive Analysis“ – der Vergleich von Mutter- (L1) und Lerner- beziehungsweise Zielsprache (L2) – ablöste. Drei Ziele standen dabei im Vordergrund: Erstens sollte die EA Einblicke in Zweitspracherwerbsprozesse ermöglichen, zweitens sollten die Ergebnisse zur Verbesserung der Lehrmethoden dienen und drittens wurde auch die Möglichkeit gesehen, Sprachenlernern mit Hilfe der EA ein gewis-

ses Maß an Sprachbewusstheit zu vermitteln (cf. Dulay et al. (1982), Kapitel 7; Ellis (1994), Kapitel 2). Verschiedene „common sense“-Hypothesen wurden im Zuge dieser Untersuchungen widerlegt. Zum Beispiel wird gerne angenommen, dass Sprachenlerner das lernen, was ihnen im Unterricht vermittelt wird. Es hat sich aber im Gegenteil herausgestellt, dass es vermutlich eine bestimmte Reihenfolge des Lernens der verschiedenen Aspekte einer Sprache gibt. Hingewiesen sei hier im Kontext von Fehlerbehandlung auf die „Natural Order Hypothesis“ von Krashen (1985), die sich stark auf die Fehleranalyse stützt, und die anschließend einsetzende Diskussion über diese.[8] Die Hypothese besagt, „that we acquire the rules of language in a predictable order, some rules tending to come early and others late. The order does not appear to be determined solely by formal simplicity and there is evidence that it is independent of the order in which rules are taught in language classes.“(Krashen, 1985, S. 1).

Wie weit der Einfluss der Muttersprache tatsächlich geht und welche Form er hat, ist nicht einfach zu ermitteln und unter anderem deshalb umstritten. Butzkamm (1993, S. 111ff) geht zum Beispiel davon aus, dass die Muttersprache etwas größeren Einfluss auf die Phonologie und die Lexik der Fremdsprache hat und eher weniger Einfluss auf die Morphosyntax. Diehl et al. (2000, S. 27f) mit Bezug auf Ellis (1994, S. 343) sehen dagegen den Einfluss der L1 als erwiesen und sehr wesentlich: „Der prägende Einfluss der L1 für den L2-Erwerb ist inzwischen unbestritten. Transfer aus der L1 gilt nicht mehr [..] als möglichst zu eliminierender Störfaktor sondern als ‘kognitiv begründete Produktionsstrategie’“. Als Ergebnis dieser groß angelegten Untersuchung wird allerdings nur das vage Fazit gezogen, dass in einigen Bereichen wie zum Beispiel der Wortstellung der Einfluss der L1 deutlich erkennbar ist, allerdings nicht in der Morphologie (Diehl et al., 2000, S. 338f). Auch in der aktuellen Diskussion zur Didaktik des DaF[9]-Unterrichts spielt der Einfluss der L1 meines Wissens eher eine untergeordnete Rolle. Das ist insbesondere der Fall, wenn für den DaF-Unterricht aus didaktischen und sozialen Gesichtspunkten multinationale Gruppen bevorzugt werden, wobei eine besondere Berücksichtigung der Muttersprache dann allein aus praktischen Gründen kaum möglich ist. [10]

Einerseits deuten einige der angesprochenen Aspekte auf weitere Module hin, die in ein ICALL-System integriert werden können, wie zum Beispiel die Berücksichtigung der L1 oder die Nutzung eines Usermodels zur Identifizierung des Spracherwerbsstandes des Lerners. Andererseits würde mit einem Modul Usermodel ein so weites Feld eröffnet, das es im Rahmen dieser Arbeit nicht berücksichtigt werden kann. Obwohl also zum Bereich EA insbesondere

[8] Verschiedene weitere Hypothesen, die Krashen und andere in diesem Zusammenhang aufgestellt haben (Language Acquisition Device etc.), sollen hier unberücksichtigt bleiben.

[9] Deutsch als Fremdsprache

[10] Angeführt werden kann hier auch die Diskussion zu „immersion programs“, in denen die L1 im Wesentlichen keine Berücksichtigung findet.

aus der psycholinguistischen Perspektive viele Forschungsergebnisse vorliegen, soll hier die didaktische Perspektive im Vordergrund stehen. Schließlich kann angemerkt werden, dass in der von mir gesichteten Literatur zur allgemeinen Didaktik der DaF-Fehlerbehandlung kaum eine Berücksichtigung der L1 stattfindet und aus diesen Gründen in dieser Arbeit weitgehend unberücksichtigt bleiben soll.

2.2.2 Fehlerbehandlung in mündlicher und schriftlicher Unterrichtssituation

Bevor auf die speziellen Aspekte der Fehlerbehandlung eingegangen wird, muss zunächst die Betrachtungsweise der „Unterrichtssituation“ geklärt werden. Obwohl in der von mir rezipierten Literatur zumeist lediglich zwischen einer mündlichen und schriftliche Unterrichtssituation unterschieden wird, kann insbesondere bei der Betrachtung des implementierten Übungstyps die Unterscheidung nicht ausreichen. Zunächst lässt sich der Modus der Kommunikation identifizieren, bei dem in einen mündlichen und einen schriftlichen Modus unterschieden werden kann. Mündlich gesehen ist eine Äußerung dann tatsächlich auditiv wahrnehmbar, während sie schriftlich gesehen nur visuell wahrgenommen, das heißt gelesen werden kann. Orthogonal dazu kann hinsichtlich der Struktur der Äußerung unterschieden werden, das heißt, strukturell kann eine Äußerung dialogähnlich sein und doch schriftlich geäußert worden sein, wie es zum Beispiel im Internet-Chat der Fall ist. Andererseits kann der Modus mündlich sein und doch in der Struktur einem Text entsprechen, wie zum Beispiel im Fall eines vorgelesenen Texts.

Diese Unterscheidungen werden in der Literatur zur Fehlerbehandlung im FU nicht gemacht: „Mündlich“ bezieht sich immer auf eine mündliche, dialogähnliche Situation oft mit dem Lehrer als Dialogpartner und „schriftlich“ bezieht sich auf die schriftliche, textähnliche Äußerung eines Lerners. In Bezug auf den konkreten FU lassen sich demnach also zunächst zwei wesentliche Bereiche unterscheiden: die Behandlung eines Fehlers beziehungsweise einer fehlerhaften Äußerung in einer mündlichen vs. einer schriftlichen Unterrichtssituation.

Hier bestehen offensichtliche und grundsätzliche Unterschiede. Unter anderem erfolgt die Begutachtung einer schriftlichen Leistung durch den Lehrer normalerweise immer erst nach dem Ende des gesamten Schreibprozesses und auch die „Rückmeldung“ an den Lerner erfolgt üblicherweise durch die Rückgabe des gesamten Textes. In einer mündlichen Unterrichtssituation besteht diese Beschränkung nicht. Ein weiterer Unterschied besteht in der Spontaneität zwischen einer mündlichen und einer schriftlichen Äußerung im FU. Während eine mündliche Äußerung üblicherweise relativ spontan geschieht und meistens in eine kommunikative Situation eingebunden ist, die selten Zeit für Fehleranalysen lässt, entstehen schriftliche Äußerungen meist mit genügend Zeit für Reflexionen über die Korrektheit eines Satzes.

Obwohl in Promis*D* die Eingabe mit Hilfe der Tastatur erfolgt, erscheint mir die Betrachtung der Korrektur von mündlichen, dialogähnlichen Äußerungen wichtig zu sein. Bei den im Programm realisierten Aufgaben geht es nicht in erster Linie um die Produktion von Text, sondern um die Simulation von Frage-Antwort-Dialogen, die unter anderem die Beantwortung mehrerer, aufeinander aufbauender Fragen erfordert. Es kann dadurch trotz des schriftlichen Modus eine Art von Kommunikationfluss entstehen, der im Idealfall nicht durch störende Fehlermeldungen und sonstige Hinweise unterbrochen werden sollte. Die Ähnlichkeit mit einer mündlichen, dialogähnlichen Unterrichtssituation beziehungsweise einer realen Kommunikationssituation ist also in dieser Hinsicht relativ groß. Desweiteren stehen hier nicht so sehr die Möglichkeiten der Gewichtung oder der Form der Fehlermarkierung im Vordergrund, wie es bei der Korrektur von schriftlichen, textähnlichen Arbeiten der Fall wäre.

Allerdings ist das System in erster Linie in der Lage, morphosyntaktische Strukturen zu analysieren und dazu Rückmeldungen zu geben. Insbesondere ist kein Spracherkenner integriert, der gesprochene in geschriebene Sprache umsetzen könnte. Wenn der Nutzer also relevantes und sinnvolles Feedback erhalten möchte, muss er Schriftsprache verwenden, die vom System in geeigneter Weise analysiert werden kann. Damit ordnet sich der Aufgabentyp der „schriftlichen Dialoge“ in einen Grenzbereich mit Eigenschaften sowohl einer Äußerung im schriftlichen wie auch im mündlichen Modus ein. Aus den genannten Gründen wird ganz wesentlich die Behandlung von fehlerhaften Äußerungen und den enthaltenen Fehlern in einer mündlichen, dialogähnlichen Unterrichtssituation im Vordergrund stehen und es wird an der verkürzten Wortwahl der Didaktikliteratur festgehalten und von mündlichen (das heißt mündlichen, dialogähnlichen) und schriftlichen (das heißt schriftlichen, textähnlichen) Äußerungssituationen gesprochen. Zum Abschluss des Abschnitts 2.2 soll auf Orthografiefehler als spezielles Problem, das nur in schriftlichen Äußerungen auftreten kann, eingegangen werden, da dieser Bereich in den darauffolgenden Kapiteln nur am Rande berücksichtigt wird.

2.2.3 Machen ausführliche Fehlermeldungen überhaupt Sinn?

Eine zentrale Dimension der Fehlermeldung ist die Ausführlichkeit, mit der auf einen Fehler eingegangen wird. Daher wird dieser Aspekt hier etwas näher beleuchtet. Einem Lehrer im FU steht üblicherweise ein Kontinuum an Möglichkeiten der Fehlerbehandlung zur Verfügung: Vom völligen Ignorieren einer fehlerhaften Äußerung bis zur intensiven Erörterung eines jeden Fehlers. Die Frage, die sich hier anschließt, ist, ob man aus didaktischer Perspektive Empfehlungen zu bestimmten Fehlerbehandlungsstrategien in bestimmten Unterrichtssituationen geben kann. Wie Königs (1995, S. 271) ausführt, existiert zu diesem Bereich keine empirische Forschung, was vor allem auf der schwierigen Fragestellung beruht: Basieren bestimmte Lern-

resultate auf einer bestimmten Fehlerbehandlung und nur auf dieser? Als ein relevantes Beispiel sei Nagata (1993) angeführt, die trotz der offensichtlichen Schwierigkeiten versucht, mit Hilfe eines ICALL-Systems herauszufinden, ob Lerner bestimmte syntaktische Konstruktionen des Japanischen besser lernen, wenn ihnen in den Übungen ausführliche Fehleranalysen präsentiert werden. Im Gegensatz zu einer Kontrollgruppe, die nur spärliche Rückmeldungen bekam, hat die Versuchsgruppe in einem abschließenden gemeinsamen Test zwar nur wenig, aber dennoch statistisch signifikant besser abgeschnitten. Zu kritisieren ist an dieser Untersuchung, dass die Anzahl der Teilnehmer in der Versuchsgruppe und der Kontrollgruppe jeweils nur 17 betrug. Meines Erachtens ist das erzielte Ergebnis damit nur eingeschränkt aussagekräftig.

Eine weitere Schwierigkeit bei der Berücksichtigung der Ausführlichkeit besteht vermutlich in den multiplen Lernzielen, die mit unterschiedlicher Gewichtung in einer bestimmten Übung verfolgt werden. Da zum Beispiel in einer Dialogübung in erster Linie das Funktionieren des Turn-Takings beachtet wird, muss der Lehrer entscheiden, welches Gewicht, das heißt, welche Ausführlichkeit er einer möglichen Fehlerbehandlung beispielsweise von morphosyntaktischen Fehlern zubilligt, oder ob dieser Fehlertyp ganz ignoriert werden soll. Dieses Abwägen der Lernziele und der damit verbundenen Ausführlichkeit der Rückmeldung stellt sich für ein ICALL-Programm mit dialogorientierten Übungen nur in begrenzten Maße als ein Problem dar. Wenn die Leistung der Fehleranalyse zwangsläufig durch die technischen Methoden eingeschränkt ist, bietet es sich meines Erachtens an, alle möglichen Fehlertypen zu identifizieren, es aber zumeist dem Lerner zu überlassen, wann und welche Fehlermeldungen er sich anschauen möchte, solange der Kommunikationsfluss nicht durch unanalysierbare Eingaben gestört wird.

Welche weiteren Konsequenzen sich für den Übungstyp des simulierten Dialogs in Promis*D* ergeben, soll ausführlich im nächsten Unterkapitel 2.3 diskutiert werden. Im Folgenden werden die verschiedenen Dimensionen der Fehlerbehandlung und deren Einbettung in den Unterricht genauer beleuchtet. Wenn dabei die Rede von der „Korrektur“ einer fehlerhaften Äußerung ist, muss berücksichtigt werden, dass im FU eher selten auf den speziellen Typ eines Fehlers eingegangen wird, mit anderen Worten, es wird in der Literatur tatsächlich kaum eine Unterscheidung zwischen unterschiedlichen Formen der Rückmeldung gemacht, die beispielsweise nur eine korrigierte Form enthalten kann, oder aber auch nähere Informationen zum Beispiel zum morphosyntaktischen Fehlertyp umfasst. „Korrektur“ muss streng genommen als die Rückmeldung verstanden werden, die lediglich eine korrigierte Form enthält, auch wenn im FU möglicherweise auch manchmal „Fehlererklärungen“ vom Lehrer präsentiert werden. Im Folgenden soll der Begriff „Korrektur“ wie auch in der Literatur verwendet werden, das heißt, es geht vor allem um die Korrektur. Schließlich muss angemerkt werden, dass selten zwischen Kompetenz- und Performanzfehlern unterschieden wird. Als Kompetenzfeh-

ler werden solche Fehler bezeichnet, bei denen man davon ausgehen kann, dass der Sprecher die korrekte Form nicht kennt, während Performanzfehler solche sind, bei denen der Sprecher eigentlich die korrekte Form kennen sollte, sie aber in der aktuellen Äußerungssituation nicht produziert hat. Im Folgenden wird daher diese Unterscheidung bis auf die Betrachtung zur Fehlerbehandlung von schriftlichen, textähnlichen Äußerungen nicht weiter berücksichtigt.

2.2.4 Möglichkeiten der Rückmeldung in einer mündlichen Unterrichtssituation

Hendrickson (1980, S. 156) präzisiert die Dimensionen zur Behandlung von Fehlern, indem er fünf Fragen formuliert, die bei der Gestaltung der Fehlerkorrekturmethode Beachtung finden sollten:

1. Should learner errors be corrected?
2. If so, when should learner errors be corrected?
3. Which learner errors should be corrected?
4. How should learner errors be corrected?
5. Who should correct learner errors?

Die Antworten zu diesen Fragen ergeben Empfehlungen für die sinnvolle Behandlung von Fehlern im Unterricht. Die Konsequenz aus diesen Empfehlungen für ein Computerprogramm, das in der Lage ist, in bestimmten Bereichen ausführliche Fehleranalysen zu liefern, ist der korrespondierende Einsatz einer Fehlerbehandlung in dem System.

Mit der Erstellung dieser Arbeit ist die Beantwortung der ersten Frage, ob überhaupt eine Rückmeldung wegen eines Fehlers erfolgen sollte, quasi positiv vorweggenommen. Obwohl die Frage für die Didaktik des FUs von erheblicher Bedeutung ist, spielt sie für den Kontext dieser Arbeit nur eine untergeordnete Rolle, da es gerade um die Möglichkeiten zur präzisen und umfangreichen Rückmeldung geht. Das gilt insbesondere dann, wenn ein ICALL-System in der Lage ist, die Fehler in den Eingaben eines Lerners an der Oberfläche zu ignorieren, sie aber trotzdem einschließlich der Analysen zu speichern. Auf diese Weise kann es dem Lerner überlassen werden, ob er die Fehlermeldung direkt oder erst nach der Bearbeitung der gesamten Aufgabe bearbeiten oder auch löschen möchte.

Wann sollten Fehler korrigiert werden? Relativ unabhängig von den anderen Fragestellungen kann der Zeitpunkt der Korrektur in einer mündlichen Unterrichtssituation prinzipiell an drei verschiedenen Punkten erfolgen (Kleppin, 1998, S. 107f):

- sofort nach Auftreten des Fehlers
- nach Vollendung der Äußerung
- in einer eigenständigen Korrekturphase

Die sofortige Korrektur hat den Vorteil, dass der Lehrer sich den Fehler nicht merken muss. Außerdem fällt dem Lerner so die Verknüpfung der Korrektur mit der fehlerhaften Äußerung leicht. Andererseits wird der Redefluss meistens deutlich unterbrochen, wenn der Lehrer nicht signalisiert, dass seine Korrektur nicht vom Lerner reflektiert werden muss. Dagegen hat die Korrektur nach Vollendung der Äußerung den Vorteil, dass der Lerner erst unterbrochen wird, wenn auch in der Kommunikation eine Zäsur vorhanden ist. Als eine weitere sinnvolle Möglichkeit erscheint eine indirekte Korrektur zum Beispiel durch eine korrigierende Echofrage, die den Kommunikationsfluss weiterlaufen lässt und gewährleistet, dass der Lerner die Korrektur versteht.

Die beiden ersten oben genannten Möglichkeiten haben den Nachteil, dass die Bewusstwerdung des Fehlers beim Lerner unter Umständen nur sehr kurze Zeit anhält. Es ist gut möglich, dass der Lerner den gleichen Fehler in der nächsten Äußerung wiederholt. Ouanes (1992) plädiert aus diesem Grund für die letzte Möglichkeit einer eigenständigen Korrekturphase, in der dann eine „Fehlertherapie“ stattfinden kann. Für die Behandlung von Fehlern, die in Dialogübungen nicht den Kommunikationfluss zum Scheitern bringen können, bietet sich meines Erachtens eine Kombination an. Einerseits sollte ein Fehler zum Beispiel mit Hilfe einer Echofrage dem Lerner indirekt angezeigt werden. Wenn die Konzentration dann nicht völlig auf die Kommunikation gerichtet ist, hat der Lerner die Möglichkeit, den Fehler zu registrieren und aufzunehmen. Im Anschluss kann dann eine „Fehlertherapie“ erfolgen, indem auf die für die Kommunikation zwar unerheblichen, aber für den Lernprozess wichtigen Fehler eingegangen wird.

Welche Fehler sollten korrigiert werden? Mit dem oben erwähnten Aspekt der Ausführlichkeit der Korrektur ist eng die Frage nach der Auswahl der zu behandelnden Fehler verknüpft, die hier meines Erachtens im Sinne von Fehlertypen verstanden werden müssen. Da nicht alle Fehler korrigiert werden können und sollten, muss wie auch bei der Ausführlichkeit eine Selektion entsprechend der Lernziele stattfinden. Wichtige Fehler, die nach bestimmten Kriterien, zum Beispiel der Häufigkeit oder der Kommunikationsstrategie auffällig sind, sollten dabei im Vordergrund stehen, während manch anderer Fehler ignoriert werden kann. In einer mündlichen Unterrichtssituation bedeutet dies, dass Fehler, die die Kommunikation behindern oder scheitern lassen können, mit höherer Priorität behandelt werden müssen als zum Beispiel kleinere syntaktische Fehler, die die Bedeutung einer Äußerung nicht berühren.[11] Ein weiteres Kriterium für die Entscheidung, ob ein Fehlertyp korrigierenswert ist, könnte auch das gerade im Unterricht

[11] Hendrickson (1980, S. 166f) weist darauf hin, dass die Korrektur üblicherweise immer auf bestimmte Typen von Fehlern zielt. Korrekturen von schriftlichen Arbeiten würden sich vermutlich eher auf die Form konzentrieren, während Korrekturen von mündlichen Äußerungen sich eher auf die Funktion und damit auf funktionale Wörter beziehen würden.

behandelte Thema sein. Wenn zum Beispiel einzelne grammatische Phänomene im Mittelpunkt stehen, bietet es sich an, dass der Lehrer insbesondere in Bezug auf diese Phänomene korrigiert.

Durch den Einsatz von computerlinguistischen Methoden zur Erkennung der Fehler ist allerdings eine bestimmte Klassifikation, die durch die Möglichkeiten des Analysemoduls und des verwendeten Grammatiktyps (Kapitel 3) determiniert ist, schon vorgegeben. Der Schwerpunkt bei Promis$D$, der im Kapitel 5 ausführlich behandelt wird, liegt auf der Erkennung und Erklärung von morphosyntaktischen Fehlern. Daher erfolgt hier keine Klassifikation von Fehlern, sondern es wird auf die Beschreibung des Umfangs des Fehlererkennungsmoduls verwiesen.

Hier muss angemerkt werden, dass „Normen im Fremdsprachenunterricht“ (und auch Akzeptabilität) ein nicht behandelter Aspekt dieser Arbeit sind. Eine Sprachnorm, die die Form der Sprache im Unterricht vor allem in der soziolinguistischen Ebene festlegt, wird im vorliegenden Fall durch die in das System integrierte Grammatik und die Dialogstrukturierung bestimmt, da mit deren Hilfe ein festgelegtes Sprachfragment beschrieben wird. In einer Unterrichtssituation geht man davon aus, dass der Lehrer üblicherweise die zu verwendende Norm festsetzt. Dagegen wird die Norm bei der Bearbeitung des simulierten Dialogs durch die Möglichkeiten der Analysemodule bestimmt. Im Zusammenhang damit besteht die Einsicht, dass der Lerner eine korrekte fremdsprachliche Äußerung machen kann, die nicht als solche erkannt werden kann. Der Lehrer ist in der überwiegenden Zahl der Fälle die Instanz, die die Akzeptabilität bestimmt, und eine Äußerung, die über die Kompetenz des Lehrers hinausgeht, kann von ihm als falsch oder zumindest als marginal gewertet werden. Dasselbe gilt offensichtlich in besonderem Maße auch für ein ICALL-Programm und wird in Abschnitt 5.1 erläutert.

Wie sollten Fehler korrigiert werden? In Kleppin (1998) werden die Möglichkeiten zum Umgang mit fehlerhaften Lerneräußerungen in verschiedenen Unterrichtssituationen erläutert, wobei angemerkt werden muss, dass die Frage nach dem „Wie?“ sehr eng mit dem „Wann?“ verknüpft ist und daher eine eindeutige Trennung schwierig ist. In so genannten „freien Phasen“ stehen die freien, wenig gelenkten Äußerungen der Lerner im Vordergrund. Eine selbstverständliche Empfehlung lautet daher, wie schon erwähnt möglichst wenig auf Fehler einzugehen, um die Kommunikation nicht unnötig zu unterbrechen, solange sie gelingt; die Mitteilung in der Äußerung steht schließlich im Vordergrund. In freien Phasen gar nicht zu korrigieren, ist nach Kleppin ein unzureichender Umkehrschluss. Gerade in diesen Phasen treten typische Fehler auf, die sich in stark gelenkten Unterrichtsphasen nicht ergeben. Es bietet sich allerdings die Möglichkeit an, die Fehler zu notieren und von der kommunikativen Situation getrennt zu behandeln. Solch eine korrekturfreie Phase sollte natürlich vom Lehrer angekündigt werden.

Meist werden zwei Arten von *Lehrer*korrekturen unterschieden: direkte und indirekte Lehrerkorrekturen. Im ersten Fall korrigiert der Lehrer die Äußerung der Lerners sofort, indem er zum Beispiel die Äußerung korrekt wiederholt und damit als bewusste Korrektur deutlich macht. Im zweiten Fall wird die Äußerung des Lerners vom Lehrer aufgegriffen und in korrekter Form zum Beispiel als Echofrage wiederholt (siehe unten). Beide Arten von Lehrerkorrekturen haben im Unterricht Vor- und Nachteile. So kann man nicht davon ausgehen, dass direkte Korrekturen grundsätzlich den Kommunikationsfluss unterbrechen. Direkte Korrekturen können auch so angebracht werden, dass der Eingriff in die eigentliche Kommunikationssituation minimal bleibt. Wie Macht (1998, S. 363) allerdings ausführt,

> „[ist] die verbreitetste Form der Fehlersignalisierung im mündlichen Diskurs [...] leider nach wie vor der direkte didaktische Eingriff durch die Lehrperson [...]. Die nimmt in der Regel die Form der unterbrechenden Berichtigung an, worauf häufig von der betreffenden Schülerin verlangt wird, die berichtigte Form mechanisch zu wiederholen. Wenn dieses Vorgehen auch noch mit einem Tadel verbunden ist, so wird damit einem kommunikativ-handelnden Sprachunterricht jede Basis entzogen."

Andererseits verfehlen indirekte Korrekturen unter Umständen ihre Wirkung, wenn dem Lerner die Korrektur verborgen bleibt, weil er zum Beispiel mit der Planung einer weiteren Äußerung beschäftigt ist. Eine Reihe von Korrekturstrategien werden von Kleppin (1998, S. 93f) und Lyster und Ranta (1997, S. 46ff) vorgeschlagen:

- Fehler können explizit korrigiert werden.
- Fehler können indirekt korrigiert werden. Beispiel:
  Lerner: „Ich habe eine Frage zu dir."
  Lehrer: „Aha, du hast eine Frage an mich. Und welche Frage?"
- Auf fehlerhafte Äußerungen hin kann zum Beispiel mit „Pardon?" nachgefragt werden.
- Fehler können diskret notiert und zu einem späteren Zeitpunkt erörtert werden.
- Fehler können offensichtlich notiert (zum Beipiel auf einer Folie) und zu einem späteren Zeitpunkt erörtert werden.
- Gestik und Mimik können einen Sprecher auf einen Fehler hinweisen, überlassen die Entscheidung über eine Unterbrechung der Kommunikation aber dem Sprecher.
- Bestimmte Phasen im Unterricht kommen unter Umständen ganz ohne Korrekturen aus.

Die Frage, die sich anschließt und beantwortet werden sollte, lautet, welche Umstände die Auswahl einer bestimmten Strategie erfordern. Das Wissen darüber ist vermutlich bei einem Lehrer diffus vorhanden, da, wie schon

öfter erwähnt, die Lernziele im Vordergrund stehen. Für die Erreichung dieser Lernziele sind bestimmte Fehler von größerer Wichtigkeit, die dann vom Lehrer entsprechend offensichtlich hervorgehoben werden können.[12]

Aus einer konstruktivistischen Perspektive ist das Vorgeben von korrekten „Lösungen“ nicht der geeignete Weg, dem Lerner Wissen zu vermitteln. In diesem Konzept soll der Lerner selbsttätig sein Wissen konstruieren. „They [die Verfechter eines entdeckenden Lernens] propose that a discovery approach to error correction might help students to make inferences and formulate concepts about the target language, and to help them fix this information in their long-term memories.“(Hendrickson, 1980, S. 163) Demnach sollte der Lerner in jedem Fall nur auf einen Fehler hingewiesen und der folgende Korrekturprozess als Wissenserwerb gesehen werden.

Dem Lehrer als Korrektor steht also eine Bandbreite an Möglichkeiten zur Hervorhebung und zur Korrektur von fehlerhaften Äußerungen zur Verfügung. Wann allerdings welche Methode eingesetzt wird, hängt wie erwähnt in erster Linie von den angestrebten Lernzielen ab. Schließlich kann in diesem Zusammenhang der Aspekt der Lernerzentriertheit genannt werden, der möglicherweise eine Anpassung der Korrekturmethode verlangt. Ein Fehler sollte also so korrigiert werden, dass die Korrektur an den Kenntnisstand des Lerners über die Sprache angepasst ist und seine sprachliche Kompetenz sowie Performanz berücksichtigt.

Wer sollte Fehler korrigieren? Für die Behandlung von mündlichen Fehlern im Unterricht mit der Frage nach dem „Wer sollte korrigieren?“ kann folgende grundsätzliche Typologie von Korrekturen angenommen werden (Henrici und Herlemann (1986), zitiert nach Kleppin (1998), S. 87):

- selbstinitiierte Selbstkorrektur
- selbstinitiierte Fremdkorrektur
- fremdinitiierte Selbstkorrektur
- fremdinitiierte Fremdkorrektur

Zwei unterschiedliche Parameter werden hier variiert: der Auslöser des Korrekturprozesses und der Korrektor. Mit dem Begriff „Fremd-“ wird hier sowohl der Lehrer als auch ein Mitschüler verstanden. In eigenen Untersuchungen hat Kleppin (Kleppin und Königs, 1991, S. 291ff) unter anderem herausgefunden, dass sich Schüler ungern von Mitschülern, sondern lieber vom Lehrer korrigieren lassen, wenn es um Fremdkorrekturen geht. Dieses Ergebnis wird von Gnutzmann und Kiffe (1993) mit Hilfe einer Befragung

[12] In einer Untersuchung haben Gnutzmann und Kiffe (1993) versucht zu analysieren, welche Art von Fehlerbehandlung Lerner als positiv bewerten und daher bevorzugen. Es hat sich herausgestellt, dass Lerner durchaus gerne kommunikative Situationen zu Ende führen, ohne vom Lehrer unterbrochen zu werden. Andererseits war den Lernern bewusst, dass sich durch unterbliebene Korrektur womöglich falsches Wissen über die Sprache festigt.

von Studenten bestätigt. Auch sie lassen sich lieber vom Lehrer als von Kommilitonen korrigieren (siehe allerdings Uesseler (1992), der gegenteilige Beobachtungen gemacht haben will).

Die Selbstkorrektur hat den entscheidenden Vorteil, dass damit der Umgang mit eigenen fehlerhaften Äußerungen und mit bestimmten sprachlichen Strukturen gefördert wird. Insbesondere die im vohergehenden Unterkapitel erwähnte Sprachbewusstheit kann meines Erachtens mit Hilfe der Selbstkorrektur trainiert werden. Der Lerner ist durch die Korrektur der eigenen Äußerung gezwungen, sich über eine korrekte Form oder auch über eine alternative Formulierung Gedanken zu machen.

2.2.5 Fehlerrückmeldung im schriftlichen Bereich: Orthografie

Bisher wurde als Schwerpunkt die Behandlung von Fehlern im mündlichen Bereich analysiert. Allerdings kann Promis*D* nur Eingaben über die Tastatur verarbeiten und weicht insofern wie erwähnt von einer mündlichen Kommunikationssituation ab. Außerdem ist das System nur sehr bedingt in der Lage, neue, das heißt unbekannte Wörter aufzunehmen und in die Analyse mit einzubeziehen. Schließlich wird der Lerner aufgefordert, ganze Sätze einzugeben, um auf die Fragen des Systems zu antworten, obwohl es sich um eine Situation ähnlich eines Dialogs handelt, der bekanntlich auch unvollständige Sätze enthält beziehungsweise enthalten kann. Während auf den vorhergehenden Seiten der allgemeine Umgang mit Fehlern in mündlichen Unterrichtssituationen diskutiert wurde, muss zur Behandlung eines konkreten Fehlers der Blick daher auch auf den schriftlichen Bereich gelenkt werden. Verschiedene zusätzliche Fragestellungen ergeben sich aus der Betrachtung der schriftlichen Situation (nach Kleppin, 1998, S. 54).

- Soll ein Fehler markiert werden?
- Wird ein Fehler nur markiert oder der Typ hinzugefügt oder gar korrigiert?
- Was folgt aus den korrigierten Sätzen?

Für die Frage, ob ein Fehler markiert werden sollte, gilt im Prinzip die Empfehlung, jeden Fehler anzumerken, es sei denn, der Lerner hat explizit die Aufgabe, einen Text selber zu korrigieren. Die Anmerkung eines jeden Fehlers ist notwendig, da der Lerner ansonsten annehmen könnte, seine Produktion sei korrekt. Ob eine Korrektur zusätzlich zu einer Markierung durch den Lehrer erfolgen sollte, hängt nach Kleppin (1998, S. 60f) vom Typ des Fehlers ab, das heißt, ob es sich um einen Performanz- oder Kompetenzfehler handelt.

- Wenn es sich um einen versehentlichen Fehler handelt, sollte nur eine Markierung erfolgen und dem Lerner die Korrektur überlassen werden.

- Wenn der Lerner den Fehler selber korrigieren kann, sollte er zwar mit einem Zeichen für den Fehlertyp versehen werden, aber nicht korrigiert werden.
- Falls es sich um einen tatsächlichen Kompetenzfehler handelt, sollte eine korrigierte Form hinzugefügt werden.

Die Frage nach der weiteren Nutzung der korrigierten Arbeiten muss meines Erachtens von der Unterrichtsplanung abhängig gemacht werden. In jedem Fall sollte wie erwähnt eine Sprachstandsanalyse durchgeführt werden, die Auskunft über den momentanen Stand und über die individuellen Problembereiche geben kann. Mit Hilfe dieser Empfehlungen lässt sich eine gezielte Fehlerbehandlung im schriftlichen Bereich durchführen.

2.2.6 Zusammenfassung

Zusammenfassend lässt sich sagen, dass konkurrierende Anforderungen von Seiten der Methoden zur Förderung der kommunikativen Kompetenz sowie von Seiten des Ziels der Verwendung korrekter Sprache besteht. Für das Erlernen kommunikativer Fähigkeiten ist die phonologische und morphosyntaktische Korrektheit von untergeordneter Bedeutung, solange der Informationsaustausch funktioniert. Fehler werden deshalb in kommunikationsorientierten Übungen nur relevant, wenn die Kommunikation zu scheitern droht, was eher bei semantisch-pragmatischen Fehlleistungen der Fall ist als bei anderen Fehlertypen. Zum Erlernen von „korrekter“ Sprachanwendung gehören aber auch ganz wesentlich morphosyntaktisch korrekte Äußerungen, die üblicherweise mit Hilfe von schriftlichen Übungen trainiert werden.

Aus diesem Konflikt ergibt sich für Lehrer im Unterricht die Empfehlung, in den verschiedenen Übungstypen Fehler unterschiedlich zu behandeln, je nachdem, welches Lernziel im Vordergrund steht. Wenn zum Beispiel die kommunikativen Fähigkeiten im Vordergrund stehen, kann also vermutlich größtenteils entweder auf die Anmerkung von morphosyntaktischen Fehlern verzichtet werden oder nur schwache Hinweise auf inkorrekte Formen gegeben werden. Umgekehrt muss in mündlichen oder wohl meist schriftlichen Übungen, bei denen die Verwendung korrekter Sprache im Vordergrund steht, jeder Fehler in einer jeweils angemessenen Form behandelt werden.

Zur Behandlung von Fehlern lässt sich außerdem festhalten, dass zwar einerseits die eigenständige Korrektur zur Konstruktion von Wissen führt, dass aber andererseits dieses auch das Memorieren von fehlerhaften Strukturen zur Folge haben kann. Wenn dagegen der Lehrer die Korrekturen vornimmt, besteht vor allem in mündlichen Unterrichtssituationen die Gefahr, dass der Lerner die Korrektur nur flüchtig wahrnimmt, weil kein Aufwand damit verbunden ist. Aus dieser Schwierigkeit ergibt sich die angesprochene Idee der intensiven Behandlung der Fehler in einem zeitlich gesonderten Abschnitt („Fehlertherapie“). Zur Förderung der im vorhergehenden Abschnitt 2.1 erwähnten Sprachbewusstheit als ein Lernziel bietet es sich an,

beispielsweise in einem speziellen Abschnitt alle Fehler ausführlich anzumerken und über die Korrektur hinaus zu erläutern, um so die strukturellen Eigenschaften der Sprache zu verdeutlichen. Schließlich muss erwähnt werden, dass die Art der Fehlerbehandlung offensichtlich auch vom Kenntnisstand des Lerners abhängig sein sollte, das heißt lernerzentriert erfolgen sollte.

2.3 Resultierende Konsequenzen für ein Sprachlernprogramm

Die folgenden Zitate zeigen, dass Didaktiker, die traditionelle Sprachlernprogramme tatsächlich im Unterricht verwenden, besonders auch die Methoden der Fehlerbehandlung für unzureichend halten.

> „Die wichtigste Frage, an deren positiver Beantwortung letztlich die Akzeptanz von multi- und telemedialen Selbstlern-, und Zusatzmaterialien abhängen wird, lautet jedoch:
>
> – Wird es möglich sein, das häufig katastrophale Feedback aller Lernprogramme auf CD-Rom und im Internet so zu verbessern, dass Rückmeldungen auf Fehler den Lerner auch weiterbringen und nicht mehr nur zu dauernder Frustration führen?“
>
> (Rösler und Tschirner, 2002, S. 151)

> „Vielleicht haben Sie mit diesem kleinen Beispiel schon eine Vorstellung davon bekommen, worin ein großes Problem für die Autoren solcher Programme liegt: Sie müssen sich mögliche Eingaben durch die Lernenden vorstellen, die entsprechende Rückmeldung für jede Lernerangabe gesondert vorsehen und in mühsamer Kleinarbeit einbauen. Besonders problematisch ist dies, wenn zahlreiche Lösungen möglich sind. Obwohl sich im Laufe der Jahre die Möglichkeiten der Programme zur differenzierteren Fehlerauswertung verbessert haben, gibt es bisher noch keine zufrieden stellenden automatische Fehleranalyse. Die Konsequenz ist, dass die Führung durch das Programm verstärkt wird, um den Arbeitsaufwand und die Fehlerquote gering zu halten.“ (Grüner und Hassert, 2000, S. 53)

> „Es gibt Beschränkungen, die durch die Software bedingt sind: Das betrifft besonders die Fehlerbehandlung und -auswertung, und das Angebot an Lösungshilfen; die Möglichkeiten der individualisierten Lernersteuerung und Fehlerbehandlung werden oft nicht ausreichend genutzt. So werden auch bei vielen neueren Programmen zu wenig Lösungshilfen angeboten und die Fehlerauswertung auf eine bloße Richtig/Falsch-Analyse eingeschränkt.“ (Grüner und Hassert, 2000, S. 156)

> „Als nachteilig wurde allerdings schon sehr früh angesehen, dass aufgrund der vergleichsweise primitiven Programmstruktur keine wirklich angemessene Eingabeanalyse und Bewertung stattfindet. [...] In den neuesten Versionen traditioneller Übungsprogramme besteht die Möglichkeit, bei der Gestaltung der Übung bis zu zehn richtige Antworten vorzusehen; trotzdem ist damit das grundsätzliche Problem der Fehlerbewertung und Fehlerkorrektur natürlich nicht gelöst.“ (Rüschoff und Wolff, 1999, S. 69)

Mit computerlinguistischen Methoden zur Analyse der Eingaben sind zusätzlich zur intelligenten Fehlererkennung und -rückmeldung auch erweiterte Aufgabentypen möglich. Dieser zweite Aspekt soll im Weiteren dazu genutzt werden, an die moderne Didaktik (Abschnitt 2.1) angepasste Aufgabenstellungen zu entwerfen. Allgemeiner gesehen ergeben sich also in zwei Bereichen Möglichkeiten und Konsequenzen aus der Funktionalität der intelligenten Eingabeanalyse in einem ICALL-System, nämlich a) zur Form von Übungen und b) zur Behandlung von fehlerhaften Eingaben.

Aus obiger Kritik an bestimmten Programmtypen und an der Fehlererkennung im Speziellen sowie aus den didaktischen Konzepten lassen sich nun Anforderungen für ein tutorielles, CL-basiertes Sprachlernprogramm entwickeln, das sowohl Übungen zur Grammatik und Lexik als auch erweiterte Übungen wie simulierte Dialoge umfasst. Im Folgenden wird zunächst der Aufgabentyp des simulierten Dialogs insbesondere in Hinblick auf die Realisierung in Proims*D* betrachtet und erläutert. Dabei zeigt sich, dass ein solcher Übungstyp didaktischen Konzepten selbst dann genügen kann, wenn die Möglichkeiten innerhalb des Systems eingeschränkt sind. Anschließend erfolgen Ausführungen zu den Möglichkeiten der Fehlererkennung und dem Feedback.

2.3.1 Didaktik von Dialogaufgaben

Durch den Einsatz computerlinguistischer Methoden kann meines Erachtens in erster Linie der Einsatz neuer Übungsformen erreicht werden, die didaktischen Anforderungen besser genügen können als Aufgaben basierend auf herkömmlicher Technologie. Neben der linguistischen Aufbereitung authentischer Sprachdaten, die hier nicht weiter behandelt werden soll, stellt der Computer im (noch fiktiven) Idealfall einen „Gesprächspartner“ dar, der zusätzlich zur Dialogführung in der Lage ist, sprachliche Fehlleistungen des Lerners zu analysieren und zu bewerten sowie geeignetes Feedback zu generieren (Atwell, 1999). Realistischerweise kann mit Methoden der CL wie erläutert neben anderen Vorteilen insbesondere die Funktionalität des Programms für den Lerner gesteigert werden, indem eine intelligente Fehleranalyse in dialogähnlichen Aufgaben zusätzliches Feedback mit einer Fehlererklärung und gegebenenfalls eine Korrektur liefert.

Zunächst sollen hier die „Bausteine des modernen FU“ (Abschnitt 2.1, Seite 10) wieder aufgegriffen werden, um zu zeigen, in welcher Form die unterschiedlichen Lerntypen von einem Programm wie Promis*D* unterstützt werden können, in dem auf der Basis von Methoden der CL ein simulierter Dialog realisiert wird. Damit soll auch gezeigt werden, dass nicht nur streng konstruktivistische Ansätze eine Realisierung der Bausteine ermöglichen, sondern auch „tutoriell orientierte Anwendungen“ (Rüschoff und Wolff, 1999).

Wenn das Programm Aufgabentypen zur Verfügung stellen kann, die sich an Dialogen orientieren, ist damit ein *inhaltsbezogenes* sowie *aufgabenorientiertes* Lernen möglich. Der Lerner wird vom System aufgefordert, in einem ihm vermutlich bekannten Kontext sprachlich zu agieren. Zusätzlich kann der Kontext beziehungsweise die Situation nicht nur für die Bearbeitung einer einzigen Teilaufgabe aktiviert, sondern bleibt über einen längeren Zeitraum bestehen, wenn, wie beispielsweise im PROMISE-System (Bauer et al., 1994; John, 1994), mehrere Dialoge thematisch aufeinander aufbauen.[13] Damit wird meines Erachtens eine deutliche „Inhaltsbezogenheit“ erreicht. Aufgabenorientierte Übungen werden durch ein zu erreichendes Ziel definiert, wobei der Weg zum Ziel aber mehr oder weniger dem Lerner überlassen werden soll. Diese Definition kann auch auf die in Promis*D* präsentierten Übungen angewendet werden. Es wird eine Situation (Verkehrsunfall) geschildert und eine darauf aufbauende Handlungsanweisung (Unfallmeldung an die Polizei) gegeben, die es dem Lerner weitgehend überlässt, welche sprachlichen Mittel er zur Bearbeitung einsetzt. Wie man in Kapitel 6 sehen kann, ist die technische Realisierung so gewählt, dass es keine Rolle spielt, welche Elemente der Unfallsituation zuerst angesprochen werden. Zugegebenermaßen stellt die Bearbeitung eines kleinen Dialogs eine wenig komplexe Aufgabe dar, in der nur mit Mühe unterschiedliche Komponenten oder Arbeitsschritte identifiziert werden können.

Die geforderte *Authentizität* ist natürlich mit einem Computer als Dialogpartner nur annähernd zu erreichen. Allerdings wird der Lerner in dem simulierten Dialog aufgefordert, seine Sprachkenntnisse kreativ und selbständig einzusetzen, das heißt durch reines „Ausprobieren“ ist die Aufgabe nicht zu lösen. Die zur Zeit auf dem Markt befindlichen kommerziellen Systeme erfüllen gerade diesen Anspruch nicht. Das Konzept eines Programms wie FLUENT I/II (Hamburger, 1994, 1995) oder PROMISE (und damit auch Promis*D*) dagegen sieht zum Beispiel vor, dass der Lerner in Situationen agiert, die ein relativ hohes Maß an „authentischen“, wenn auch reduzierten Interaktionsformen bieten.

Offensichtlich ist ein *projektbasiertes* Lernen in einem solchen System

[13] In PROMISE wurde das Thema „Unfallbericht“ weiter ausgebaut, sodass zusätzlich Dialoge wie beispielsweise „Unfallrekonstruktion“ und „Gerichtsverhandlung“ thematisch auf dem ursprünglichen Dialog der „Unfallmeldung“ aufbauten und so das Rahmenthema weitergeführt haben.

nicht möglich, wohingegen meines Erachtens gewisse eine *Prozessorientierung* gegeben sein kann. Die Prozessorientiertheit der Bearbeitung einer Dialogaufgabe ergibt sich aus der komplexen Aufgabenstellung, die über einen „längeren“ Zeitraum hinweg in einem kontextuell geschlossenen Rahmen verfolgt wird.

Der Lernprozess in einer Dialogübung kann insofern *konstruktivistisch* sein, als dass der Dialog auf unterschiedlichen Wegen beendet werden kann. Über die unterschiedlichen Möglichkeiten kann dann das Modell der Fremdsprache konstruiert werden. Auch das Feedback des Systems kann von einem Lerner genutzt werden, zusätzliches Wissen über die Sprache zu gewinnen, da das Feedback vertiefte Einblicke in die Sprachstrukturen zulässt.

Schließlich erlaubt die Nutzung des Computers die Integration von *multimedialen* Elementen, die das Lernen auf unterschiedlichen Wahrnehmungskanälen fördert. Beispielsweise ist der Dialog in Promis*D* so konstruiert, dass die Reaktionen des Computers auf Lernereingaben fest in einer Datenbasis enthalten sind. Einerseits wird dadurch die Flexibilität des Systems eingeschränkt; andererseits aber wird auf diese Weise die akustische Wiedergabe der Reaktionen ermöglicht. Die geforderte *Offenheit* des Systems ergibt sich gerade nicht, da mit der Dialogsimulation eine eindeutige Aufgabenstellung verbunden ist, um mit Hilfe dieser Beschränkung dem simulierten Dialogpartner geeignete Reaktionen zu ermöglichen.[14]

Diese Aufstellung demonstriert, dass eine Reihe von „Bausteinen“ mit Hilfe der Dialogsimulation ausgefüllt werden können. Darüber hinaus können auch die angestrebten Ziele des FU – kommunikative Kompetenz, Sprachbewusstheit und Lernbewusstheit – zu einem System wie Promis*D* in Beziehung gesetzt werden. Insbesondere sollte bei der Bearbeitung eines Dialogs die kommunikative Kompetenz gefördert werden, da der Lerner wie in einer realen, kommunikativen Situation gefordert ist, sein Sprachwissen mit geringer Einschränkung des Kontextes anzuwenden. Da der Lerner darüberhinaus gefordert ist, grammatisch korrekte und vollständige Sätze zu verwenden, wird auch das Sprachbewusstsein gefördert. Zusätzlich liefert das Feedback des Systems Material für eine Reflexion der verwendeten sprachlichen Strukturen. Offensichtlich muss dagegen die Erlangung der Lernbewusstheit im Wesentlichen durch den Lehrer vermittelt werden.

2.3.2 Fehlerbehandlung

Die aus der computerlinguistischen Analyse resultierenden Strukturen, das heißt insbesondere Hinweise auf Fehler, sollten dem Lerner in geeigneter Weise präsentiert werden. Dazu wird in diesem Abschnitt zunächst allgemein auf die Integration von Fehlermeldungen in ICALL-Systeme und anschließend im besonderen in dialogorientierte Übungen eingegangen. Danach folgt die

[14] Die Anbindung von zusätzlichen Informationsquellen wie zum Beispiel der erwähnten Lernerenzyklopädie ist zwar nicht realisiert worden, aber prinzipiell möglich.

Betrachtung der Möglichkeiten zur Berücksichtigung der L1, der Behandlung von orthografischen Fehlern sowie der Möglichkeit der Selbstkorrektur, womit eine Bewertung des Übungstyps der simulierten Dialog aus unterschiedlichsten Perspektiven erreicht sein sollte.

Aus den oben gemachten Anmerkungen geht hervor, dass es einerseits wünschenswert ist, wenn ICALL-Programme ausführliche Fehleranalysen liefern können, um den Lerner zu unterstützen. Andererseits soll die Gestaltung von Dialogaufgaben didaktisch angepasst erfolgen und daher sollte das Programm nicht nach jeder fehlerhaften Eingabe den Dialog mit einer Fehlermeldung unterbrechen. In ICALL-Programmen, die zwar eine genaue Fehleranalyse liefern, aber keine dialogischen Übungen anbieten, muss dieser letzte Punkt natürlich nicht berücksichtigt werden. Die Übungen in Nagata (1993) oder Heift (2001) sind zum Beispiel streng satzbasiert, das heißt, der Lerner wird dazu aufgefordert, einen Satz kontextlos zu konstruieren. Entsprechend wird hier auch die Fehlermeldung dem Lerner direkt im Anschluss an die Eingabeanalyse präsentiert. Der Nutzen von ausführlichen Fehlermeldungen zeigt sich in beiden Publikationen. Wie erwähnt, beschreibt Nagata (1993) eine Untersuchung, in der Lerner mit Hilfe eines ICALL-Systems und ausführlichen Fehlermeldungen eine sprachliche Konstruktion sicherer beherrschten als eine Kontrollgruppe ohne genaues Feedback. Hierbei handelt es sich um eine der wenigen Untersuchungen, die gezeigt haben, dass eine Erklärung des Fehlers im Bereich von syntaktischen Konstruktionen den Lerner nachweislich beim Erwerb dieser unterstützten kann. Heift (2001) hat mit Hilfe von Web-Statistiken untersucht, inwieweit Fehlermeldungen eines Parsers tatsächlich von den Lernern gelesen wurden. Als Aufgabe wurden dem Lerner dabei Grundformen präsentiert, die zu einem sinnvollen Satz mit flektierten Wörtern zusammengesetzt werden sollten. Auch hier hat sich gezeigt, dass sich die Mehrheit der Nutzer des Systems anscheinend die Fehlermeldungen anschaut und die entsprechenden Korrekturen an der inkorrekten Eingabe vornimmt, das heißt, dass die Lerner also die Fehlermeldungen in Bezug auf die Grammatikalität in Korrekturen umsetzen konnten.

Morphosyntaktische Fehler sollten in einem dialogbasierten System, den didaktischen Vorgaben folgend, zwar dokumentiert werden, damit der Lerner in die Lage versetzt wird, die von ihm produzierten Äußerungen zu beurteilen. In einem simulierten Dialog jedoch sollten die Fehlermeldungen im Hintergrund bleiben, um einen möglicherweise entstehenden Kommunikationsfluss nicht zu unterbrechen, wenn das System mit hinreichender Sicherheit eine Analyse der Eingabe berechnen kann. Mit Hilfe von präzisen Fehlermeldungen, die auf bestimmten grammatischen Konzepten beruhen, wird dabei in Promis*D* nicht zuletzt die Erlangung der geforderten Sprachbewusstheit unterstützt.

Für den Fall, dass die Kommunikation scheitern sollte, ist im Unterricht die Möglichkeit zum Nachfragen gegeben. Der Lehrer kann zum Beispiel um eine veränderte Wiederholung der Äußerung bitten, um die Kommunikation

fortzusetzen. Eine ähnliche Strategie ist im vorliegenden System implementiert. Dem Lerner wird die Möglichkeit gegeben, eine zweite Eingabe zu machen, falls die erste Eingabe nicht als Antwort auf die zuvor gestellte Frage beziehungsweise als Fortsetzung des Dialogs interpretiert werden kann. Auf diese Weise wird ein „Nachfragen des Gesprächspartners“ simuliert, das auf einer Verstehensstörung basiert, indem die Form der Reaktion auf einen sehr groben Fehler beziehungsweise auf einen möglichen Abdeckungsmangel des Systems gerade nicht als Fehlermeldung (oder gar als Abbruch der Aufgabe) gestaltet ist.

Aus einer anderen Perspektive wird mit der Reaktion auf bestimmte fehlerhafte Eingaben durch den Dialogpartner eine Unterscheidung beziehungsweise Trennung von „Dialogpartner“ und „Sprachtutor“ erreicht. Während der simulierte Gesprächspartner auf bestimmte Fehler „im Dialog“ reagieren kann, besteht zusätzlich die Möglichkeit, präzisere Fehlermeldungen mit zusätzlichen Informationen zum Fehler separat zu präsentieren. Diese zusätzlichen Meldungen können als eine Art Sprachtutor verstanden werden, der relativ unabhängig vom Dialog relevante Informationen zu fehlerhaften Eingaben bietet.

Wenn die Fehleranalyse aus technischer Sicht im Rahmen eines ICALL-Systems näher betrachtet wird, lassen sich mehrere Stufen identifizieren (cf. Menzel, 2003). Fehler können demnach

- regristriert, (Gab es einen Fehler?)
- lokalisiert, (Wo war der Fehler?)
- identifiziert, (Was war falsch?)
- evaluiert, (Wie schwerwiegend war der Fehler?)
- erklärt/erläutert (Welche Rückmeldung gibt es zum Fehler?) und möglicherweise auch
- korrigiert (Wie lautet die korrigierte Form?) werden.

Während die ersten drei Stufen zunächst unberücksichtigt bleiben können, da es sich um Analyseschritte handelt, die meines Erachtens intern im System ablaufen, spielen die letzten drei Stufen bei der Betrachtung der didaktisch adäquaten Fehlerbehandlung in einem ICALL-System eine wesentliche Rolle, wie schon im vorhergenden Abschnitt 2.2 bei der Betrachtung des Begriffs „Korrektur“ angedeutet wurde. Dem Lerner kann auf der Basis einer Evaluierung beispielsweise mitgeteilt werden, dass in einem simulierten Dialog die aktuelle Lernereingabe nicht interpretiert werden konnte und daher die Eingabe vermutlich einen schwerwiegenden Fehler enthält. Die Erklärung eines vom System identifizierten Fehlers muss so gestaltet sein, dass sie auch für den Lerner verständlich ist und kann, wie in Abschnitt 3.3.2 erläutert wird, auch die morphosyntaktische Struktur der gesamten Eingabe umfassen, die den Lerner beim Erwerb der Sprachbewusstheit unterstützen kann. Schließlich kann möglicherweise dem Lerner eine Korrektur präsentiert

werden, wenn die Aufgabenstellung dieses nicht als Leistung des Lerners vorsieht.

Es kann in Bezug auf die Analysefähigkeiten eines ICALL-Systems auch die Frage gestellt werden, inwieweit auf die Muttersprache (L1) des Lerners bei der Identifikation und insbesondere bei der Erklärung von Fehlern Rücksicht genommen werden kann und muss, da sich wie erwähnt in der psycholinguistischen Forschung bestätigt, dass in bestimmten Bereich ein Einfluss der L1 bei der Produktion der L2 erkennbar ist, auch wenn dieser Aspekt in der Literatur zur Fehlerbehandlung im FU kaum Berücksichtigung findet. Wie in Abschnitt 5.1 näher erläutert wird, gibt es in der Literatur vor allem zu so genannten antizipationsbasierten[15] Systemen einige Vorschläge, nach denen die Fehlerregeln in Hinblick auf die L1 der Nutzer modelliert worden sind (Schwind, 1988; Krüger-Thielmann, 1992), da sich mit Hilfe der L1 eine Grundlage für die Gestaltung der Fehlerregeln finden lässt. Für antizipationsfreie Systeme muss im Gegensatz dazu gelten, dass Fehler nur ohne Einbezug der L1 identifiziert werden können. Menzel und Schröder (1998c, S. 288) schreiben dazu:„Therefore it remains unclear, how a truly model-based approach to the diagnosis of interference phenomena can be designed at all.“ Die Integration von zusätzlichen Regeln in die Grammatik, die eine L1-Konstruktion beschreiben und zur Fehleridentifikation und -erklärung herangezogen werden können, entspricht im Wesentlichen der Antizipation, wie sie in den antizipationsbasierten Systemen vorgenommen wird. In einem ICALL-System, das Fehler antizipationsfrei analysiert, können daher Fehler meines Erachtens nur mit Bezug auf die modellierte Sprache, das heißt die Fremdsprache identifizierbar sein. Die Berücksichtigung der L1 kann möglicherweise über die Integration einer Art Lernermodellierung in einem ICALL-System realisiert werden. Eine wesentliche Aufgabe solch eines Moduls besteht üblicherweise in der Registrierung der Fehlertypen, die von Lernern besonders häufig gemacht werden, und einer entsprechenden Adaptation des Systems. Wenn dann auf der Grundlage dieser Lernermodellierung beziehungsweise der darin enthaltenen Fehlerhäufigkeiten vom System bestimmte Fehlererklärungen präsentiert werden oder ein bestimmter Übungstyp empfohlen wird, sollten dabei ganz wesentlich auch Fehler berücksichtigt werden, die aufgrund des Transfers von der L1 zur L2 entstanden sind.

Die im vorhergehenden Unterkapitel angeführte differenzierte Behandlung von Fehlern in schriftsprachlichen Äußerungen auf der Basis der Lernerkenntnisse setzt in einem ICALL-System eine umfangreiche Lernermo-

[15]Unter Antizipation von Fehlern sollen hier Verfahren zusammengefasst werden, die entweder Informationen zu Fehlern in der Grammatik (so genannte Fehlerregeln) beziehungsweise im Lexikon oder in Meta-Regeln (als bekanntes Beispiel Weischedel und Sondheimer, 1983) kodieren. Antizipationsfreie Ansätze dagegen belassen die linguistischen Daten im System nahezu unverändert und versuchen, Fehler mit Hilfe des Analyseverfahrens, das heißt des Parsings zu diagnostizieren.

dellierung voraus. Da diese nicht implementiert worden ist und da zusätzlich verschiedene Verarbeitungsmodule auf das Vorhandensein der Wörter im Systemlexikon angewiesen sind, ist die Behandlung von orthografischen Fehlern pragmatisch gelöst worden. Im Fall einer Eingabe, die nicht im Systemlexikon enthalten ist, werden dem Lerner mögliche Korrekturkandidaten zur Auswahl präsentiert. Der Lerner bekommt so mitgeteilt, welches Wort seiner Eingabe nicht im System enthalten ist und welche möglichen Korrekturen ein bekanntes Wort ergeben würden. Ob der Lerner einen Vorschlag auswählt oder eine ganz andere Korrekturstrategie anwendet, bleibt ihm überlassen.

Hinzu kommt eine linguistisch-technische Einsicht. Ein abgeschlossenes Lexikon, das alle Wörter enhält, kann bekanntermaßen aufgrund der morphologischen Produktivität und anderer Faktoren wie zum Beispiel die Bildung von Komposita nicht generiert werden (cf. Baayen, 2001). Mit Hilfe der Angabe von möglichen Korrekturen wird der Lerner explizit darauf hingewiesen, welche Wörter sich im Lexikon befinden und welche deshalb überhaupt verarbeitet werden können. Die technische Realisierung wird in Abschnitt 6.2.1 erläutert. Eine Erkennung von Morphologiefehlern (zum Beispiel „gehte“ als fehlgeleitete Regelanwendung) mit Hilfe einer grundlegenden morphologischen Analyse wäre zwar gerade auch für das Deutsche wünschenswert, ist aber in Promis*D* nicht implementiert worden. Vermutlich ließe sich nur mit sehr großem Aufwand ein System konstruieren, dass Orthografie- beziehungsweise Morphologiefehler ohne die Interaktion mit dem Lerner korrigieren und so den simulierten Dialog fortführen kann.

2.3.3 Zusammenfassung

Es lässt sich festhalten, dass Computer zur Verwirklichung von modernen didaktischen Prinzipien eingesetzt werden können, sodass nicht nur das Sprachenlernen an sich, sondern sowohl die Sprachbewusstheit als auch die Lernbewusstheit gezielt gefördert werden können. Um diese Ziele zu erreichen, sind Lerntypen wie zum Beispiel aufgabenorientiertes und prozessorientiertes Lernen mit authentischen Materialien und Interaktionsformen notwendig. In erster Linie kommen dafür Konzepte in Frage, in denen der Computer eher als Explorationswerkzeug der Sprache dient, um einem Lerner die eigenständige Konstruktion von sprachlichem Wissen zu erlauben (siehe Ludewig, 2003). Allerdings muss auch ein gewisser Rahmen gegeben sein, in dem der Lerner beispielsweise durch das Feedback eines Lehrenden unterstützt wird. In Selbstlernsituationen lassen sich die sehr offenen, konstruktivistischen Ansätze noch weniger umsetzen, da hier eine noch stärkere Notwendigkeit der Instruktion besteht, die aber natürlich nicht das einzige Mittel sein darf.

Um didaktisch sinnvoll zu sein, muss die Behandlung von Fehlern im Unterricht in erster Linie der Interaktionsform im Unterricht angepasst werden. Wie sich in verschiedenen Untersuchungen außerdem gezeigt hat, kann ein

ausführliches Fehlerfeedback den Lerner beim Verständnis von fremdsprachlichen Strukturen und bei der Erfassung des eigenen Lernstandes deutlich unterstützen.

Damit sind die Anforderungen an ein didaktisch sinnvolles Computerprogramm zum Sprachenlernen definiert. Das Programm sollte möglichst authentische Interaktionsformen bieten, die ein inhaltsbezogenes Arbeiten ermöglichen, das heißt ein simulierter Dialog als Übungsform sollte so flexibel wie möglich gestaltet werden, um ein Maximum an kommunikativer Interaktion zuzulassen. Das linguistische Feedback an den Lerner muss dabei einerseits viel Information über die Lernereingabe bieten und andererseits so präsentiert werden, dass es der momentanen Aufgabenstellung und dem Wissensstand des Lerners angepasst ist. In der Dialogsimulation sollte also der Kommunikationsfluss einerseits nur dann unterbrochen werden, wenn die Fortführung des Dialogs nicht erfolgen kann (oder wenn die Programmstruktur dies erzwingt). Andererseits ist im vorliegenden Szenario die Chance gegeben, dass morphosyntaktische Fehler während der Dialogfortführung präzise identifiziert und gespeichert werden können, um für eine anschließende Fehlerarbeit eine größtmögliche Flexibilität zu gewährleisten.

Kapitel 3

Grammatikmodellierung

In diesem Kapitel wird die dem System zugrunde liegende Grammatiktheorie *Lexical Functional Grammar* (LFG) vorgestellt und erläutert, aus welchen Gründen die LFG für den Einsatz in einem ICALL-System geeignet ist und welche Eigenschaften der Theorie dabei die Funktionalität des Systems sinnvoll unterstützen. Die Anforderungen ergeben sich meines Erachtens einerseits aus dem System, das natürliche Sprache automatisch analysieren und mögliche Fehler in der Eingabe identifizieren soll, und andererseits aus den Ansprüchen des Lerners an eine verständliche Rückmeldung.

Zu Beginn des Kapitels sollen einige Anforderungen spezifiziert werden und in Abschnitt 3.2 erläutert werden, wie die Grundstrukturen der LFG aussehen. In Abschnitt 3.3 wird darauf aufbauend gezeigt, welche Eigenschaften im Kontext von ICALL besonders relevant sind, und wie die LFG die Anforderungen erfüllen kann. Zum Abschluss wird dargestellt, wie einige ausgewählte sprachliche Phänomene in Promis$D$ modelliert worden sind.

3.1 Anforderungen

Meines Erachtens lassen sich drei Anforderungen an eine Grammatiktheorie stellen, die in einem ICALL-System zur Anwendung kommen soll.

Die Grammatiktheorie sollte linguistisch adäquate Beschreibungen liefern können.
Obwohl diese Anforderung wohl von jeder Grammatiktheorie für sich in Anspruch genommen wird, kann meines Erachtens die „Adäquatheit“ oft aufgrund bestimmter Eigenschaften angenommen werden. Die Theorien basierend auf transformationellen Ansätzen zum Beispiel beanspruchen üblicherweise unter anderem eine gewisse Erklärungsadäquatheit in Bezug auf einen idealen Sprecher/Hörer und in Bezug auf die universalen Eigenschaften der natürlichen Sprachen unter dem Stichwort „Universal Grammar“. Dagegen nimmt die LFG für sich unter anderem in Anspruch, besonders in typologischer Hinsicht weitreichende Beschreibungen auf der Basis der strukturellen

Eigenschaften der Theorie liefern zu können. Ein anderer Gesichtspunkt ist die Modularität der Theorie. Um die Herkunft eines bestimmten linguistischen Phänomens erklären zu können, kann es sich anbieten, eine Theorie in Module beziehungsweise strukturelle Ebenen zu unterteilen und damit eine präzise Beschreibung zu erreichen. Auf den Aspekt der Adäquatheit wird zusammen mit der Beschreibung der LFG in Abschnitt 3.2 genauer eingegangen.

Die Grammatiktheorie sollte für eine Implementierung geeignet sein, das heißt, die formalen Eigenschaften der strukturellen Elemente der Theorie sollten bekannt sein.
Zur automatischen rechnergestützten Analyse werden oft Phrasenstrukturregeln (PS-Regeln) und daran annotierte Feature-Gleichungen zum Aufbau von Feature-Strukturen verwendet, da einerseits die formalen Eigenschaften bekannt sind und somit die Umsetzung in ein Programm mit relativ geringem Aufwand möglich ist und andererseits die linguistischen Beschreibungsmöglichkeiten als hinreichend adäquat gesehen werden. Im Rahmen der LFG wurde auf ersteren Aspekt von vornherein Wert gelegt (Dalrymple et al., 1995), sodass als Konsequenz vielfältige Implementierungen der LFG vorliegen (siehe Abschnitt 3.3.1).

Die Grammatik sollte dem Grammatikverständnis eines Lerners angepasste Erläuterungen liefern können.
Für den Einsatz einer Grammatiktheorie in einem ICALL-System muss dieser Bereich als ein wesentlicher Aspekt gesehen werden. Um eine Rückmeldung für einen Lerner zu generieren, muss das Ergebnis der Analyse so gestaltet sein, dass sich daraus ein für den Lerner verständliches Feedback ableiten lässt. Zu unterscheiden sind hier

- das Feedback zu einem vom Parser identifizierten Fehler und
- die Präsentation einer Analyse als morphosyntaktische Beschreibung eines Satzes.

Für Ersteres eignen sich mit mehr oder weniger Aufwand für der Generierung des Feedbacks vermutlich unterschiedliche Grammatiktheorien, da sich grundlegende Konzepte wie Kongruenz und Rektion sowie funktionale Elemente in (fast) jedem Fall identifizieren lassen. Für die zweite Aufgabe allerdings sollte eine enge Beziehung zwischen der Grammatiktheorie und der zur „Veranschaulichung“ gewählten Konzepte in so genannten Lernergrammatiken bestehen, wenn ein Lerner die Analyse verstehen und daraus Schlussfolgerungen ziehen können soll. Eine Lernergrammatik bzw. eine pädagogische Grammatik (Storch, 1999) hat im Gegensatz zu einer computationellen Grammatik oder einer Grammatiktheorie die Aufgabe, eine für den Lerner verständliche Beschreibung einer Sprache zu liefern, die außerdem unter Umständen in der Lage ist, den Fremdsprachenlernprozess zu unterstützen. Zu nennen sind zum Beispiel die *Deutsche Grammatik* von Helbig und Buscha (1996) oder das *Lehr- und Übungsbuch der deutschen Gram-*

matik von Dreyer und Schmitt (2000), die diese Ziele verfolgen. Der Lerner sollte die Möglichkeit haben, die entscheidenden morphosyntaktischen Eigenschaften einer Sprache nachzuschlagen, um so Hilfsmittel zur Bewältigung von Aufgaben an der Hand zu haben.[1]

Insbesondere auf den Aspekt der Präsentation einer Analyse soll in Abschnitt 3.3.2 eingegangen werden. Dabei sollte erstens die funktionale Perspektive der LFG im Vordergrund stehen, die sich meines Erachtens auch in verschiedenen Lernergrammatiken wiederfindet. Zweitens ist eine relativ starke lexikalische Perspektive, die Lexeme und die mit ihnen verknüpften Informationen in den Mittelpunkt stellt, ein weiterer Bereich, der sich bei der Analyse von Lernergrammatiken als hervorhebenswert gezeigt hat. Offensichtlich können aber bestimmte „Prinzipien“ in einem allgemeinen Sinne, die die Eigenschaften der verwendeten strukturellen Elemente einer Theorie beschreiben, nicht zur Erklärung gegenüber dem Lerner herangezogen werden. Beispielsweise ist die im P&P-Ansatz (Principles & Parameters; Chomsky, 1981) vorgenommene Trennung von Verb-Suffix als Träger der Wortforminformationen und Stamm in der D-Struktur und die anschließende Zusammenführung durch die Ableitungskette in der S-Struktur für den typischen Sprachenlerner wohl kaum einsichtig.

Aufbauend auf diesen zum Teil allgemein computerlinguistischen und zum Teil didaktischen Anforderungen soll im Folgenden gezeigt werden, dass die LFG eine geeignete Kandidatin einer Grammatiktheorie zum Einsatz in einem ICALL-System darstellt. Damit sind andere Grammatiktheorien nicht ausgeschlossen, aber bestimmte Aspekte der Theorie zeichnen sie in Bezug auf den computerlinguistischen Einsatz und die Erläuterungsmöglichkeiten zu sprachlichen Phänomenen aus. Im Anschluss an die allgemeine Darstellung der LFG folgt eine erweiterte Diskussion dieses Aspektes.

3.2 Grundstrukturen der LFG-Theorie

In den letzten Jahren sind verschiedene Einführungen zur LFG erschienen, wie zum Beispiel Bresnan (2001), Falk (2001) und Kroeger (2004), woran sich zeigt, dass das allgemeine Interesse an der LFG in der Sprachwissenschaft zugenommen hat. Dieses liegt unter anderem an neueren Erweiterungen der LFG wie zum Beispiel der Kombination von LFG und Optimality Theory (Frank et al., 2001), LFG und der so genannten GLUE-Semantik (Dalrymple, 1999) und den Ansätzen zu stochastischen Parsingverfahren auf Grundlage von LFG (DOP-LFG; Bod und Kaplan, 1998; Bod et al., 2003).

Die LFG versucht, sprachliche Phänomene aus einer lexikalischen und

[1] Dabei gehe ich davon aus, dass die Ausarbeitungen in den Lernergrammatiken tatsächlich der inhaltlichen Vermittlung im Unterricht entsprechen. Die Diskussion um den Stellenwert des Grammatikunterrichts (zum Beispiel vs. kommunikativen Unterricht) soll hier nicht geführt werden.

funktionalen Perspektive zu beschreiben. Dieser Ansatz unterscheidet sich insbesondere vom P&P-Ansatz, in dem Sprachbeschreibungen im Wesentlichen auf Phrasenstruktur-Konfigurationen basieren (Chomsky, 1981). Gegen die Annahme dieses Konzeptes sprechen verschiedene Argumente. Zwei Beispiele seien hier angeführt (nach Austin, 2001).
Im Englischen erlaubt das Verb „talk about“ als Komplement normalerweise nur eine Nominalphrase und kein satzartiges Komplement.

(3.1) a. We talked about [the fact that he was unhappy $_{NP}$] for weeks.
b. *We talked about [that he was unhappy $_{CP}$] for weeks.

Wenn allerdings das satzartige Komplement in eine Topikposition gebracht wird, ist der Satz grammatisch.

(3.2) [That he was unhappy $_{CP}$] we talked about for weeks.

Diese Asymmetrie kann in der LFG mit Hilfe einer funktionalen Unterscheidung zwischen Topik und Objekt erklärt werden.
Als zweites, oft zitiertes Beispiel seien so genannte nicht konfigurationale Sprachen, wie zum Beispiel Malayalam (Drawidisch, Indien) und Warlpiri (Pama-Nyungisch, Australien) angeführt. Die Wortstellung ist hier sehr frei und eine Asymmetrie in der Konstituenz lässt sich nicht ohne weiteres erkennen. Das folgende Beispiel aus Falk (2001, S. 19, nach Mohanan (1982)) zu Malayalam[2] verdeutlich dies.

(3.3) a. Kutti aanaye kantu.
Kind.NOM Elefant.ACC sah
Das Kind sah den Elefanten.
b. Aanaye kutti kantu.
c. Aanaye kantu kutti.
d. Kantu aanaye kutti.
e. Kantu kutti aanaye.
f. Kutti kantu aanaye.

Hieraus ergeben sich Hinweise für seperate Strukturen von Konstituenz und Funktion, die nicht in transformationellen Grammatiken realisiert werden.

Bei der LFG handelt es sich um eine „parallele Constraint-basierte Syntaxtheorie“. Parallel heißt, dass verschiedene Strukturen zur Beschreibung von Sätzen parallel aufgebaut werden, zwischen denen eine Abbildung (Mapping) definiert wird. Die Modularität ist erstrebenswert, wenn damit eine Reihe von Phänomenen umfassend erklärt werden können (Falk, 2001). Dieser Aspekt unterscheidet LFG ganz wesentlich von anderen Grammatiktheorien, die nur auf einer einzigen Strukturform basieren. Constraint-basiert ist

[2]Einige Diakritika werden hier nicht dargestellt.

die Theorie deshalb, weil sich die Wohlgeformtheit der Analyseergebnisse nicht aus einer mehr oder weniger langen Ableitungskette ergibt, sondern durch statische Constraints festgelegt wird. Schließlich handelt es sich bei den Verfahren zum Aufbau der Strukturen um monotone Verfahren, in denen ausschließlich Informationen hinzugefügt werden und die Strukturen nicht im Sinne einer Transformation verändert werden. Dieser Aspekt wiederum trägt ganz wesentlich zur Implementierbarkeit der Theorie bei.[3]

K-Struktur

Eine wesentliche Art von Struktur ist die Konstituenten-Struktur (K-Struktur), die auf kontextfreien Phrasenstrukturbäumen (PS-Bäumen) beruht. Üblicherweise wird ein modifizierter $\overline{X}$-Ansatz gewählt, um die Konstituenten zu organisieren. Im Unterschied zu anderen Theorien gilt aber das „lexical integrity principle“, das nur ganze Wörter als terminale Elemente zulässt. Bewegungen von Konstituenten sind wie erwähnt nicht vorgesehen. Daraus ergeben sich zum Teil unterschiedliche Verwendungen der aus der $\overline{X}$-Theorie bekannten Kategorien. Das folgende Beispiel einer „unannotierten“ Regel verdeutlicht, dass auch Elemente regulärer Ausdrücke verwendet werden können.

(3.4) S → NP VP PP*

Ein Satz besteht demnach aus einer NP, einer VP und keiner oder beliebig vielen PPn, wobei das Sternchen den so genannten Kleene-Star darstellt, der die Anzahl der möglichen PPn bestimmt. Zusätzlich werden meist weitere Markierungen angenommen, wie zum Beispiel runde Klammern „( )“, die Optionalität andeuten sollen, und das Plus-Zeichen „$^{+}$“, das aussagt, dass eine Kategorie mindestens einmal auftreten muss.

F-Struktur

Parallel zu einem Phrasenstrukturbaum wird die F-Struktur aufgebaut, wobei „F“ (zufällig) entweder für „Funktionale“ oder für „Feature“ stehen kann.

[3] Die Eignung der LFG als Modell zu Sprachbeschreibung lässt sich meines Erachtens auch aus der Verwendung in weiteren Kontexten herleiten. Pienemann (1998) zum Beispiel verwendet die LFG als Modell zur Beschreibung seiner „Processability Theory“ im Kontext einer Theorie zum Zweitspracherwerb und Schwarze (1995) benutzt die LFG in einer deskriptiven Grammatik des Italienischen. Levelt (1989, S. 161f) führt drei Argumente an, warum er die LFG als Grundlage für seine Theorie bevorzugt. Erstens handelt es sich um eine explizite Theorie, die explizite Formulierung von Prozeduren zur direkten Generierung von Sätzen, wie sie sich an der Oberfläche darstellen. Zweitens ist die LFG lexikonbasiert, was sich gut mit dem Konzept der Lemmata als Basis zur Generierung verbinden lässt. Drittens schließlich bietet sich die Theorie an, da nicht nur diese, sondern auch weitere psycholinguistische Theorien von einer zweigeteilten Repräsentation mit Konstituenten- und Funktionsstrukturen ausgehen. Allerdings gilt natürlich auch für Levelt (S. 162):„It should be kept in mind, however, that no grammatical theory can claim to be the correct one.“

F-Strukturen bestehen aus komplexen Attribut-Wert-Matrizen[4] und können morphosyntaktische, semantische und diskursive Informationen sowie Subkategorisierungsangaben enthalten. Durch die Strukturierung mit Hilfe von komplexen Werten, die vor allem Werte von grammatischen Funktionen (SUBJ, COMP etc.) enthalten, werden verschiedene Schichten erreicht. Das folgende Beispiel zeigt eine sehr kleine und vereinfachte, aber korrekte F-Struktur für den Satz „Ich lache“.

(3.5)
$$\begin{bmatrix} \text{PRED} & = \text{`LACHEN <SUBJ>'} \\ \text{TENSE} & = \text{pres} \\ \text{SUBJ} & = \begin{bmatrix} \text{PRED} & = \text{`ICH'} \\ \text{PERS} & = 1 \end{bmatrix} \end{bmatrix}$$

Im ersten PRED-Feature dieser Struktur befindet sich einerseits eine semantische Repräsentation des Prädikats (LACHEN) und andererseits eine Liste mit den Argumentstellen des Verbs (SUBJ), das hier also nur ein Subjekt fordert. Das Feature SUBJ mit einem komplexen Wert korrespondiert mit dem entsprechenden Element der Subkategorisierungsliste und enhält die Informationen zum Subjekt, in diesem Fall das PRED-Feature zum Subjekt und die Angabe zur Person. Die Kongruenzinformation [PERS = 1] befindet sich insbesondere nicht in der „äußeren“ F-Struktur. Die Argumente innerhalb der spitzen Klammern deuten an, dass es sich um thematische Argumente handelt. Nicht thematische Argumente wie zum Beispiel expletive Subjekte werden außerhalb der Klammer an die Liste notiert.

(3.6)
$$\begin{bmatrix} \text{PRED} & = \text{`REGNEN} < \ > \text{SUBJ'} \end{bmatrix}$$

Dieser Fall tritt auch in so genannten Raising-Konstruktionen auf, in denen ein Element mehrere grammatische Funktionen zum einen für das Matrixverb und zum anderen für das eingebettete Verb übernimmt, wie das folgende Beispiel (3.7) zeigt. *dir* kann sowohl als das Objekt zu *befehlen* als auch das Subjekt zu *lachen* interpretiert werden.

(3.7) Ich befehle dir zu lachen.

Grammatische Funktionen lassen sich in Argument- und Non-Argument-Funktionen unterscheiden. Argument-Funktionen zeichnen sich dadurch aus, dass sie in der Subkategorisierungsliste eines Verbs auftreten können (zum Beispiel SUBJ), während es sich bei Non-Argument-Funktionen um Funktionen handelt, die nicht durch ein anderes Element gefordert werden (zum Beispiel ADJUNCT). Um nicht beliebige F-Strukturen zuzulassen, unterliegen die Strukturen bestimmten Wohlgeformtheitsbedingungen, das heißt, die Informationen in einer F-Struktur müssen eindeutig, vollständig und kohärent sein.

[4] Attribut und Feature verwende ich synonym.

- Eindeutigkeit: Jedes Attribut darf nur einen Wert haben.
- Vollständigkeit: Eine F-Struktur ist vollständig, gdw. alle vom Prädikat regierten Argument-Funktionen als Attribute vorhanden sind.
- Kohärenz: Eine F-Struktur ist kohärent, gdw. alle als Attribute vorhandenen grammatischen Argument-Funktionen von einem Prädikat regiert werden.

Die folgende F-Struktur für den Satz „Ich lache dich“ ist beispielsweise nicht kohärent, da es für das OBJ-Feature kein Argument im PRED-Feature, das heißt in der Subkategorisierungsliste gibt.

(3.8)
$$\begin{bmatrix} \text{PRED} & = \text{‘LACHEN <SUBJ>’} \\ \text{TENSE} & = \text{pres} \\ \text{SUBJ} & = \begin{bmatrix} \text{PRED} & = \text{‘ICH’} \\ \text{PERS} & = 1 \end{bmatrix} \\ \boxed{\text{OBJ}} & = \begin{bmatrix} \text{PRED} & = \text{‘DICH’} \\ \text{PERS} & = 2 \end{bmatrix} \end{bmatrix}$$

Zum Teil können diese Bedingungen erst nach dem Ende des Aufbaus der Strukturen abgetestet werden, da ja die Informationen nach und nach zusammengefügt (unifiziert) werden. Die Vollständigkeit lässt sich beispielsweise erst feststellen, wenn der Aufbau der Strukturen abgeschlossen ist, während die Eindeutigkeit und Kohärenz zu jeder Zeit gültig sind.

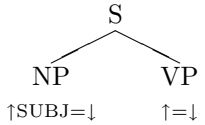
Konstruktion einer F-Struktur

Um den Aufbau der komplexen F-Strukturen zu gewährleisten, werden die Knoten in den Phrasenstrukturbäumen mit Annotationen versehen. Diese funktionalen Gleichungen sind so zu verstehen, dass sie Informationen von dem aktuellen Knoten an den Mutterknoten weiterreichen. Im folgenden Beispiel sind zwei funktionale Gleichungen enthalten.

(3.9)
$$\begin{array}{lcccc} \text{S} & \rightarrow & \text{NP} & & \text{VP} \\ & & \uparrow\text{SUBJ}=\downarrow & & \uparrow=\downarrow \end{array}$$

Die erste Gleichung ↑SUBJ=↓ setzt die Information in der F-Struktur des Knotens NP mit der Information des SUBJ-Features im Mutterknoten S gleich. Die zweite Gleichung ↑=↓ reicht die von unten kommenden Informationen einfach weiter nach oben. Damit wird üblicherweise zugleich der Kopf der Phrase in der funktionalen Sichtweise festgelegt. Deutlicher wird dieser Mechanismus, wenn die Regel als ein Baum dargestellt wird.

(3.10)



Die Annotation an dem VP-Knoten ↑=↓ bedeutet also, dass die F-Struktur des VP-Knotens, im Wesentlichen also die mit dem Verb verknüpfte F-Struktur aus dem Lexikon, mit der F-Struktur des S-Knotens identifiziert wird. Die F-Struktur der NP wird als Wert des SUBJ-Features eingefügt und diese komplexe Feature-Struktur wird dann mit der F-Struktur des Verbs zusammengefügt beziehungsweise unifiziert. Wie der Lexikoneintrag zum Verb *mag* in Beispiel (3.11) zeigt, spezifiziert der Eintrag die Kongruenz- und Rektionsbedingungen für einen bestimmten Verbrahmen.

Die Verbindung zwischen den einzelnen Strukturen basiert jeweils auf einer Abbildungsrelation, die einfach definiert werden kann. Die Abbildung von K- auf F-Strukturen geschieht mit Hilfe der oben angeführten Annotationen an den PS-Regeln. Für jeden Knoten in einem Baum existiert eine so genannte F-Description, das heißt eine Menge von funktionalen Gleichungen. Diese werden den Annotationen entsprechend unifiziert.

Eine Beispielanalyse (K- und F-Struktur) des Satzes „Der Mann mag Milch“ wird in der Abbildung (3.1) unten dargestellt. Die Variablen f_1, f_2 und f_3 und die Pfeile deuten an, welche Bereiche der K-Struktur welchen Bereichen der F-Struktur zugeordnet sind. Dem Knoten f_2:NP ist zum Beispiel die Teil-F-Struktur zugeordnet, die die Informationen über das Subjekt enthält. Mit Hilfe der Variablen und so genannter funktionaler Gleichungen lässt sich ein formales Verfahren zur Herleitung der F-Strukturen mit Hilfe der Informationen aus den lexikalischen Einträgen und den annotierten PS-Regeln angeben.

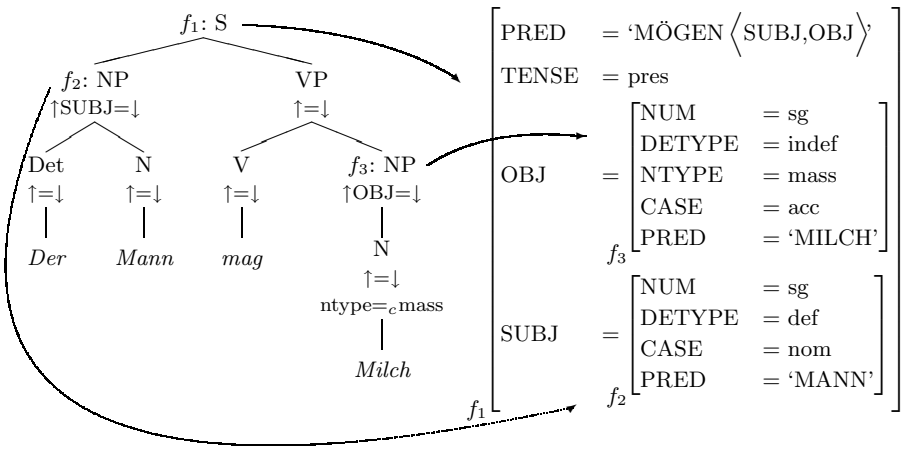


Abbildung 3.1: Mapping von der K- auf die F-Struktur

Eine funktionale Gleichung wäre in diesem Fall (f_1 SUBJ)=f_2, aber auch (f_3 PRED)='MILCH', das heißt, jeder Pfeil wird zur Generierung der Gleichungen durch eine Variable ersetzt. Für jeden Knoten im Baum existiert dann eine Variable, die mit einer Teil-F-Struktur korrespondiert. Die Menge aller funktionalen Gleichungen ermöglicht die Konstruktion einer F-

Struktur. Dieses Mapping wird auch unter dem Namen „ϕ-projection“ als Funktion dargestellt.

Lexikoneinträge stellen in erster Linie die notwendigen Informationen zum Aufbau einer F-Struktur zur Verfügung. Die folgenden zwei Beispiele verdeutlichen den generellen Aufbau.

(3.11) Mann, N,
$$\begin{bmatrix} \text{NUM} & = \text{sg} \\ \text{CASE} & = \text{nom} \\ \text{PRED} & = \text{‘MANN’} \end{bmatrix}$$

mag, V,
$$\begin{bmatrix} \text{PRED} & = \text{‘MÖGEN} < \uparrow \text{SUBJ}, \uparrow \text{OBJ} > \text{’} \\ \text{TENSE} & = \text{pres} \\ \text{OBJ} & = \begin{bmatrix} \text{CASE} & = \text{acc} \end{bmatrix} \\ \text{SUBJ} & = \begin{bmatrix} \text{CASE} & = \text{nom} \\ \text{PERS} & = 1 \vee 3 \\ \text{NUM} & = \text{sg} \end{bmatrix} \end{bmatrix}$$

Während der Eintrag zu *Mann* nur die Eigenschaften dieses Wortes festlegt, enthält der Eintrag von *mag* auch Kongruenz und Rektionsinformationen. Es wird festgelegt, dass das Subjekt Singular Nominativ sein muss und das Objekt den Kasus Akkusativ haben muss. Andernfalls kann mit Hilfe der Unifikation keine widerspruchsfreie F-Struktur erzeugt werden. Einige zusätzliche Erweiterungen in diesem Bereich, wie zum Beispiel „Existential Constraints“ und „Functional Uncertainty“, die üblicherweise in der LFG angenommen werden, sind in Abschnitt 3.4 (Seite 70ff) dargestellt.

Typologisch gesehen dienen die unterschiedlichen Strukturen, das heißt die K- und die F-Struktur zur Darstellung der Unterschiede in verschiedenen Sprachen. Es wird in der LFG angenommen, dass die F-Struktur zur funktionalen und inhaltlichen Beschreibung eines Satzes sprachübergreifend im Wesentlichen gleich bleibt. Die K-Struktur ist die Ebene, in der Anpassungen an einzelne Sprachen vorgenommen werden sollten (cf. Butt et al., 1999). Für nicht konfigurationale Sprachen ergibt sich allerdings eine wesentliche Schwierigkeit in der Verknüpfung von grammatischen Funktionen und Phrasenstrukturelementen, da die Funktion eines Wortes beziehungsweise einer Phrase nicht durch die Position, sondern zumeist durch die morphologischen Eigenschaften bestimmt wird. Es wurde daher ein Konzept entwickelt, die Information zu grammatischen Funktionen aus der Morphologie, das heißt aus einem Lexikoneintrag und nicht aus den Annotationen an den PS-Regeln zu ziehen und damit den Aufbau einer F-Struktur zu ermöglichen (Nordlinger, 1998).

K- und F-Strukturen sind die beiden wichtigsten Elemente in der LFG, die in dem hier vorgestellten Programm verwendet werden. Zusätzlich werden zur Vervollständigung der Theorie aber noch andere Bereiche wie zum Beispiel die A(rgument)-Struktur angenommen. Über ein Mapping von thematischen Rollen in der Argumentliste eines Verbs auf grammatische Funk-

tionen in der Subkategorisierungsliste werden die lexikalischen Einträge generiert, die für den Aufbau einer F-Struktur notwendig sind. Mit Hilfe der Ebene der A-Struktur lassen sich bestimmte syntaktische Phänomene wie zum Beispiel die Passivbildung erklären, die nicht oder nur unvollständig auf der Basis von K- und F-Struktur sowie einfachen, lexikalischen Regeln beschrieben werden können, indem das Mapping von thematischen Rollen auf grammatische Funktionen entsprechend geändert wird. Die Einführung einer solchen zusätzlichen Struktur geschieht insbesondere in Hinsicht auf eine Generalisierung und Modularisierung der Theorie. In verschiedenen Publikationen wurden diverse zusätzliche Ebenen vorgeschlagen, um weitere Bereiche der linguistischen Beschreibung zu eröffnen. Erwähnt seien hier σ-Strukturen für die Semantik und m-Strukturen für die Morphologie. Auf die Ansätze der LFG zur Beschreibung von speziellen syntaktischen Phänomenen soll hier nicht weiter eingegangen werden. In den Abschnitten 3.3.2 sowie 3.4 folgen dazu einige weitere Erläuterungen insbesondere zu der für Promis$D$ entwickelten Grammatik.

Nachdem nun die grundlegenden Elemente der Theorie der LFG beschrieben worden sind, folgt im nächsten Abschnitt zunächst eine kurze Darstellung der Anforderungen an eine Theorie, die in einem ICALL-System für die Analyse der Lernereingaben genutzt werden soll. Anschließend wird für die Eignung der LFG als pädagogische Grammatik plädiert und zum Abschluss des Kapitels auf die in Promis$D$ modellierten Phänomene eingegangen.

3.3 LFG im ICALL-Kontext

Um die Eignung der LFG in Hinblick auf die Anwendung in einem ICALL-System zu analysieren, ergeben sich meines Erachtens zwei Perspektiven. Einerseits kann die LFG aus technisch-theoretischer Sicht betrachtet werden, um damit den praktischen Einsatz in einem solchen System zu beleuchten. Andererseits sollte aus didaktischer Perspektive geklärt werden, inwieweit die LFG zur Erläuterung von syntaktischen Phänomenen für den Sprachenlerner (auch gegenüber anderen Theorien) von Vorteil ist. Dafür ist vor allem der Vergleich mit Lernergrammatiken relevant, wenn angenommen werden kann, dass einem Lerner die darin enthaltenen Erläuterungen zu morphosyntaktischen Sachverhalten verständlich sind.[5]

3.3.1 LFG zur rechnergestützen Analyse natürlicher Sprache

Die LFG zeichnet sich unter anderem dadurch aus, dass die zur Sprachbeschreibung verwendeten Grundstrukturen formal fundiert sind (siehe Dalrymple et al., 1995). Sowohl annotierte PS-Regeln als auch Feature-Strukturen sind schon längere Zeit (zum Beispiel Karttunen, 1984) wohldefinierte

[5]Eine erste Ausarbeitung der folgenden Konzepte findet sich in Reuer (2003).

Mittel, die als formal abgesichert angesehen werden können. Auch die in der LFG-Theorie eingesetzten Erweiterungen wie zum Beispiel „Functional Uncertainty“ oder Adjunkt-Listen als Werte in Feature-Strukturen sind innerhalb eines formalen Ansatzes entwickelt worden (siehe Seite 73). Insbesondere die auf dem P&P-Konzept (Chomsky, 1981) basierenden transformationellen Theorien sind in dieser Hinsicht anders strukturiert. Es hat hier zwar auch Ansätze zur Formalisierung und Implementation gegeben (Campone und Wehrli, 1996), aber diese formalen Ansätze sind nicht Grundlage der Theorie, sondern wurden im Nachhinein mit der Theorie verknüpft.

Aus dieser formalen Orientierung heraus ergibt sich, dass die Theorie relativ leicht implementiert werden kann (zum Beispiel Eisele und Dörre, 1986; Rohrer und Schwarze, 1988). Das bekannteste Beispiel für eine Implementation eines LFG-basierten Parsers ist die so genannte „Grammar Writer's Workbench“, die in den Labors der Firma Xerox entwickelt worden ist (Kaplan und Maxwell, 1996). Anschließend wurde der Parser auch im „Xerox Linguistics Environment“[6] eingesetzt. Als weitere größere Implementationen muss unter anderem auch das System XLFG von Clément und Kinyon (2001)[7] angeführt werden. Schließlich demonstriert auch der relativ häufige Einsatz in Systemen zur maschinellen Übersetzung (von Kaplan et al. (1989) bis Way (2003)) die Eignung von LFG zur rechnergestützen Analyse natürlicher Sprache.

Abschließend soll auf die praktische Modellierung größerer computerlinguistischer Grammatiken eingegangen werden. Zu den neueren umfassenden Grammatikimplemetierungen gehören Langer (2001, PS-Regeln mit flachen Featurestrukturen, also LFG-”ahnlich), Müller (1999, HPSG) sowie Butt et al. (1999, ParGram-Projekt), in der LFG als Grammatiktheorie gewählt wurde. Im Rahmen dieses letzten Projektes wurde nicht nur eine Grammatik für das Deutsche entwickelt, sondern es wurden in einem parallelen Ansatz außerdem Grammatiken unter anderem für das Englische und Französische sowie für strukturell ganz andere Sprachen wie Urdu und Japanisch entworfen. Hier hat sich gezeigt, dass das Konzept der LFG zur Trennung von K-Strukturen zur Beschreibung der sprachspezifischen Eigenheiten sowie F-Strukturen zur sprachübergreifenden Beschreibung auch tatsächlich realisiert werden kann. Angaben über die Abdeckung der LFG-Grammatik liegen leider nicht vor.

Vor diesem Hintergrund kann LFG als eine Grammatiktheorie angesehen werden, die die Anforderungen eines sprachverarbeitenden Systems erfüllt. Die Theorie ist formal fundiert, was eine relativ einfache und effiziente Implementierung erlaubt und hat sich bei der Modellierung größerer Grammatiken auch im Bereich der maschinellen Übersetzung bewährt. LFG stellt sich damit als gut geeignet dar, auch in einem dialogorientierten ICALL-System

[6] Stand 2.2005 – http://www2.parc.com/istl/groups/nltt/xle/
[7] Stand 2.2005 – http://www.lionel-clement.net/xlfg/

als computerlinguistische Grundlage für die Analyse von Lernereingaben verwendet zu werden.

3.3.2 LFG als Lernergrammatik

Eine Lernergrammatik soll hier nicht als ein Grammatikfragment verstanden werden, das die Sprachkenntnisse eines Lerners beschreibt, sondern als eine Grammatik, die es einem Sprachenlerner ermöglicht, grammatische Erkenntnisse über eine Sprache zu gewinnen. In gedruckter Form existieren eine Reihe von (Lerner-)Grammatiken, die zum Teil speziell für bestimmte Lernstufen konzipiert worden sind (Dreyer und Schmitt, 2000; Reimann, 1996; Rug und Tomaszewski, 1993).
Obwohl es in dieser Arbeit in erster Linie um die Identifizierung von Fehlern in Lernereingaben und die daran anschließende Rückmeldung geht, soll hier eine erweiterte Darstellung erfolgen. Es soll ergründet werden, inwieweit sich LFG nicht nur zur Fehleranalyse im ICALL-Kontext eignet, sondern auch, ob die morphosyntaktische Beschreibung einer LFG-artigen Analyse dem Lerner zur Veranschaulichung präsentiert werden kann. Notwendig ist dafür insbesondere ein Vergleich mit Lernergrammatiken, die morphosyntaktische Sachverhalte in einer dem Unterrichtskonzept angepassten Art und Weise präsentieren.

Abgrenzung von anderen Grammatiktheorien

Neben LFG sind auch andere Grammatiktheorien bzw. Parser basierend auf anderen Theorien in ICALL-Systemen zum Einsatz gekommen, die auf Grund der linguistischen Analyse Rückmeldungen an einen Lerner geben können sollen (cf. Heift, 1998; Vandeventer, 2001). Dabei handelt es sich um die HPSG (Pollard und Sag, 1987, 1994) und den P&P-Ansatz (Chomsky, 1981) respektive. In diesem Abschnitt soll daher dieser Aspekt und die möglichen Konsequenzen beleuchtet werden. Hier geht es im Gegensatz zum vorhergehenden Abschnitt nicht in erster Linie um die formalen Grundlagen oder die Implementierbarkeit, sondern um die Nutzung der Theorie zur Erläuterung von grammatischen Strukturen und Eigenschaften und um die Frage, ob sich aus der Verwendung unterschiedlicher Theorien relevante Unterschiede für das Feedback ergeben.

Zur Rückmeldung eines Fehlers sind unter anderem Informationen zu grammatischen Kategorien und Belegungen wie Kasus, Genus etc., zu grammatischen Funktionen wie Subjekt, Objekt etc., Informationen zur Wortstellung, zur Fehlerposition wie zum Beispiel „Kongruenzfehler Subjekt – Verb“ oder „Konstituentenfehler an Position 3“ etc. und schließlich zu Fehlertypen wie „Kasusfehler Nom – Akk“ oder „Fehlender Artikel“ notwendig, die sich als Fehlererklärung zusammenfassen lassen. All diese Informationen müssen in erster Linie mit tatsächlich realisierten lexikalischen Einheiten verknüpft

sein, das heißt, dass Erklärungen nicht auf der Basis von so genannten Spuren oder leeren Knoten gegeben werden dürfen, da diese Konzepte für einen Lerner meistens unbekannt sein dürften. Es ist klar, dass diese Informationen in unterschiedlichen Grammatiktheorien auch in unterschiedlichen Strukturtypen enthalten und dass daher Fehlermeldungen zu einer Lernereingabe auf den Strukturen sehr unterschiedlicher Grammatiktheorien basieren können. Aus diesem Grund ist im Allgemeinen also zu erwarten, dass für eine konkrete Fehlermeldung die strukturelle Grundlage eher unerheblich ist, wenn die angeführten Informationen zur Generierung einer Meldung zur Verfügung stehen. Es ist allerdings auch denkbar, dass die Analyseergebnisse dem Lerner wie zum Beispiel in Rypa und Feuerman (1995) mehr oder weniger direkt präsentiert werden; in diesem Fall spielt aber die Wahl der Grammatiktheorie eine entscheidende Rolle.

Während in HPSG als Ziel eine extreme Verlagerung der Informationen auf das Lexikon verfolgt wird, erfolgt im Gegensatz dazu im P&P-Ansatz eine Konzentration auf die Konstituentenstruktur. Insbesondere in letzterer Theorie wird dabei versucht, viele syntaktische Zusammenhänge wie zum Beispiel die Bestimmung von grammatischen Funktionen oder so genannte Kontrollphänomene auf der Basis der Konstituentenstruktur zu beschreiben. Ziel ist es also, eine Theorie zu erhalten, die in einer möglichst uniformen Art morphosyntaktische Phänomene beschreibt, um die Erkenntnis umzusetzen, dass einerseits eine Grundlage für die universelle sprachliche Kompetenz beschrieben werden soll und dieses andererseits mit möglichst „ökonomischen“ Mitteln geschehen soll. Dieser letzte Aspekt wird allerdings meines Wissens in keiner Lernergrammatik umgesetzt. In Letzterer ist vor allem die Beachtung der Oberflächenstrukturen relevant, die mit Hilfe von möglichst reduzierten und intuitiven „Regeln“ beschrieben werden müssen, die ihrerseits so wenig Struktur wie möglich berücksichtigen sollten. Neben dieser Reduktion der Struktur in Lernergrammatiken wird außerdem eine sehr „lokale“ Sichtweise angestrebt, die nicht so sehr die Struktur eines ganzen Satzes in den Vordergrund stellt, sondern sich eher auf die Zusammenhänge innerhalb von Phrasen konzentriert.

Ein weiteres Element, das in Lernergrammatiken nicht vertreten ist, ist das $\overline{X}$-Schema, das die Phrasenstruktur sowohl in HPSG als auch im P&P bestimmt. Das strenge Schema wird dazu verwendet, den generellen Aufbau der Phrasenstruktur in allgemeiner und systematischer Weise zu restringieren. Diese grundsätzliche Einhaltung eines theoretischen Konzeptes lässt sich meines Erachtens einem Lerner einer konkreten Sprache nur schwer vermitteln. Dazu gehört auch eine Terminologie wie die Kategoriennamen *CP/IP* einer speziellen Theorie (Chomsky, 1981), die zwar linguistisch motiviert ist, für einen Lerner jedoch Verwirrung stiften kann. Zu diesem Bereich muss auch das Konzept der strengen *Head-Daughter-Konfiguration* gezählt werden, das im Rahmen der HPSG verfolgt wird („Head-Feature-Principle“).

Ein weiteres Konzept der GB, das sich nicht mit Konzepten aus Lerner-

grammatiken verknüpfen lässt, sind die so genannten *S-* und *D-Strukturen*[8] aus einer transformationellen Grammatiktheorie, die unterschiedliche Informationen über einen Satz zur Verfügung stellen. Üblicherweise wird im Fremdsprachenunterricht nur von der Oberflächenstruktur als einziger Ebene ausgegangen. Auch die zur Generierung der S-Struktur notwendigen Transformationen, die durch Bewegungsregeln wie zum Beispiel *move-α* gesteuert werden, sind meines Erachtens einem Lerner nicht zugänglich, auch wenn in einzelnen Grammatiken auf Bewegungsmetaphern zurückgegriffen wird, um zum Beispiel topikalisierte Strukturen zu erläutern.

Wie schon oben angeführt, bedeutet diese Kritik an den Theorien HPSG und P&P allerdings nicht, dass sie grundsätzlich ungeeignet wären. Die Implementationen aus Heift (1998) und Vandeventer (2001) demonstrieren offensichtlich das Gegenteil. Als Beispiel für eine geeignete lernerorientierte Eigenschaft ist die lexikalische Fundierung der HPSG zu sehen. Meines Erachtens ist dieser Aspekt auch als zentral für die Erläuterungen in den Lernergrammatiken anzusehen. In vielen Fällen können „Regeln" – zum Beispiel zur Valenz oder zur Wortstellung – nur mit Einschränkungen aufgestellt werden, sodass viele Beispiele angeführt werden müssen, um den Sachverhalt an einzelnen Lexemen zu verdeutlichen. Trotzdem gestaltet sich die Extraktion von Erläuterungen für einen Lerner aufgrund der oben angeführten Gründe vermutlich schwieriger als bei Analysestrukturen auf der Basis von LFG.

Genereller gesehen kann hier zwischen Regeln und Beschränkungen, die sich auf die strukturellen Elemente einer Theorie beziehen und solchen Annahmen, die Beschränkungen über Beschreibungen darstellen, unterschieden werden, wobei eine klare Trennung vielleicht nicht immer möglich ist. Während Erstere (z.B. Unifikation) wohl kaum zur Erläuterung gegenüber einem Lerner verwendet werden können, handelt es sich bei Zweiteren (z.B. Kohärenz/Vollständigkeit) offensichtlich um Beschränkungen, die durchaus dem Lerner präsentiert werden können, wenn sie zum Beispiel zur Beschreibung eines fehlerhaften Satzes gelockert werden müssen.

Nicht ohne Grund wurden in Hubbard (1994) die Theorien Relational Grammar, LFG und GPSG zur näheren Betrachtung ausgewählt. Es handelt sich hierbei um nicht transformationelle Theorien, die außerdem nach Hubbard die Eigenschaft haben, „more or less ignored by language teachers" zu sein. Er möchte die Theorien genauer vorstellen, um damit Sprachlehrer und -lerner in die Lage zu versetzen, Generalisierungen zu entdecken, präzisere Regeln zu formulieren und Beziehungen zwischen Form und Bedeutung zu entdecken. Hubbard (1994, S. 59f) führt drei Implikationen aus den Grundgedanken der LFG für den Sprachunterricht auf.

- „Unlike transformational grammar and its direct descendants, grammatical functions are primitives. Thus [...] it is reasonable to state pedagogical rules using them.

[8] Im Englischen Deep- und Surface-Structure

- The lexicon is a central part of the grammar, not just dumping place for exceptions to rules. Lexical rules relate verb forms of active and passive to one another [...] This binds the learning of vocabulary and grammar more closely than is usually done in grammar texts, and certainly more so than in texts inspired by structural or earlier transformational approaches.

- Similar, in [LFG] the content of a lexical entry is much more than a phonological form and one or more meanings: It includes all of the subcategorization information for constituent and functional structure. This suggests that the learning of vocabulary, particularly verbs, should include as much of this subcategorizational information as possible. For example, it is not enough for students to memorize lists of verbs taking infinitve versus gerund complements if other important aspects of subcategorization, such as whether or not the gerund or infinitive is preceded by an NP, are ignored.“

Demnach sind also nicht nur die Beschreibungen von linguistischen Sachverhalten, sondern auch die strukturellen Elemente in der LFG so modelliert, dass sie für Lerner verständlich sind und neue Einsichten bringen können.

LFG zur Veranschaulichung

Wenn ein ICALL-System mit Hilfe einer ausführlichen, morphosyntaktischen Analyse und genauem Feedback auf Lernereingaben reagiert, dann sollte sich die Rückmeldung an den Konzepten einer Lernergrammatik orientieren. Das Feedback sollte idealerweise dem Wissen des Lerners über die zu lernende Sprache und über grammatische Beschreibungen entsprechen und in diesem Sinne leicht verständlich sein. Zusätzlich ist denkbar, dass nicht nur Feedback zu Lernereingaben erzeugt wird, sondern dass auch weitergehende Informationen zum Beispiel aus dem vom Parser verwendeten Lexikon in einem Nachschlagemodus in verständlicher Form generiert werden oder dass die Parseergebnisse dem Lerner mehr oder weniger direkt präsentiert werden. Krüger-Thielmann (1992, S. 47) schreibt dazu:„Die linguistische Adäquatheit der Beschreibung ist m.E. Voraussetzung für ein System, das mit diesem Wissen nicht nur Analysen durchführen, sondern dem Schüler auch Auskunft über Lexikoneinträge oder Grammatikregeln geben soll.“ Anzumerken ist, dass die Beschreibung einerseits natürlich „linguistisch“ adäquat sein sollte, aber andererseits vor allem auch pädagogisch adäquat sein sollte, um den Lerner optimal zu unterstützen.

Rypa und Feuerman (1995) argumentieren nicht für die Verwendung von LFG als Analysegrammatik von Lernereingaben, sondern insbesondere als pädagogische Grammatik zur Beschreibung von syntaktischen Strukturen in Unterrichtstexten. In dem von ihnen entwickelten System CALLE90 werden dem Lerner Strukturbäume zur Verdeutlichung von komplexen syntak-

tischen Phänomenen präsentiert, die ein Parser für Sätze aus Übungstexten generiert. Dazu wurde ein „Detector“ entwickelt, der sowohl aus der K-Struktur als auch aus der F-Struktur auf der Basis von Templates verständliche Hilfetexte gestalten kann. „Consequently these representations offer the potential for the student or the system to initiate exploration of linguistic phenomena and relationships, thereby leading students to a deeper awareness of structures in the target language.“(Rypa und Feuerman, 1995, S. 61) In Thomann (2002) wird ein Konzept vorgestellt, in dem die LFG als Grundlage zur Vermittlung von Regularitäten des Arabischen in Sprachkursen verwendet wird. In einigen Beispielen wird gezeigt, dass die Beschreibungen auf der Basis von LFG wesentlich kompakter als in Lehrbüchern und trotzdem verständlich dargestellt werden können. Wenn ein ICALL-System in der Lage ist, morphosyntaktische Analysen für eine Lernereingabe zu liefern, kann es sich also anbieten, auch die Erläuterungen für den Lerner aus der tatsächlich verwendeten Grammatikmodellierung abzuleiten. Schließlich kann in diesem Kontext auch die schon erwähnte Grammatik des Italienischen von Schwarze (1995) genannt werden, in der ein LFG-artiges Konzept verwendet wird.

Im Gegensatz zu diesen Beispielen erfolgt im System ALICE-chan (Levin und Evans, 1995) die Analyse der Lernereingabe mit Hilfe einer modifizierten LFG-Grammatik, wobei allerdings ein Bezug zwischen dem Feedback des Systems und der verwendeten Grammatiktheorie nicht hergestellt wird. Die Meldungen des Systems zu fehlerhaften Eingaben beschränken sich auf die Informationen, die explizite Fehlerregeln in der Grammatik liefern können. Damit ist eine Erklärung des Fehlersachverhaltes von der Grammatiktheorie entkoppelt und kann beliebig gestaltet werden.

Die hier angeführten Beispiele zur Nutzung der LFG als pädagogische Grammatik demonstrieren, dass LFG anscheinend eine der Grammatiktheorien ist, die sich für die Präsentation von morphosyntaktischen Sachverhalten eignet und somit auch in ICALL-Systemen in dieser Hinsicht einen gewissen Vorteil gegenüber anderen Theorien besitzt. Um diese Perspektive ein wenig weiter auszubauen, bietet es sich an, in einem konkreten Vergleich Konzepte bzw. Beschreibungen aus Lernergrammatiken mit den in der LFG üblichen Ansätzen in Beziehung zu setzen.

Exemplarischer Vergleich von LFG und Lernergrammatiken

Im Folgenden werden zu einigen exemplarischen Bereichen, in denen im Heringer-Korpus (siehe Abschnitt 3.4.1) häufiger Fehler auftreten, einige Zitate aus drei Grammatiken zusammengestellt. Diese werden den durch die LFG gebotenen Möglichkeiten und Beschränkungen gegenübergestellt, um auf diese Weise zu zeigen, dass die in der LFG verwendeten Strukturen den Anforderungen an eine Lernergrammatik genügen können. Hierbei wird gelegentlich auf die in Promis*D* verwendete Modellierung als Beispiel eingegangen. Anzumerken ist, dass hier nicht die linguistische oder didaktische

Adäquatheit der Darstellungen in den pädagogischen Grammatiken diskutiert werden kann (zum Tempus siehe zum Beispiel Hennig, 1997). Von Bedeutung ist dagegen hier insbesondere die Terminologie und ganz wesentlich die Art der Strukturierung des linguistischen Wissens im Vergleich mit der LFG. Bei den deskriptiven Grammatiken handelt es sich um zwei Lernergrammatiken, die explizit für den Einsatz im Deutschunterricht entwickelt worden sind, und um eine allgemeine Grammatik:

- Dreyer und Schmitt (2000): *Lehr- und Übungsbuch der deutschen Grammatik - Neubearbeitung*, Hueber, Ismaning
- Kars und Häussermann (1997): *Grundgrammatik Deutsch*, Diesterweg, Sauerländer, Frankfurt a.M., Aarau
- Eisenberg (1999): *Grundriß der deutschen Grammatik - Der Satz*, Metzler, Stuttgart, Band 2

Exemplarisch werden folgende Bereiche analysiert, die sich unter anderem bei der Analyse der untersuchten Lernerkorpora als relativ häufige Fehlerquellen gezeigt haben. 1. Artikelloser Gebrauch eines Substantivs; 2. Sechs Tempora; 3. Wortstellung/Topologie und 4. Funktionale Elemente/Valenz.

1. Artikelgebrauch

Dreyer und Schmitt (2000):

S. 22ff: Ohne Artikel werden gebraucht:
1. Personennamen [...] 2.a) unbestimmte Mengenbegriffe ohne nähere Bestimmung, z.B. [...]. *Beachten Sie*: Ist der Begriff näher bestimmt, z.B. durch Attribute oder adverbiale Angaben, steht der bestimmte Artikel: z.B. [...] b) Flüssigkeiten und Materialangaben ohne nähere Bestimmung, z.B. [...] c) Eigenschaften und Gefühle ohne nähere Bestimmung, z.B. [...]
Anmerkung
Nach den Präpositionen *ohne, zu, nach, vor* u.a. steht oft kein Artikel

Kars und Häussermann (1997):

S. 77f: a) Der bestimmte Artikel [...]
b) Der unbestimmte Artikel [...]
c) Wann nehmen wir *keinen* Artikel?

- Anrede, Name [...]
- Zeitbezeichnung ohne Präposition [...]
- Ausruf [...]
- „Aufgabe“: Beruf, Nationalität, Weltanschauung [...]
- „unbestimmte Menge“: Material, abstrakte Werte [...]

- [...]

Eisenberg (1999):

S. 158: Stoffsubstantive können [...] auch im Singular ohne Artikel oder eine andere determinierende Einheit auftreten.

S. 160: Die Möglichkeiten eines artikellosen Gebrauchs haben die Eigennamen mit den Stoffsubstantiven gemeinsam.

S. 403: Artikellosigkeit kann etwa bei Stoffsubstantiven (kalte Milch) oder pluralischem Substantiv (hohe Bäume) als Kern gegeben sein.

LFG:
Substantive, die im Singular ohne Artikel gebraucht werden können, werden in der LFG üblicherweise mit Hilfe eines Features markiert. Enthält der Lexikoneintrag zum Beispiel das Feature [NTYPE = mass], kann die PS-Regel, die Substantive ohne Artikel nur mit diesem Feature zulässt, angewendet werden (Butt et al., 1999, S. 83ff). „Normale“ Substantive werden mit dem Wert [NTYPE = norm] oder Ähnlichem versehen. Des Weiteren lassen sich in ähnlicher Weise Titel wie *Herr* oder *Professor* mit dem Feature [NTYPE = title] versehen, um ihre spezielle Distribution zu modellieren.

Obwohl es sich um eine syntaktische Beschränkung handelt, wird hier versucht, semantische Kriterien zur Eingrenzung der Gruppe der Substantive zu verwenden. Die anderen möglichen Gruppen (Flüssigkeit, Materialangaben etc.) könnten auf die gleiche Weise als Feature im Lexikoneintrag untergebracht werden. Es scheint allerdings im Rahmen der Entwicklung einer computationellen Grammatik nicht angebracht, das Feature NTYPE auch mit vielfältigeren (semantischen) Werten wie zum Beispiel [NTYPE = fluid] zu belegen, um dieselben syntaktischen Sachverhalte zu modellieren. Eigennamen, die üblicherweise auch ohne Artikel verwendet werden, erhalten zumeist eine eigene Wortart PN, um auf diese Weise die besondere Distribution modellieren zu können.

Die aus den Lernergrammatiken zitierten Elemente zeigen, dass die Angabe einer einfachen, allgemein gültigen Regel für diesen Bereich schwierig ist, da die Artikellosigkeit unter anderem nicht nur von der Eigenschaft des Substantivs abhängt. Als Beispiel seien Berufsbezeichnungen in Verbindung mit Kopula-Konstruktionen angeführt („Ich bin Lehrer“). In der Grammatik von Kars und Häussermann (1997) wird dieser Fall zu weit generalisiert, da die Verbindung mit der Kopula-Konstruktion nicht erläutert wird. Meines Erachtens ist das vermutlich auch einer der Gründe, warum in Eisenberg (1999) nicht der Versuch einer präziseren Beschreibung unternommen wird und die Erläuterungen auf unterschiedliche Bereiche verteilt werden.

Der Vergleich zeigt, dass die Anreicherung der Lexikoneinträge mit eher semantischen Informationen den Intuitionen in den Grammatiken entspricht.

Für diesen Bereich kann also nicht eine „lexikalische Regel“ oder ähnliches angegeben werden, mit der alle relevanten Lexikoneinträge erfasst werden könnten. Wenn also dem Lerner eine Analyse präsentiert wird, in der sich beispielsweise die F-Struktur durch ein NTYPE-Feature auszeichnet, kann damit die besondere Eigenschaft dieses Substantivs in verständlicher Weise hervorgehoben werden.

2. Tempora

Dreyer und Schmitt (2000):

S. 34: Die meisten Verben bilden das Perfekt und das Plusquamperfekt mit dem Hilfsverb *haben*, einige mit dem Hilfverb *sein* [...].

S. 35:

	Präsens	*Präteritum*	*Perfekt*	*Plusquamperfekt*
Sg	ich lache	ich lachte	ich habe gelacht	...
...				
Pl	...	...	...	...

	Futur I	*Futur II*
Sg	ich werde lachen	...
...		
Pl	wir werden lachen	...

S. 36: Das Futur I wird mit *werden* und dem Infinitiv, das Futur II mit *werden* und dem Infinitiv Perfekt (=*haben* oder *sein* + Partizip Perfekt) gebildet.

Kars und Häussermann (1997):

S. 19: Wir zeigen Ihnen hier die vier Zeitstufen: Präsens, Präteritum, Perfekt, Plusquamperfekt. [...] Die Zukunftsform („Ich werde dich nie vergessen“) drücken wir im Deutschen mit dem Modalverb *werden* aus.

S. 26: Das Perfekt besteht aus zwei Teilen: Hilfsverb + Partizip II
ich bin... gefahren
ich habe... gehört

S. 41: **werden** als Hilfsverb = PASSIV
werden als Modalverb = VERMUTUNG/ERWARTUNG

Eisenberg (1999):

S. 105f: Wir setzen sechs Tempora an [...]. Nur das Präsens und das Präteritum im Aktiv haben einfache, (d.h. synthetische) Formen. Perfekt, Plusquamperfekt sowie Futur 1 und 2 im Aktiv und alle Formen des Passivs sind zusammengesetzt.

S. 107: Sowohl unter den starken als auch unter den schwachen Verben gibt es solche, die das Pf[9], Pqpf und Fut2 mit **sein** bilden, und solche, die **haben** erfordern. Bei einigen Verben ist sowohl **sein** als auch **haben** möglich.

LFG

Wie oben ausgeführt, findet sich in den pädagogischen Grammatiken zumeist eine Gliederung in 6 Tempora. Die Kategorisierung von *werden* als Modalverb zur Bildung des Futurs in Kars und Häussermann (1997) muss als Ausnahme gewertet werden und ist nur durch die sehr starke Hervorhebung der kommunikativen Funktion zu begründen. Kars und Häussermann (1997, S. 260) schreiben dazu:„Von der Zukunft kann man nicht erzählen oder berichten. Man kann ihr gegenüber nur eine bestimmte Haltung einnehmen."

Die Gliederung in 6 Tempora lässt sich leicht auf eine LFG-Grammatik übertragen. Üblicherweise wird ein TENSE-Feature angenommen, das das Tempus eines Satzes mit einfacher finiter Verbform bestimmt. Im Fall von Auxiliar-Konstruktionen wird in der neueren LFG angenommen, dass das Auxiliar selber nur die Tempusbestimmung beisteuert und diese Information um Angaben des Partizips ergänzt werden (zum Beispiel [PERF = +]). Das bedeutet vor allem, dass das Auxiliar im Gegensatz zur ursprünglichen Modellierung in Kaplan und Bresnan (1982) kein PRED-Feature besitzt; das Partizip bzw. der Infinitiv steuern das PRED-Attribut bei und ergeben somit eine flache F-Struktur, in der das Auxiliar nur eine untergordnete Rolle spielt. Der Typ des Auxiliars (haben/sein) wird durch ein weiteres Feature AUXTYPE festgelegt. Aus diesen Angaben, die sich für einen einfachen Aussagesatz alle auf der obersten Ebene einer F-Struktur befinden, lassen sich meines Erachtens leicht verständliche Meldungen generieren (siehe Abschnitt 5.5).[10]

Eine gewisse Schwierigkeit besteht in den Angaben zur Finitheit in synthetischen Formen. Mit Hilfe der Features muss festgelegt werden, dass das Auxiliar finit ist und gleichzeitig, dass das Partizip nicht finit sein darf. Unter anderem aus diesem Grund wird in Butt et al. (1999, S. 65f) die so genannte *m*-Struktur vorgeschlagen, in der einige der verbalen morphologischen Eigenschaften in Anlehnung an die K-Struktur hierarchisch kodiert werden.

3. Wortstellung

Dreyer und Schmitt (2000):

[9] Pf = Perfekt, Pqpf = Plusquamperfekt, Fut2 = Futur 2

[10] Modalverben dagegen führen ein eigenes PRED-Feature ein, sodass in diesem Fall eine tiefere Verschachtelung realisiert wird.

S. 83: *dass*-Sätze sind Nebensätze [...], d.h., das konjugierte Verb steht am Ende des Satzes.

S. 126: 1. Ein Satz besteht aus bestimmten *Satzgliedern*: Subjekt, Prädikat, Objekten, adverbialen Angaben usw. [...]
5. Der Hauptsatz ist ein unabhängiger, vollständiger Satz. Das konjugierte Verb steht immer in der Position II.
6. Im Hauptsatz kann das Subjekt von der Position I auf die Position III (IV) wechseln, d.h. es bewegt sich um das konjugierte Verb (Position II) wie um eine Achse.
Anmerkungen:
1. [...]
2. Der Wechsel des Subjekts von der Position I zur Position III wird im Folgenden *Umstellung* genannt.

Kars und Häussermann (1997):

S. 142: Der Nebensatz mit *so daß (daß)* steht immer rechts vom Hauptsatz.

S. 188: Auf Position II steht immer das markierte Verb.
[...]
Wenn Position I mit einem anderen Element besetzt ist, folgt der Nominativ rechts von Position II.

S. 192: Tendenz nach links haben: Nominativergänzung, [...]

S. 194: Tendenz nach rechts haben: die Präpositionalergänzung, [...]

S. 196: Im Nebensatz steht das markierte Verb am Ende.

S. 200: Das Nachfeld

Eisenberg (1999):

S. 308: Wo dieser Begriff [Spannsatz] verwendet wird, expliziert man ihn über die Endstellung des finiten Verbs.

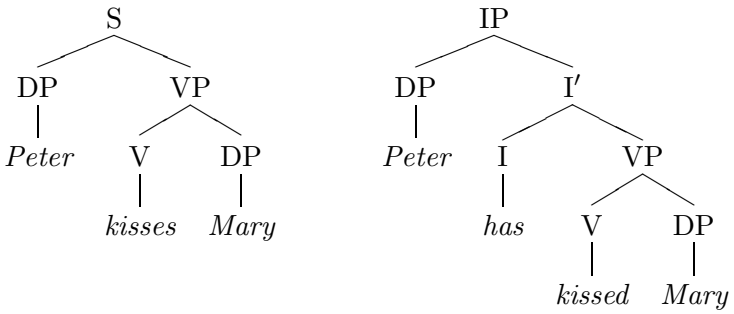
S. 388: Allgemeingültige und vollständige Beschreibungen der Topolgie des Kernsatzes unterscheiden heute meist sechs Positionen oder Felder gemäß [Abbildung] 6 [: Konj, Vorfeld, Fin, Mittelfeld, Infiniter VK und Nachfeld], die natürlich unterschiedlich benannt und weiter differenziert werden.

S. 389: Sätze vom Typ 8 beschreibt man manchmal mit dem Begriff Subjekt-Verb-Inversion [...].

LFG

In den erwähnten LFG-Einführungen (Bresnan, 2001; Falk, 2001) werden zumeist CP/IP-basierte Phrasenstrukturen verwendet, um die damit verbundenen Generalisierungen auszudrücken. Allerdings kann ein $\overline{X}$-Ansatz nicht konsequent durchgehalten werden, da es unter anderem in der LFG keine Transformationen gibt. Außerdem werden meist weitere Prinzipien angenommen, die die Entwickler zwingen, vom strengen $\overline{X}$-Schema abzuweichen. Dazu gehört zum Beispiel das Prinzip der „Ökonomie der Beschreibung“, welches besagt, dass alle PS-Knoten optional sind und nicht verwendet werden, es sei denn, um eine wohlgeformte F-Struktur zu produzieren oder zur Semantik beizutragen. Das Prinzip der lexikalischen Integrität besagt unter anderem, dass nur *Wörter* als Elemente an den Blättern eines Baumes angenommen werden dürfen. Die Konsequenz ist, dass für sehr ähnliche Strukturen unter Umständen zwei unterschiedliche K-Strukturen verwendet werden. Die PS-Bäume in Beispiel (3.12) stammen aus Falk (2001, S. 50f), der für Sätze mit finiten Vollverben S als Kategorie und für Sätze mit Auxiliaren IP vorschlägt.

(3.12)



Wie die Zitate zeigen, wird dagegen in allen deskriptiven Grammatiken die Sichtweise des topologischen Feldermodells auf die Phrasenstruktur vertreten, auch wenn die Terminologie und die Ausführlichkeit voneinander abweichen.

Butt et al. (1999, S. 40f) schreiben zum Konzept der ParGram-Grammatik: „Within our LFG approach we could have written rules which would have allowed for this superficially clean representation of the c-structure [dem $\overline{X}$-Ansatz]. [...] Within ParGram we have therefore adopted the approach of overt labeling of c-structures and take up the position that the particular distinctions we have made in terms of the formulation of the c-structure rules correspond to clear descriptive and typological distinction found within and across languages.“ Eine ähnliche Strukturierung mit der Betonung des Deskriptiven findet sich in der hier entwickelten Grammatik, die ein „klassisches“ Vokabular verwendet (siehe Abschnitt 3.4).

In Clément et al. (2002) wird der Versuch unternommen, das Feldermodell in einer LFG-Grammatik für das Deutsche zu realisieren. Es wird

gezeigt, dass mit Hilfe einer sehr kompakten PS-Grammatik ein wesentlicher Bereich der Phrasenstruktur des Deutschen abgedeckt werden kann.[11] Dieses Konzept, in dem sogar Feldnamen wie beispielsweise „Vorfeld“ als Kategorien in den PS-Regeln verwendet werden, wurde zwar in der für Promis*D* entwickelten Grammatik nicht verwirklicht und kann daher nicht für Rückmeldungen genutzt werden, aber es finden sich zentrale Ideen des Topologiemodells auch in der Promis*D*-Grammatik wieder, wie im nächsten Abschnitt gezeigt wird. Damit wird einerseits erreicht, dass die Darstellung der Grammatik den Konzepten in Lernergrammatiken ähnelt und andererseits, dass eine kompakte und möglichst umfassende Grammatikmodellierung zur automatischen Analyse zur Verfügung steht.

4. Valenz

Dreyer und Schmitt (2000):

S. 67: 1. Die transitiven Verben (Verben die ein Akkusativobjekt haben) zeigen eine Handlung: [...]
2. Die intransitiven Verben (Verben, die kein Akkusativobjekt haben) zeigen das Ergebnis einer Handlung. [...]

S. 70: Rektion bedeutet, dass bestimmte Verben einen bestimmten Kasus fordern.
Es gibt keine festen Regeln, welches Verb welchen Kasus „regiert“.

S. 342: *das Objekt* = im Satz: 1. das Akkusativobjekt [...]. 2. das Dativobjekt [...]. 3. das Genitivobjekt [...].

S. 344: *das Subjekt* = im Satz: Der Satzteil im Nominativ [...]

S. 345: *transitive Verben* = Verben, die ein Akkusativobjekt bei sich haben können: [...]

Kars und Häussermann (1997):

S. 3: Ein ganzer Satz braucht normalerweise ein Verb und ein Substantiv oder ein Pronomen: [...]

S. 9: Im Verb steckt ein Programm. Dieses Programm ist konstant. Welche Mitspieler (Ergänzungen) das Verb wählt, das liegt in seinem Programm fest.[...]
Das Verb *blühen* braucht nur einen Mitspieler (eine Ergänzung): [...]

[11] Auch im Rahmen von anderen Grammatiktheoriekonzepten wurden verschiedene Ansätze zur Integration von Ideen des Feldermodells eingesetzt, zum Beispiel im Dependenzmodell (Bröker, 1999) oder in der HPSG (Müller, 1999).

S. 10: Das satzbauende Programm, das im Verb steckt, nennen wir Valenz.

S. 13: Der Akkusativ nennt meistens das Ziel oder das Objekt (das „direkte Objekt“). [...]
Der Dativ nennt – wenn er vom Verb abhängt – den Partner. [...]

Eisenberg (1999):

S. 58: Wir [...] verstehen unter Valenz ausschließlich syntaktische Valenz. [...] Unter der Stelligkeit eines Verbs versteht man also die Zahl seiner gleichzeitig möglichen *Ergänzungen* (Subjekt und Objekte).

S. 59: Der häufigste Typ unter den zweistelligen Verben ist das transitive Verb der traditionellen Grammatik wie **sehen** [...], das einen Nominativ (das Subjekt) und einen Akkusativ (das direkte Objekt) regiert [...].

LFG

Die funktionale Perspektive dominiert in der LFG und spiegelt sich in den F-Strukturen wieder. Außerdem beziehen sich die wichtigsten Prinzipien auf die Ebene der funktionalen Elemente, wie im vorhergehenden Abschnitt erläutert wurde. In der theoretischen LFG werden die Ergänzungen („grammatische Argument-Funktionen“) üblicherweise mit Hilfe einer Reihe von Prinzipien und Regeln aus den Angaben in der A-Struktur (Argument-Struktur, siehe Seite 52), also der Ebene der thematischen Rollen, abgeleitet. Mit Hilfe der A-Struktur werden zusätzlich lexikalische Regeln strukturiert, die zum Beispiel Passiv-Formen generieren. Hierfür sind allerdings bestimmte Annahmen über die thematischen Strukturen notwendig, die anderen bekannten Ansätzen widersprechen (Dowty, 1991).

In der praktischen Entwicklung einer Grammatik wird der Bereich der Generierung von Lexikoneinträgen auf der Basis der A-Struktur vermutlich selten implementiert; zum Beispiel wurde im ParGram-Projekt diese Ebene nicht berücksichtigt, sodass die lexikalischen Regeln diese Ebene nicht ausnutzen können (Butt et al., 1999, S. 60). Auch in dem hier verwendeten Lexikon sind die funktionalen Einheiten direkt in der Subkategorisierungsliste der Verben enthalten. Die A-Struktur findet in der Implementierung keine Berücksichtigung, sondern die thematischen Rollen sind direkt im Lexikoneintrag enthalten. Der folgende Lexikoneintrag für das Verb *lacht* mit der Spezifikation [FORM = agent] für das Subjekt kann das verdeutlichen.

(3.13) lacht, V,
$$\begin{bmatrix} \text{PRED} & = \text{`HEITERKEIT} < \uparrow \text{SUBJ} > \text{'} \\ \text{TENSE} & = \text{pres} \\ \text{SUBJ} & = \begin{bmatrix} \text{CASE} & = \text{nom} \\ \text{PERS} & = 3 \\ \text{NUM} & = \text{sg} \\ \text{FORM} & = \text{agent} \end{bmatrix} \end{bmatrix}$$

Außerdem wird mit Hilfe des Wertes des PRED-Features auch die Valenz des Verbs festgelegt ('HEITERKEIT <↑SUBJ>'). Mit Hilfe der Kohärenz- und Vollständigkeitsprinzipien wird auf diese Weise eine korrekte Struktur bestimmt. Üblicherweise können die Elemente der Subkategorisierungsliste wie erwähnt auch außerhalb der „Klammer" aufgeführt werden. Damit soll angedeutet werden, dass das Element außerhalb der Liste zwar eine syntaktische Funktion für das Verb spielt, aber nicht durch eine thematische Rolle spezifiziert wird. Dieser Fall tritt zum Beispiel bei expletiven Subjekt und in so genannten Raising-Konstruktionen auf, in denen ein Element mehrere grammatische Funktionen zum einen für das Matrixverb und zum anderen für das eingebettete Verb übernimmt. Allerdings taucht die Problematik der multiplen grammatischen Funktionen von einzelnen Elementen im Satz als Thema nicht in den von mir untersuchten Lernergrammatiken auf.

In allen untersuchten Lernergrammatiken ist die funktionale Perspektive auf die Elemente in einem Satz vorherrschend. Begriffe wie „Subjekt" und „Objekt" werden zum großen Teil, wie aufgeführt, definiert und erläutert. Eine Einbindung von thematischen Rollen zur Bestimmung der grammatischen Funktionen, wie sie in bestimmten Varianten der LFG vertreten wird, kommt allerdings in keiner pädagogischen Grammatik vor. Wie erwähnt sind dagegen in meinem Vollformenlexikon die funktionalen Elemente direkt in der Subkategorisierungsliste der Verben enthalten und die thematischen Rollen sind Elemente der entsprechenden F-Struktur, um eine Interpretation zu unterstützen. Das PRED-Attribut enthält eine Repräsentation der Bedeutung und eine Subkategorisierungsliste, wobei die Elemente der Liste den Subkategorisierungsmöglichkeiten des Verbs entsprechen.

Die Entscheidung von Kars und Häussermann (1997), nicht von Subjekt und Objekt, sondern von Nominativ- und Akkusativergänzung zu sprechen, wird nicht plausibel begründet:„Und statt *Subjekt* und *Objekt* sagen wir *Ergänzung*, mit diesem Begriff kann man bei der Erklärung der Satzmodelle und der Wortposition im Satz besser operieren."(Kars und Häussermann, 1997, S. 259) Vermutlich sollen hier die Gemeinsamkeiten zwischen den verschiedenen Ergänzungstypen hervorgehoben werden, die durch die scheinbar unterschiedlichen Begriffe „Subjekt" und „Objekt" verdeckt werden könnten. Es muss angemerkt werden, dass auch einige Lehrwerke für Deutsch als Fremdsprache diese Terminologie verwenden und die Grundgrammatik sich insofern angepasst hat. Schließlich kann die Unterscheidung in unterschiedliche Typen von grammatischen Funktionen zum Beispiel in

Argument-Funktionen und Non-Argument-Funktionen aus der LFG für eine natürlichsprachliche Ausgabe genutzt werden. Diese Unterscheidung entspricht in weiten Teilen den Begriffen *Ergänzungen* beziehungsweise *Angaben*. Es kann also auch für den Bereich der Valenz davon ausgegangen werden, dass die Strukturierung der Valenz in der LFG für einen Lerner ohne größere Schwierigkeiten verständlich sein sollte.

Fazit

Damit ist der Teil des exemplarischen Vergleichs von Terminologie und Strukturierung in pädagogischen Grammatiken sowie in der LFG abgeschlossen. Es hat sich gezeigt, dass in den vorgestellten Bereichen die Modellierung der linguistischen Phänomene in der LFG ausreichend Übereinstimmung zwischen ihrer Beschreibung und den Erläuterungen in pädagogichen Grammatiken existiert, um für den Lerner relevantes Feedback erzeugen zu können. Insbesondere für den Bereich des funktionalen Aufbaus und den damit verbundenen Kongruenz- und Rektionsbeziehungen lassen sich präzise Meldungen über die Struktur beziehungsweise über Fehlerangaben in der Struktur erzeugen. Gerade dieser Bereich ist bedeutsam, da hier relativ viele Fehler gemacht werden, wie in der Auswertung des Heringer-Korpus in Abschnitt 5.4.2 zu erkennen ist. Ein ähnliches Bild ergibt sich für den Bereich der Valenz. Auch hier kann auf der Basis der Subkategorisierungsliste und der F-Struktur eine sinnvolle Beschreibung gegeben werden, die dem grammatischen Wissen des Lerners entsprechen dürfte.

3.3.3 Zusammenfassung

In diesem Abschnitt wurde gezeigt, dass die LFG eine Theorie ist, die die Anforderungen an eine Grammatiktheorie zum Einsatz in einem ICALL-System erfüllen kann. Dabei hat sich die LFG besonders in typologischer Hinsicht als geeignet erwiesen, da die morphosyntaktischen Phänomene einer Vielzahl von divergierender Einzelsprachen beschrieben worden sind. Auch für das Deutsche wurde bereits eine umfangreiche Grammatik entwickelt. Für den Einsatz der LFG in einem sprachverarbeitenden System ist natürlich insbesondere der letzte Aspekt von entscheidender Bedeutung.

Es handelt sich dabei außerdem um eine Theorie, die schon in den Ursprüngen (zum Beispiel Kaplan und Bresnan, 1982) so aufgebaut wurde, dass die verwendeten Strukturen formalen Anforderungen genügen. Da die formalen Mittel relativ einfach in Implementierungen umgesetzt werden können, lässt sich auch die zweite angeführte Anforderung erfüllen.

Schließlich wurde in diesem Abschnitt gezeigt, dass die Theorie der LFG gut dazu geeignet scheint, linguistisches Wissen in einem System zu modellieren, das auch in der Lage sein soll, Rückmeldungen zu Analysen der Lernereingaben geben zu können. Wie gezeigt wurde, ergeben sich sowohl

aus der F-Struktur zur Kodierung von morphosyntaktischen Eigenschaften als auch aus der K-Struktur zur Beschreibung der Wortstellung große Übereinstimmungen mit den Beschreibungen in pädagogischen Grammatiken.

1. Die Verwendung des Artikels im Deutschen lässt sich nicht durch eine allgemeine Regel beschreiben, sondern muss für jedes Lexem angegeben werden. Dieses drückt sich in den Lernergrammatiken dadurch aus, dass eine unvollständige und vage Liste von Oberbegriffen gegeben wird, mit denen die Verwendung charakterisiert wird. Diese lexikalische Orientierung ist in der LFG ein fundamentales Konzept, welches hier unter anderem seine Bestätigung findet.

2. Das Tempus wird nicht durch die Verknüpfung von Stamm und Suffix (Bewegung von V nach I) innerhalb des PS-Baumes realisiert, sondern als Information, die direkt mit einem Lexem beziehungsweise dem Lexikoneintrag verknüpft ist. Auch die Zusammensetzung einer synthetischen Form ergibt sich direkt aus den entsprechenden Features der F-Struktur.

3. Obwohl die Beschreibung der Wortstellung in der theoretischen Literatur zumeist basierend auf dem CP/IP-Konzept beruht, wird in der praktischen Grammatikentwicklung hiervon abgewichen, wie das erwähnte Beispiel Butt et al. (1999) zeigt. Für das Deutsche kann sogar die Wortstellung präzise dem topologischen Modell entsprechend strukturiert werden, um die gewünschten Generalisierungen zu erreichen, wodurch eine weitgehende Übereinstimmung mit den Lernergrammatiken gegeben ist.

4. Die hierarchische Struktur der F-Struktur wird in erster Linie durch die vom Verb lizensierten grammatischen Funktionen bestimmt. Damit steht, wie auch in einer Lernergrammatik, die funktionale Perspektive auf einen Satz im Vordergrund. Zudem lassen sich in einem PS-Baum die Abschnitte der grammatischen Funktionen auf der Basis der Annotationen mühelos identifizieren.

Damit wurde gezeigt, dass die Repräsentationen der LFG eine gute Basis für die Generierung von Beschreibungen darstellen, die sich zur Erläuterung für einen Lerner eignen (siehe Abschnitt 5.5).

3.4 Beschriebene Phänomene

In diesem Abschnitt wird erläutert, welche sprachlichen Phänomene von der in Promis*D* eingesetzten Grammatik berücksichtigt werden. Die Modellierung der Grammatik erfolgte dabei in erster Linie auf der Grundlage von Lernereingaben bei der Benutzung des hier vorgestellten Systems und auf der

Grundlage des Heringer-Korpus (Heringer, 1995). Es wurde also nicht angestrebt, einen bestimmten Umfang der Grammatik in Bezug auf eine Menge an Phänomenen zu entwickeln, sondern die Entwicklung orientierte sich an den in den Beispielkorpora enthaltenen Sätzen. Dabei wurde versucht, die in einer Art evolutionären Prozess neu hinzugefügten syntaktischen Bereiche so allgemein wie möglich zu beschreiben, um so von den konkreten Beispielen aus eine Generalisierung zu erreichen.

Hervorzuheben ist, dass hier nicht eine Argumentation für bestimmte Analysen erfolgen soll. Die Konstruktion der Grammatik basiert auf vielerlei Quellen wie zum Beispiel Butt et al. (1999), Müller (1999) oder Langer (2001), in denen über die Gestaltung einer umfangreichen Grammatik zur rechnergestützten Analyse berichtet wird, aber auch auf deskriptiven Grammatiken des Deutschen wie zum Beispiel die von Eisenberg (1999).

Im folgenden Abschnitt werden zunächst die verwendeten Korpora kurz beschrieben, im Anschluss sollen einige Details des implementierten Formalismus präsentiert werden und schließlich erfolgt dann ein Überblick über die Abdeckung der in Promis*D* eingesetzten Grammatik. Damit sollte eine Einschätzung der von der Grammatik abgedeckten, sprachlichen Phänomene möglich sein, die sich aus der Menge der aus den Korpora gewählten Sätzen ergibt.

3.4.1 Korpora

Um einerseits Material für eine Evaluation des hier entwickelten Verfahrens zu haben (siehe Kapitel 5.4.2) und andererseits den Umfang des zu modellierenden Sprachfragments bestimmen zu können, wurden zwei Korpora herangezogen. Bei dem einen handelt es sich um Sätze, die ich bei Probeläufen des Programms mit Studierenden an der Humboldt-Universität und der Freien Universität in Berlin gesammelt habe. Das andere Korpus wurde von Heringer 1995 aus Materialien des Goethe-Instituts in München aus den 70er Jahren zusammengestellt. Die folgende Tabelle 3.1 enthält die wesentlichen Informationen über die Korpora.

Das HU/FU-Korpus entstand durch Studierende, die Promis*D* ausprobiert haben. Sie kamen zumeist aus osteuropäischen Ländern sowie zum kleineren Teil aus Westeuropa und Südamerika und nahmen alle an Mittelstufen-Kursen in Deutsch als Fremdsprache am Sprachenzentrum der HU teil oder waren Studierende des Kollegs der FU Berlin. Als wesentliche Besonderheit dieses Korpus muss hier erwähnt werden, dass die Studierenden vermutlich durch die vom Pogramm simulierte Dialogsituation oft dazu „verleitet“ wurden, nur Teilphrasen als Antworten auf die konkreten W-Fragen einzugeben. Auch darauf muss das System also in der Lage sein, in geeigneter Weise zu reagieren. Wie das geschieht, wird im nächsten Kapitel ausführlicher beschrieben. Schließlich muss erwähnt werden, dass es sich bei den

	HU/FU-Korpus	Heringer-Korpus
fehlerhafte Sätze	103	7107
Wörter	683	82013
Wörter/Satz	6,6	11,5
Produktion	Promis*D*-Nutzer 2000	? (70er Jahre)
Herkunft; Meta-Information	ausländische Studierende der HU und FU Berlin	unbekannte Lerner am Goethe-Institut München
Stufe	Mittelstufe	ca. Mittelstufe
Besonderheiten	z.T. nur Phrasen	vollständige Sätze, keine weiteren Meta-Informationen

Tabelle 3.1: Übersicht über die Korpora

103 Sätzen um die morphosyntaktisch fehlerhaften Sätze handelt. Es wurden auch korrekte Sätze produziert, die hier aber weiter keine Rolle spielen sollen.

Es ist nicht bekannt, auf welche Art das Heringer-Korpus (Heringer, 1995) entstanden ist. Da es sich ausschließlich um vollständige Sätze handelt, wurde das Korpus vermutlich schriftlich produziert, und auch über das Niveau der Lerner kann nur gemutmaßt werden. Dieses Korpus wurde von Heringer und seinen Mitarbeitern vollständig auf Fehler annotiert und in ein Sprachlernsystem integriert, um Lernern mit Hilfe von fehlerhaften Sätzen die Vermeidung von häufigen Fehlern zu vermitteln.

Zur Entwicklung der Grammatik wurden die von mir gesammelten Lernereingaben vollständig berücksichtigt, während aus dem Heringer-Korpus nur eine Anzahl von 75 Sätzen verwendet wurde. Die Sätze aus dem Heringer-Korpus wurden zufällig aus der Menge der Sätze ausgewählt, die morphosyntaktische Fehler enthalten. Diese Einschränkung wurde vorgenommen, da im Korpus auch Sätze mit orthographischen, rein morphologischen und anderen Fehlern enthalten sind, die grundsätzlich nicht mit den in Promis*D* realisierten Verfahren zur Erkennung von morphosyntaktischen Fehlern identifiziert werden können und daher bei der Auswahl ausgeklammert wurden. Die Grammatik umfasst 1486 PS-Regeln, wenn alle als optional markierten Elemente der rechten Seite einer Regel ausmultipliziert werden. Weitere Erläuterungen zu den Eigenschaften der Korpora und den ausgewählten Sätzen finden sich in Abschnitt 5.4.2.

3.4.2 Skizze der technischen Realisierung

Grammatik

Zu Beginn erfolgen einige Ausführungen zum technischen Aufbau der Grammatik und im Anschluss daran wird der Aufbau des Lexikons erläutert. Die PS-Regeln in meiner Grammatik haben die folgende allgemeine Form in

Prolog-Notation (cf. Rohrer und Schwarze, 1988), was der anschließenden, klassischen LFG-Regel in Beispiel (3.15) entspricht:

(3.14)
```
s(_,_,_) ---> [np(subj,[subj=[case=nom|_]|_],_),
                vp(_,_,_)].
```

(3.15)

S	→	NP		VP
		↑SUBJ=↓		↑=↓
		↓CASE=↓NOM		

Der Pfeil (`--->`) wurde als Operator neu definiert. Die 3 Argumente in den „POS-Fakten“ (`vp(_,_,_)`) haben die Funktion, erstens eine mögliche grammatische Funktion, zweitens die annotierte F-Struktur und drittens einen möglichen Fehlerwert zu enthalten. In der zweiten Position können auch weitere beschränkende Attribute enthalten sein, die die zu diesem Knoten gehörige F-Struktur spezifizieren beziehungsweise beschränken. Im Beispiel ist das Feature `case=nom` eingefügt, das also den Kasus des Subjekts festschreibt, wenn nicht ein Fehlerfall eintritt. Die Verwendung des Fehlerwertes an der dritten Stelle wird in Abschnitt 5.2.1 (Seite 129) erläutert.

Nicht terminale Elemente können als optional (eingebettet in den Funktor `opt/1`) markiert werden; dann dürfen die Elemente ein- oder keinmal auftauchen. Außerdem können die Elemente in einem `kleene/1`-Term enthalten sein, mit dem der Postfix-Operator * (Kleene-Stern) modelliert wird, der üblicherweise in regulären Ausdrücken Verwendung findet. Dann dürfen die Elemente keinmal oder beliebig oft im Satz enthalten sein (siehe Beispiele (3.16) und (3.17)).

Grammatische Funktionen können elementar sein wie im Beispiel 3.14 oben. Es gibt in der Implementation auch die Möglichkeit, den theoretischen Ansätzen in der LFG folgend, eine Art *functional uncertainty* der Länge 1 zu modellieren. Anstelle einer Funktion muss dafür der Platzhalter `gf` (= grammatical function) eingesetzt werden. Wenn in der Abarbeitung einer PS-Regel der `gf`-Platzhalter identifiziert wird, stellt eine Liste die in der speziellen Grammatik verwendeten Funktionen zur Verfügung. Für alle Elemente dieser Liste werden dann entsprechende F-Strukturen und darauf aufbauend Chart-Items generiert. Hierarchisch stärker strukturierte F-Strukturen, die üblicherweise mit Hilfe der *functional uncertainty* modelliert werden, können auf diese Weise nicht konstruiert werden. Außerdem lässt sich festlegen, dass die F-Struktur in einer Liste enthalten sein soll, womit beispielsweise die in der LFG verwendete Annotation ↓∈(↑ADJUNCT) modelliert wird. In der Implementation ist das nur für die grammatische Funktion des Adjunkts und für „funktionslose“ Listen wie etwa im Fall der Koordination vorgesehen.

(3.16)
```
np(_,_,_) --->
            [det(_,_,_),
             npbar(_,_,_),
             kleene(pp(adjunct,_,_))].
```

(3.17) NP → DET NPbar (PP)*
↑=↓ ↑=↓ ↓∈(↑ADJUNCT)

Die erste Regel zeigt in Prolog-Notation, wie eine Präpositionalphrase als Adjunkt in einer Liste an eine Nominalphrase angefügt sein kann. Die zweite Regel (3.17) zeigt noch einmal dasselbe in typischer LFG-Notation.

Die folgende Regel zeigt vereinfacht, wie zwei NPn koodiniert werden können. Dabei wird von der Regel festgelegt, dass der Kasus der einzelnen NPn übereinstimmen muss und die koordinierte NP den Numerus Plural besitzt. Die Kongruenz des Kasus wird mit Hilfe des identischen Variablennamens `C` erreicht.

(3.18)
```
np(_,_,_) --->
            [np([],[[case=C|_]|_],_),
             konj(_,[konjtype=coord,num=pl|_],_),
             np([],[[case=C|_]|_],_)].
```

In welcher grammatischen Funktion die Gesamt-F-Struktur auftaucht, wird in einer anderen Regel festgelegt. Die folgende Abbildung (3.19) zeigt beispielhaft eine Teil-F-Struktur für die Phrase „der Mann und die Milch“.

(3.19)
$$\left[\begin{array}{ll} & \left\{\begin{array}{ll} \texttt{konjtype} & = \texttt{coord} \\ \texttt{num} & = \texttt{pl} \\ \left[\begin{array}{ll} \texttt{num} & = \texttt{sg} \\ \texttt{detype} & = \texttt{def} \\ \texttt{case} & = \texttt{nom} \\ \texttt{pred} & = \texttt{`MANN'} \end{array}\right] & \\ \left[\begin{array}{ll} \texttt{num} & = \texttt{sg} \\ \texttt{detype} & = \texttt{def} \\ \texttt{ntype} & = \texttt{mass} \\ \texttt{case} & = \texttt{nom} \\ \texttt{pred} & = \texttt{`MILCH'} \end{array}\right] & \end{array}\right\} \\ \texttt{subj} = & \end{array}\right]$$

Mit Hilfe dieser technischen Lösungen sind mit gewissen Einschränkungen die wesentlichen Konzepte der Theorie der LFG realisiert worden, wie sie für den Grammatikteil, das heißt für die annotierten PS-Regeln angenommen werden. Anschließend wird nun die technische Umsetzung der Konzepte zum Lexikon erläutert.

Lexikon

Die Attribut-Wert-Gleichungen kommen im Lexikon in den folgenden drei Ausprägungen vor. Disjunktionen – auch für atomare Werte – wurden nicht implementiert.

Das erste Attribut-Wert-Paar stellt eine einfache definierende Gleichung dar, die den Wert für ein bestimmtes Attribut festlegt. Die zweite Gleichung

LFG-Form	Prolog-Notation
NUM = sg	`num = sg`
CASE $=_c$ nom	`case == nom`
GEN $\neq$ m	`gen \= m`

Tabelle 3.2: Gegenüberstellung LFG – Prolog Attribut-Wert-Gleichungen

ist mit einem kleinen c annotiert, um anzudeuten, dass es sich hier um eine so genannte „constraining equation“ handelt. Damit wird gefordert, dass ein bestimmter Wert zu einem bestimmten Attribut zwingend vorhanden ist. Es muss also eine einfache definierende Gleichung in der F-Struktur vorhanden sein. Die letzte Gleichung legt fest, dass ein bestimmtes Attribut nicht mit einem bestimmten Wert auftreten darf. Damit wird im Wesentlichen eine kompaktere Repräsentation der Eigenschaften eines Lexems erreicht und so beispielsweise die Zahl der Einträge für Adjektive deutlich reduziert. Allerdings ergeben sich durch diese Notation in bestimmten Fällen gewisse Schwierigkeiten bei der Fehleridentifizierung, auf die in Abschnitt 5.5.3 (Seite 170) eingegangen wird.

Einträge im Lexikon können entweder als Vollformen oder mit Hilfe von Templates abgekürzt eingetragen werden. Eine Vollform sieht folgendermaßen aus (siehe auch Beispiel (3.11), Seite 52).

(3.20)
```
lex('sehe',
        v(_,[pred='SEHEN'(subj,obj_dir),
             mood=fin, modus=ind, tense=pres,
             subj=[form=agent, pers='I', num=sg|_],
             obj_dir=[form=gegenstand, case=acc|_]|_],_)).
```

Mit Hilfe von Templates[12] lassen sich die Angaben für regelmäßige Formen drastisch verkürzen, die mit Hilfe eines Compilers in die Vollform expandiert werden.

(3.21)
```
lexx('Ecke',n,['',n_pl],[pred='ECKE', gen=f|_]).
```

Das Substantiv *Ecke* mit der Wortart `n` hat im Singular die regelmäßige Form ohne weitere Endungen, während im Plural die Endung *-n* angefügt werden muss. Die Liste `[' ',n_pl]` dient nur dem Zugriff auf die richtigen Templates, das heißt, es werden damit unter anderem keine Aussagen über die Anzahl der tatsächlich generierten Vollform-Einträge gemacht. Die Informationen in der abschließenden F-Struktur sind die invariablen Angaben. Die Bedeutung wird als `ECKE` und das Genus mit `f` kodiert. Damit sind alle wesentlichen theoretischen Konzepte der LFG in der Implementation umgesetzt.

[12] Das Konzept der lexikalischen Regeln wird in der Implementierung nicht grundsätzlich berücksichtigt, obwohl einige Templates diese Funktion übernehmen können.

3.4.3 Linguistische Modellierung

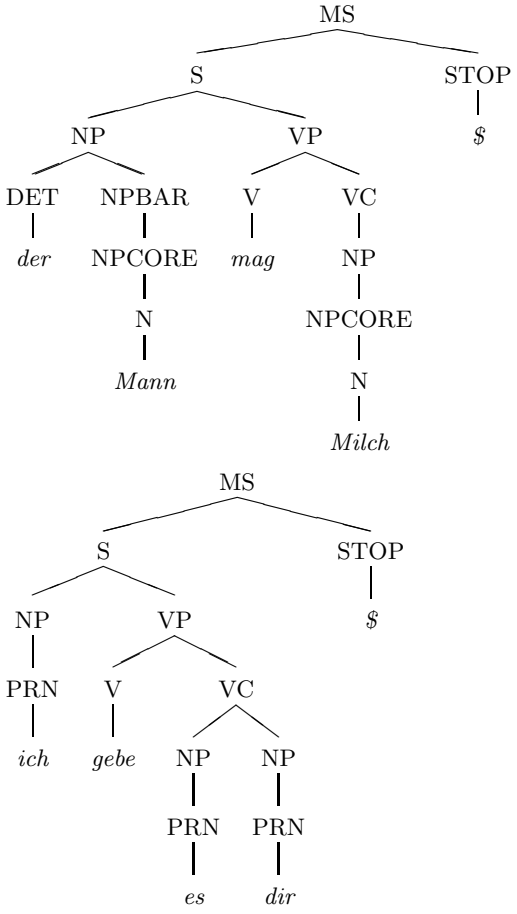
Im Folgenden wird auf einzelne linguistische Bereiche der Grammatik eingegangen, wobei nur Konstruktionen erwähnt werden, die linguistisch interessant sind, im Vergleich mit einer Lernergrammatik relevant sind oder die Schwierigkeiten bei der Modellierung bereiten. Es muss noch einmal betont werden, dass hier nicht für bestimmte, linguistisch motivierte Konstruktionen argumentiert werden soll, sondern lediglich einzelne Bereiche präsentiert werden, die eine gewisse Plausibilität besitzen.

Die grobe Struktur eines einfachen deklarativen Satzes wurde aus der ParGram-Grammatik übernommen. Einerseits wird also versucht, eine hierarische Struktur mit rechtsverzweigenden Teilbäumen zu modellieren. Andererseits werden beispielsweise für das Mittelfeld eher flache Bäume angenommen, um die Variabilität in diesem Bereich darzustellen.

Die Beispiele in (3.22) auf Seite 77 demonstrieren einerseits den relativ freien Gebrauch der Kategorien und andererseits die an das Deutsche angepasste Strukturierung gegenüber herkömmlichen PS-Konzepten. Die Kategorie VC steht für „VerbComplement“ und fasst alle Komplemente in einer Art Mittelfeld zusammen. Im zweiten Beispiel sind beide Komplement-NPn als Schwestern unter dem VC-Knoten eingefügt. Während mit der Kategorie VC ein Teil des Mittelfeldes abgedeckt wird, ist die Subjekt-NP in ihrer invertierten Stellung direkt nach dem finiten Verb eine direkte Tochter der VP (hier nicht dargestellt). Neben der einfachen Grundwortstellung sind in der Grammatik außerdem einfache subordinierte Nebensätze in Erst- und in Zweitstellung sowie die Koordination von zwei Hauptsätzen modelliert worden.

Schließlich sei hier beispielhaft ein Phänomen erwähnt, dass zwar als ein relativ grundlegender Bestandteil einer computerlinguistischen Grammatik erscheinen mag, sich aber in den untersuchten Korpora nicht findet. Es handelt sich um Partizipkonstruktionen in NPn, wie zum Beispiel „der den Mann beißende Hund“, die daher auch nicht in der vorliegenden Grammatik modelliert worden. Das gilt beispielsweise auch für Konjunktiv-I-Formen und doppelte Infinitive. Hier deutet sich an, dass im ICALL-Szenario anscheinend eine Grammatik beziehungsweise ein Lexikon entwickelt werden kann, die bezogen auf die erwartbaren Eingaben als umfassend bezeichnet werden kann.

(3.22)



Nominalgruppe

Der Bereich der Nominalgruppe enthält PS-Regeln sowohl für Genitiv-Attribute als auch für Präpositional-Adjunkte, wobei letztere als Elemente der Wert-Liste des ADJUNCT-Attributs auftreten. Das Konzept von Feature-Struktur-Listen als Werte wurde für diese Zwecke theoretisch in der LFG eingeführt und auch in meiner Implementation realisiert. Genitiv-Attribute werden als Werte eines eigenen Attributs eingeführt, da sie im Gegensatz zu Adjektiven und PPn als Modifizierer sinnvoll nur einmal auftreten können.

Schwierigkeiten sind Kombinationen mit Namen wie zum Beispiel *Herr Meyer*, *Wilhelm Meyer* oder *Wilhelm-Straße*. Für jeden dieser Fälle wurde eine eigene Regel in die Grammatik mit aufgenommen, die sich durch ein spezielles POS (`pn` für Eigennamen) oder durch eine spezielle Belegung des NTYPE-Features auszeichnet. Die Regel für den vollständigen Eigennamen wie *Wilhelm Meyer* lautet zum Beispiel:

(3.23)
```
npbar(_,_,_) ---> [kleene(pn(adjunct,_,_)),
                   pn(_,_,_)].
```

In diesem Fall wird der letzte Name einer solchen Gruppe als Kopf gesehen, dem beliebig viele Eigennamen vorangehen können. Bei der Regel für Kombinationen mit *Frau*, *Herr* oder Titeln wie *Doktor* wird dagegen das erste Element als Kopf kodiert. Das Konzept gilt auch für Kombinationen wie *das Krankenhaus Bethel*. Umgekehrtes gilt für Kombinationen wie *Wilhelm-Straße*, in der *Straße* den Kopf darstellt.

Koordinierte NPn wurden nur als ganze NPn aufgenommen, das heißt, elliptische Fälle sind nicht modelliert (siehe Beispiel 3.19). Bei den im Heringer-Korpus enthaltenen koordinierten NPn handelt es sich fast ausschließlich um zwei vollständige NPn, wie das folgende Beispiel zeigt.

(3.24) *Es entstehen erneut Nachdenken über ihre Natur und ihre Zukunft.

Die Grammatik enthält außerdem Regeln für die Analyse von NPn, die nur aus einem Artikel oder einem Pronomen (Ich sehe *alles*) bestehen. Schließlich werden auch einfache Relativsätze erkannt, die manchmal von Lernern verwendet wurden.

(3.25) Ich habe das Auto$_i$ gesehen, das$_i$ gestohlen wurde.

Konstruktionen mit entfernten Relativsätzen wie in Beispiel (3.25) wurden nicht in die Grammatik aufgenommen; sie tauchen in den von mir in der Evaluation untersuchten Sätzen nicht auf.

Verbalgruppe

Die Verbalgruppe umfasst alle wichtigen Arten von Komplementen wie direkte und indirekte Objekte, komplexe Komplemente wie zum Beispiel dass- beziehungsweise Fragesätze, Präpositionalkomplemente, Infinitivkonstruktionen sowie Regeln für Partikelverben. Auch topikalisierte Sätze sind ohne größere Beschränkungen möglich. Im Unterschied zu den üblichen Annahmen in der LFG-Literatur wirkt sich eine Topikalsierung in meiner Modellierung aber nicht auf die F-Struktur aus; ein Feature wie zum Beispiel `TOPIC` wird nicht verwendet.

Eine der Übungen (Wegbeschreibung) ist thematisch so gestaltet, dass Imperative geeignete Reaktionen des Lerners darstellen. Dem virtuellen Gesprächspartner könnte zum Beispiel in der Übung empfohlen werden: „Gehen Sie die Bahnhofstraße entlang!“ Eine vereinfachte Regel für Imperativformen hat die folgende Form.

(3.26)
```
s(_,_,_) --->
              [opt([part(_,[parttype==quest|_],_)]),
                 v(_,[modus=imp,mood=fin|_],_),
               opt([np(subj,[subj=[prontype==sie|_]|_],_)]),
               opt([vc(_,_,_)]),
               opt([v(_,[vtype==trenn|_],_)])].
```

Damit lassen sich ein mögliches Subjekt (`np(subj, ...)`), weitere Komplemente (`vc(_,_,_)`) und schließlich abgetrennte Verbpartikel (`v(_,[vtype ...])`) beschreiben. Das in der PS-Regel aufgeführte Partikel `part` kann als „Bitte [gehen Sie ...]“ realisiert werden.

Zweitens muss der Bereich der Kopula-Konstruktionen hervorgehoben werden. Kopula-Konstruktionen werden auf der Basis einer eigenen POS (vcop) durch einen eigenen Satz an Regeln erfasst, da die mit Hilfe von Kopula realisierbaren Konstruktionen wie bekannt erheblich von den sonstigen Komplementtypen abweichen.

In der hier realisierten Grammatik wird wie allgemein üblich in die 3 Tempora [TENSE = fut], [TENSE = pres] und [TENSE = past] (Futur, Präsens und Präteritum) unterschieden. Diese Werte stammen entweder aus den Lexikoneinträgen der finiten Vollverben oder werden durch die Auxiliare beigesteuert. Die Kombination mit dem Attribut PERF erlaubt dann die Kategorisierung in die unterschiedlichen Tempora. Beispielsweise enthält ein Perfektsatz die Attribut-Wert-Paare [TENSE = pres] und [PERF = +]. Auf dieser Basis lassen sich leicht Erläuterungen für den Nutzer des Systems ableiten.

Präpositionalgruppe

Die Präpositionalgruppe umfasst neben „normalen“ auch verschmolzene Präpositionen, die einerseits einige Features zur Verfügung stellen, die normalerweise vom Artikel geliefert werden und andererseits ist nur eine NP als Komplement zur Präposition zulässig, die keinen Artikel enthält. Postpositionen, wie zum Beispiel *die Straße entlang*, sind versuchsweise integriert worden und werden auch als solche analysiert.

PPn werden in der F-Struktur in einem ADJUNCT-Feature zusammengefasst, wenn nicht das Verb dafür subkategorisiert. Für den Fall, dass ein Verb eine bestimmte Präposition verlangt, wird wie in der LFG üblich ein Feature eingeführt, das einerseits die thematische Rolle der Präposition charakterisiert und andererseits einen komplexen Wert hat, der alle Informationen der PP enthält. Die folgende vereinfachte F-Struktur für den Satz „Ich wohne in Berlin.“, für den angenommen wird, dass das Verb *wohnen* für eine PP subkategorisiert, macht dies deutlich.

(3.27)
$$\left[\begin{array}{ll} \text{PRED} & = \text{‘WOHNEN <SUBJ,IN>’} \\ \text{TENSE} & = \text{pres} \\ \text{SUBJ} & = \left[\begin{array}{ll} \text{PRED} & = \text{‘ICH’} \\ \text{PERS} & = 1 \end{array}\right] \\ \text{IN} & = \left[\begin{array}{ll} \text{PRED} & = \text{‘IN <OBJ>’} \\ \text{PCASE} & = \text{IN} \\ \text{OBJ} & = \left[\begin{array}{ll} \text{PRED} & = \text{‘BERLIN’} \end{array}\right] \end{array}\right] \end{array}\right]$$

In der LFG wurde für diese Strukturierung ein Mechanismus entwickelt, der es erlaubt, den Wert des Features PCASE, das aus dem Lexikoneintrag der Präposition stammt, gleichzeitig als Feature für die gesamte PP einzusetzen. Es muss damit in der entsprechenden PP-Regel nicht jede mögliche subkategorisierbare Präposition vorgesehen werden, sondern nur zwischen Adjunkten und subkategorisierten PPn unterschieden werden. Hier gelten offensichtlich auch die Vollständigkeits- und Kohärenzprinzipien, da das Element IN der Subkat.-Liste auch als Feature auftritt. Schließlich können in dieser Kategorie Pronominaladverbien wie *darüber* erwähnt werden, für die eine Analyse als PPn realisiert wurde.

Adjektive/Adverbien

Für die Modellierung von Adjektiven und Adverbien wurde eine sehr grobe Einteilung vorgenommen, die pragmatischen Grundsätzen folgt. Die so genannten adverbialen Adjektive (Eisenberg, 1999, S. 220ff) wurden von mir in die Klasse der Adverbien aufgenommen, obwohl verschiedene, vor allem semantische Gründe dagegen sprechen. Die adjektivischen Adverbien erhalten zusätzlich ein Attribut, das sie als solche auszeichnet. Damit können dann die syntaktischen Eigenschaften der Klasse modelliert werden. Auch die „echten“ Adverbien sind als solche markiert, sodass eine Unterscheidung in syntaktischer Hinsicht möglich ist. Das flektierte Adjektiv kommt nur in der Position vor dem Substantiv vor und wird wie auch die PPn als Element unter dem Attribut ADJUNCT zusammengefasst.

Allgemeiner gesehen lässt sich an dieser Aufstellung der unterschiedlichen Bereiche der Grammatik erkennen, dass die von mir betrachteten, authentischen Sätze, die die Grundlage für die Modellierung der Grammatik bildeten, eher einfache morphosyntaktische Konstruktionen umfassen. Das bedeutet natürlich nicht, dass eine „einfache“ Grammatik schon für die Analyse von freien Lernereingaben genügen würde, da auch für den Fall einer komplexeren, aber morphosyntaktisch korrekten Eingabe das System nicht einen Fehler identifizieren darf, sondern nach Möglichkeit auch dann eine geeignete Strukturbeschreibung liefern sollte. Aber es demonstriert, dass womöglich die morphosyntaktische Analyse von Lerneräußerungen in einem ICALL-System einen Bereich darstellt, in dem bereits mit eingeschränkter Abdeckung der Grammatik umfassende Übungstypen realisiert werden können.

3.5 Zusammenfassung

Zu Beginn dieses Kapitels wurden drei Anforderungen an eine Grammatiktheorie, die in einem ICALL-System verwendet werden soll, formuliert: Erstens sollte die Grammatiktheorie eine gewisse linguistische Adäquatheit

besitzen, zweitens muss die Theorie formal fundiert sein, damit sie implementierbar ist, und drittens sollte die Theorie Analysen liefern können, die in für Lerner verständliche Meldungen über morphosyntaktische Sachverhalte umgesetzt werden können. Im Anschluss an die Formulierung der Anforderungen wurde die Theorie der LFG vorgestellt, um auf dieser Basis zeigen zu können, dass die LFG zu den Grammatiktheorien gehört, die diesen Anforderungen genügen. Ein besonderer Schwerpunkt ist dabei die Möglichkeit der Generierung von lernerorientierten Analyseergebnissen. Im Einzelnen ergeben sich folgende Resultate:

1. Die LFG gehört zu den sehr häufig verwendeten Grammatiktheorien, wenn es um die praktische Modellierung von sprachlichen Phänomenen in computerlinguistischen Anwendungen geht. Darüberhinaus hat sich die LFG vor allem auch aus theoretischer und – spezifischer gesehen – aus typologischer Sicht als geeignet erwiesen, für eine Reihe sehr unterschiedlicher Sprachen linguistisch adäquate Beschreibungen liefern zu können.

2. Die Theorie ist von Beginn an formal fundiert gewesen, was auch zu dem angeführten, intensiven Einsatz in sprachverarbeitenden Systemen geführt hat. Damit stellt sie eine hervorragende Grundlage für die Nutzung in einem ICALL-System dar.

3. Schließlich kann ein Parser auf der Basis von LFG-artigen Strukturen Beschreibungen liefern, die eine hinreichende Ähnlichkeit mit Beschreibungen aus Lernergrammatiken aufweisen. Dabei zeigen sich beispielhaft mindestens in folgenden Bereichen Korrespondenzen:

   - Die Perspektive auf das Lexikon als ein wesentliches Element, das nicht ausschließlich inkompatible, lexembezogene Informationen enthält, wird auch in Lernergrammatiken deutlich.
   - Wie auch in Lernergrammatiken werden Morphologie und Syntax in einen bestimmten Sinne strikt getrennt, nämlich dass zur Analyse eines Satzes nur ganze Wörter betrachtet werden. Damit ist also die morphologische Komplexität einer Sprache losgelöst von der Konstruktion eines Satzes.[13]
   - Die Wortstellung kann zumindest für das Deutsche so gestaltet werden, dass sie weitgehend den Darstellungen in Lernergrammtiken entspricht, die sich in erster Linie an einem topologischem Modell orientieren.

[13] Dieses gilt zumindest für die deutsche Sprache, in deren Beschreibung grammatische Funktionen als Annotationen an PS-Regeln realisiert sind. Bei einer nicht konfigurationalen Sprache wie Malayalam kann die angesprochene Trennung nicht aufrecht erhalten werden.

- Die funktionale Perspektive stellt das Grundgerüst für den Aufbau der F-Struktur dar und findet sich als eine entscheidende Betrachtungsweise auch in Lernergrammatiken wieder.

Wenn die in der LFG verwendeten Strukturen zum großen Teil mit denen in Lernergrammatiken korrespondieren, sollte eine Erläuterung für den Lerner auf der Basis des Analyseergebnisses leicht verständlich sein. Offensichtlich sollte aber auch sein, dass nicht möglicherweise auch andere Grammatiktheorien die relevanten Eigenschaften besitzen und dass eine weitergehende Bestätigung der Hypothese von der Eignung der LFG in diesem Kontext kaum empirisch begründet werden kann.

Im Anschluss wurde der Umfang der in Promis$D$ modellierten Grammatik erläutert. Hier hat sich unter anderem gezeigt, dass bestimmte syntaktische Konstruktionen nicht in den betrachteten Sätzen auftreten. Diese Tatsache legt nahe, dass auch mit Hilfe einer eingeschränkten Grammatik schon die wesentlichen Lernereingaben analysiert werden können. Aufwändige Konstruktionen mit multiplen Infinitiven in einem Satz oder Partizipialkonstruktionen in einer NP kommen in den zu Grunde gelegten Sätzen der Evaluation nicht vor. Andererseits sind in Promis$D$ alle wesentlichen Eigenschaften des Formalismus umgesetzt, sodass beispielsweise Adjunkte als Menge in einer F-Struktur adäquat repräsentiert werden können.

Meines Erachtens bietet sich die LFG also zum Einsatz als Analysegrammatik in einem ICALL-System an, da sie einerseits so strukturiert ist, dass sie sich leicht in einer Implementation umsetzen lässt und sie andererseits mit den Konzepten in Lernergrammatiken in gewisser Hinsicht korrespondiert und daher für Rückmeldungen zu Lernereingaben und Darstellungen genutzt werden kann.

Kapitel 4

Allgemeine Systembeschreibung

Das folgende Kapitel liefert in erster Linie eine Programmbeschreibung, die sich auf die Sicht eines Nutzers konzentriert. Zunächst erfolgt eine generelle Vorstellung des Übungsaufbaus, ohne dass mögliche Reaktionen auf fehlerhafte Eingaben berücksichtigt werden, und im Anschluss wird in Abschnitt 4.2 erstens gezeigt, zu welchen allgemeinen Fehlertypen Rückmeldungen erfolgen können, und zweitens, in welcher Form die Rückmeldungen dem Lerner präsentiert werden. Einerseits wird demonstriert, wie die in Kapitel 2 vorgestellten allgemeinen Anforderungen an ein tutoriell orientiertes ICALL-System umgesetzt werden können und andererseits wird gezeigt, welche Möglichkeiten der Fehlermeldung sich aus der in Promis*D* realisierten Programmstruktur und der computerlinguistischen Eingabeanalyse ergeben. Zum Abschluss des Kapitels (Abschnitt 4.4) wird auf die Architektur des Systems eingegangen, die einen kurzen Überblick über die Interaktion der verschiedenen Programmmodule und über die Verarbeitung einer Eingabe bietet. Die Einordnung der in den darauf folgenden Kapiteln vorgestellten Module in den Gesamtrahmen wird damit hoffentlich erleichtert.

4.1 Benutzerschnittstelle

Zu Beginn sollen noch einmal einige Ergebnisse aus Kapitel 2 rekapituliert werden, um die Verbindung dieser Aspekte und der im Anschluss folgenden Darstellung der Benutzerschnittstelle zu erleichtern. Das Üben von Dialogen dient im Unterricht in erster Linie der Förderung der kommunikativen Fähigkeiten. Die Erlangung der kommunikativen Kompetenz wiederum stellt eines der wesentlichen Ziele des Fremdsprachenunterrichts dar. Um diesem Ziel näher zu kommen, ist das Programm so konzipiert, dass dem Lerner Dialogsituationen präsentiert werden, die ein selbstständiges und aktives Formulieren des Lerners erfordern. In diesen Frage-Antwort-Dialogen wird

der Nutzer aufgefordert, in verschiedenen Situationen auf die Fragen des simulierten Dialogpartners zu antworten. Beispielhaft sind drei Dialogübungen modelliert worden: eine persönliche Vorstellung, ein Unfallbericht sowie eine Wegbeschreibung. Anregungen zu ähnlichen Dialogübungen finden sich auch in einigen Sprachlehrwerken wie zum Beispiel Themen (1997, S. 93ff), Deutsch aktiv (1986, S. 92ff). Dieser Übungstyp unterscheidet sich von denen traditioneller Lern- und Übungsprogrammen sowie tutoriell orientierter Multimedia-Anwendungen in entscheidender Hinsicht. In diesen Programmen besteht üblicherweise die Interaktion in einer Aufgabenstellung maximal mit der Auswahl aus einer Liste von vorgefertigten Antworten oder die Aufgabenstellung ist sehr stark eingeschränkt („Füge die richtige Form des Hilfsverbs 'haben' ein"). In Promis*D* dagegen ist der Lerner gefordert, sein eigenes Wissen zum Gebrauch der Sprache zu aktivieren. Die Leistung des Lerners liegt einerseits im Verständnis einer konkreten Frage und andererseits in der vollständig freien Produktion eines Antwortsatzes. Das System unterstützt den Lerner dabei, indem ein plausibler Kontext für den Dialog und eine präzise Reaktion aufgrund einer ausführlichen computerlinguistischen Analyse geliefert werden kann.

4.1.1 Grundlegendes zur Aufgabenstellung

Beim Aufruf des Programms wird der Benutzer zu Beginn gefragt, ob ein bestimmtes Verzeichnis zur Aufzeichnung der Übungsdaten verwendet werden soll. Das Programm legt dann gegebenenfalls ein Verzeichnis an, in dem Eingaben des Lerners einschließlich der Fehlermeldungen gespeichert werden. Diese können dann zur Nachbearbeitung der Aufgaben, zum Beispiel zur genaueren Sprachstandsanalyse verwendet werden (siehe Seite 97).

Hat sich der Lerner für eine Dialogübung entschieden, wird zuerst in einem seperaten Fenster die Aufgabenstellung zum Teil auch multimedial erläutert. Im Fall der Übung „Unfallbericht" wird zum Beispiel comicartig gezeigt, wie ein Auto versucht, einem Hund auf der Straße auszuweichen und in Folge dessen gegen eine Ampel fährt. Am Ende dieser Präsentation wird eine Telefonzelle gezeigt, von der aus der Lerner die Polizei „anruft" und „den Unfall meldet". Verwiesen sei hier noch einmal auf das Studienprojekt PROMISE (Bauer et al., 1994; John, 1994), das die Vorlage für diese Form der Aufgabenstellung lieferte (siehe Seite 5).

Nun beginnt die eigentliche Dialogübung, das heißt, ein Polizist meldet sich und der Lerner muss eine Reaktion zeigen beziehungsweise einen Satz mit Hilfe der Tastatur eingeben. Die folgende Abbildung 4.1 zeigt diese Situation.

Nach und nach müssen nun verschiedene Fragen zum Unfall, zu Verletzten sowie zum Namen und zur Adresse des Anrufers beantwortet werden. Als Beispiel soll hier eine mögliche Antwort auf die erste Frage gezeigt werden.

In Abbildung 4.2 kann man erkennen, dass das den Polizisten simulie-

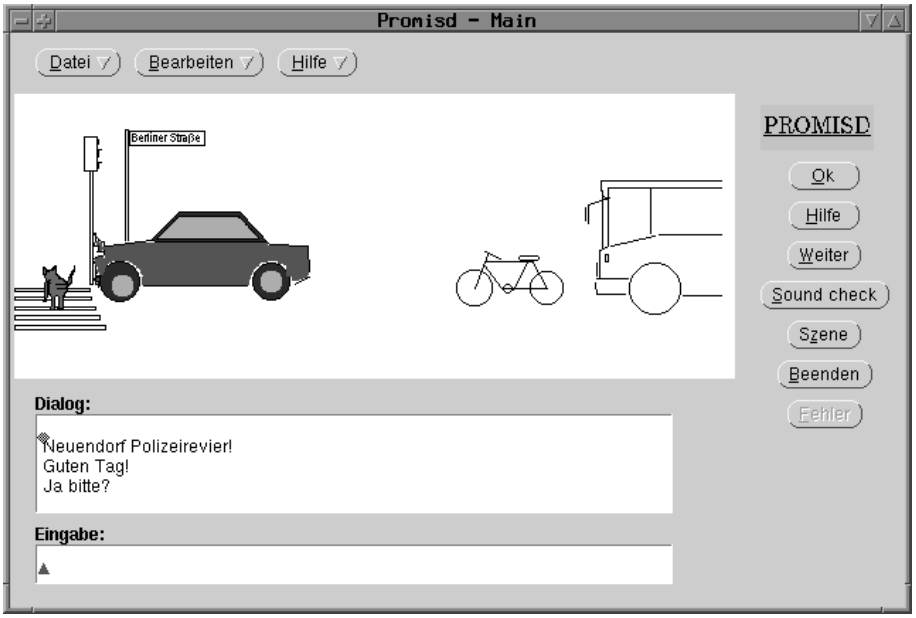


Abbildung 4.1: Start des Unfall-Dialogs

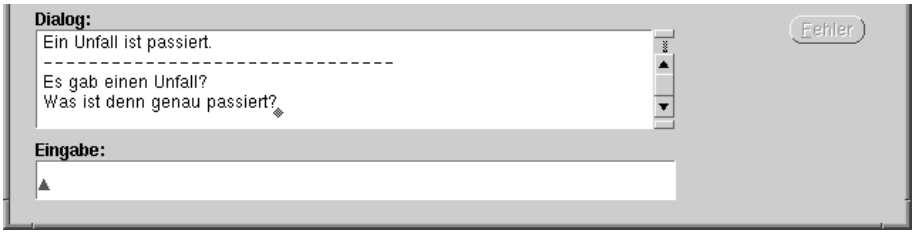


Abbildung 4.2: Beispielantwort als erste Reaktion

rende System zuerst eine Reaktion auf die Lernereingabe „Ein Unfall ist passiert.“ ausgibt, nämlich eine Bestätigung der Aussage, die wiederum für den Lerner ein Zeichen sein kann, dass seine Eingabe wie intendiert interpretiert worden ist. Anschließend fragt der simulierte Polizist weiter nach, falls die Eingabe als Antwort interpretierbar war. In diesem Fall ist eine Präzisierung der Unfallbeschreibung notwendig; das System erwartet also eine Aussage über die Kollision eines Autos mit einer Ampel. Das System ist so flexibel ausgelegt, dass zu Beginn auch wie in einem echten Telefongespräch der eigene Name genannt werden kann oder der Unfallhergang sofort genauer beschrieben werden kann, solange nicht mehr als ein Satz eingegeben wird. Die Funktionsweise des Dialogmoduls aus technischer Sicht wird in Abschnitt 6.1 näher erläutert.

Wenn der Dialog erfolgreich bearbeitet worden ist, ergibt sich mit der folgenden Reaktion in Abbildung 4.3 des „Polizisten“ das Ende des Dialogs.

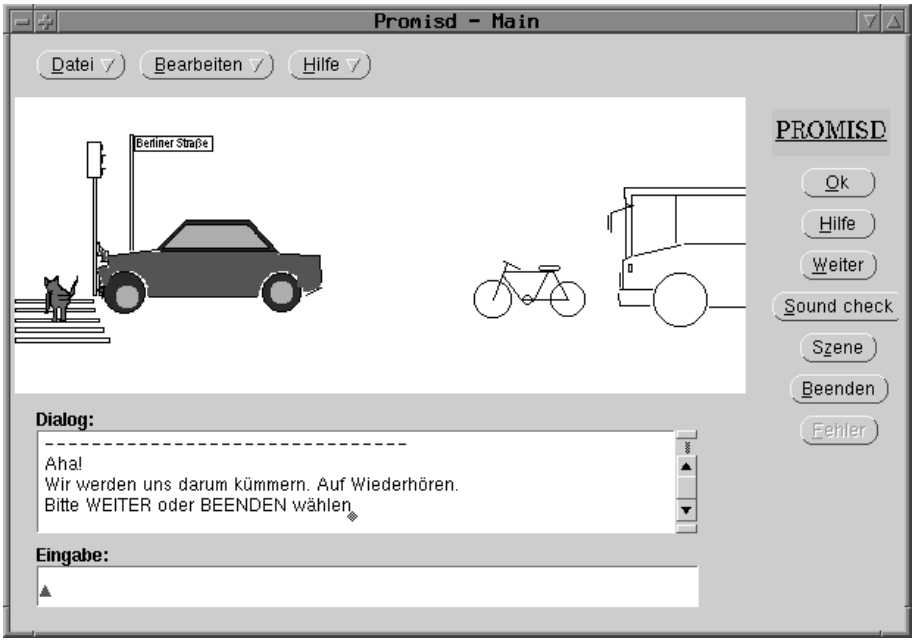


Abbildung 4.3: Ende des Unfall-Dialogs

Auch diese Formulierung ist nicht generisch für alle Dialoge, sondern stammt aus der Wissensbasis zur Situation der Unfallmeldung. Der Lerner kann nun einen weiteren Dialog bearbeiten oder die Bearbeitung dieses Aufgabentyps beenden.

4.1.2 Abbruch des Dialogs

Die Dialogführung ist so gestaltet, dass der Dialog nach zwei nicht interpretierbaren Eingaben abbricht.

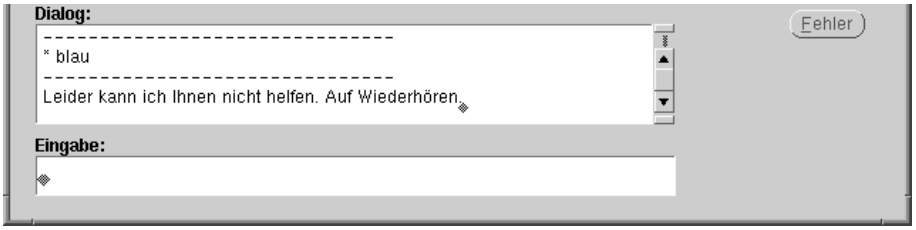


Abbildung 4.4: Abbruch des Dialogs

Im Screenshot in Abbildung 4.4 kann man erkennen, dass eine unsinnige Eingabe vom Lerner gemacht worden ist, die der simulierte Dialogpartner mit dem Abbruch des Dialogs quittiert. Diese Reaktion könnte so auch von einem echten Gesprächspartner kommen, das heißt, falls der Lerner sich an-

scheinend unkooperativ zeigt oder auch nur die Grenzen des Systems testen möchte, kann mit einem derartigen Abbruch eine sehr plausible Reaktion vermittelt werden.

An dieser Stelle kann angemerkt werden, dass das System nur in der Lage ist, Eingaben zu verarbeiten, die mit Hilfe der Wissensbasis über eine bestimmte Unfallsituation interpretiert werden können (siehe Abschnitt 6.1). Ein Dialogabbruch wie oben dargestellt erfolgt also auch, wenn der Nutzer über einen Unfall berichtet, der nicht dargestellt worden ist und damit nicht in der Wissensbasis modelliert worden ist, oder der Nutzer zwar eine morphosyntaktisch korrekte Eingabe macht, aber gar nicht über einen Unfall berichtet.

4.1.3 Hilfefunktionen

Obwohl das Ziel des Systems ja eigentlich darin besteht, den Lerner zu selbstständigen Aktionen aufzufordern, wurde doch aufgrund der Testläufe eine Hilfe zur aktuellen Aufgabenstellung integriert. In einigen Fällen waren die Lerner sehr unsicher, was sie auf eine Frage antworten sollten und haben daraufhin nach einer Hilfe gesucht. Hier zeigt sich, dass Promis$D$ auch ein Computerprogramm ist, von dem eine bestimmte allgemeine Funktionalität erwartet wird. Zu jeder möglichen Frage des Systems befindet sich in der Wissensbasis eine beispielhafte Antwort auf die jeweilige Frage. Wenn der Lerner auf den Hilfe-Knopf klickt, wird dieser Satz wie in Abbildung 4.5 dargestellt einmal angezeigt. Zur Fortsetzung muss der Lerner dieses Hilfefenster aber wieder entfernen und den Satz aus der Erinnerung eingeben. Dem Lerner wird auf diese Weise eine kontextsensitive Hilfe geboten, die eine spezielle Hilfestellung für den momentanen Stand der Abarbeitung des Dialogs bietet.

Einerseits wird dem Lerner auf diese Weise eine Anregung präsentiert, wie der Dialog weitergeführt werden könnte und andererseits wird so die Funktionalität des Computerprogramms erhöht. Da der Satz aber nicht abgetippt werden kann, sondern vor der Eingabe wieder entfernt werden muss, wird eine indirekte Hilfe modelliert, die die Konzentration des Lerners auf den Inhalt der Äußerung lenkt. Anstelle dieser „vollständigen“ Hilfe lässt sich auch eine gestufte Hilfe denken, die dem Lerner im ersten Versuch nur beschränkt Hilfestellung gibt. Weitere Ausführungen dazu folgen in Abschnitt 4.3.

Zusätzlich zur Hilfe durch einen Beispielsatz lässt sich außerdem eine Klangdatei aufrufen, in der die Ausgabe des Systems in gesprochener Form gespeichert ist („Sound check“-Knopf in Abbildung 4.5). Auf diese Weise lässt sich ein weiterer Wahrnehmungskanal ansprechen, der den Lerner zusätzlich bei der Bewältigung der aktuellen Aufgabe unterstützen sollte.[1]

[1] Offensichtlich sind weitere multimediale Erweiterungen denkbar, die aber nicht realisiert worden sind. Es bietet sich so zum Beispiel der Einsatz eines so genannten Text-

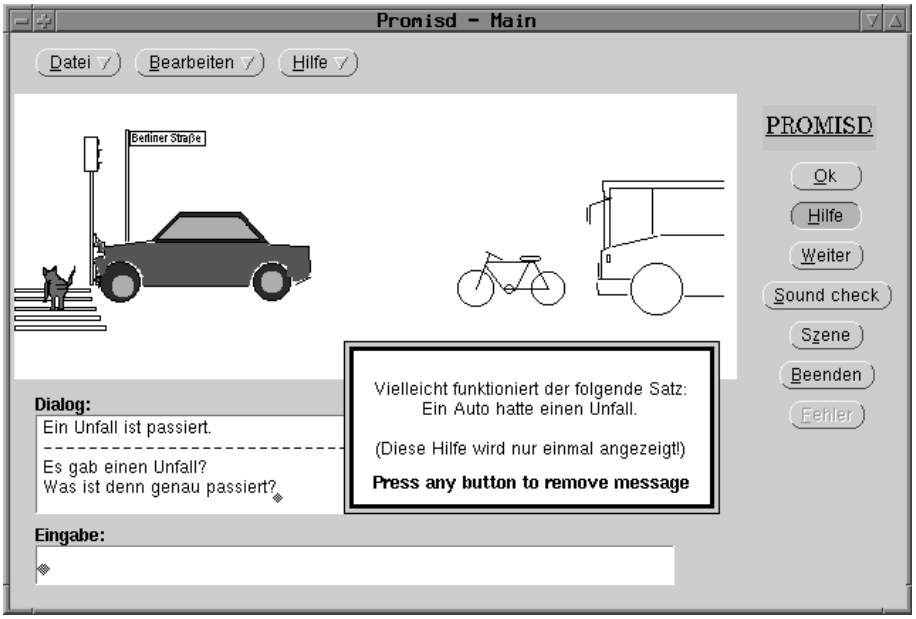


Abbildung 4.5: Hilfe zum Dialog

Eine weitere Hilfe, die das Verständnis des Lerners der vom Polizisten produzierten Äußerungen unterstützen kann, ist die Möglichkeit, Wörter im Dialog-Fenster anzuklicken und so einen aus dem Systemlexikon heraus generierten Lexikoneintrag präsentiert zu bekommen. In Abbildung 4.6 ist beispielsweise das Wort „Unfall“ angeklickt worden.

Das zusätzliche Fenster präsentiert die Informationen aus dem Systemlexikon, die für einen Lerner relevant sein könnten. Wie zu erkennen ist, sind die Attribute direkt in Terme übersetzt, die für Sprachenlerner mit grundlegenden linguistischen Kenntnissen verständlich sein sollten.[2] Vor dem Kasus-Attribut befindet sich ein Minus-Zeichen, das in ähnlicher Form auch im Systemlexikon enthalten ist. Dieses Feature hat die Bedeutung, dass das Wort „Unfall“ in Verbindung mit Numerus Singular alle Kasus außer Genitiv haben kann. Die Information zu diesem Attribut, dass die Form „Unfall“ für alle anderen Kasus gilt, ist nicht im Lexikon enthalten und kann somit nicht direkt an den Lerner ausgegeben werden. Sehr leicht integrierbar wäre hier eine zusätzliche Funktion, die für das Deutsche die „fehlenden Kasus“

To-Speech-Moduls (TTS) an, um zusätzlich die Eingaben des Lerners hörbar zu machen. Damit wäre eine umfassende multimediale Aufbereitung einer Dialogübung erreicht, wenn man davon ausgeht, dass TTS-Systeme heutzutage schon für ein Fremdsprachenlernszenario akzeptablen Output leisten.

[2]Falls das nicht der Fall ist, sollte das System bis zu einem gewissen Grad in der Lage sein, sich dem Vorwissen eines Lerners anzupassen und daraufhin die Präsentation von Hilfetexten und Fehlermeldungen zu verändern. Nähere Erläuterungen dazu erfolgen in Abschnitt 5.5.

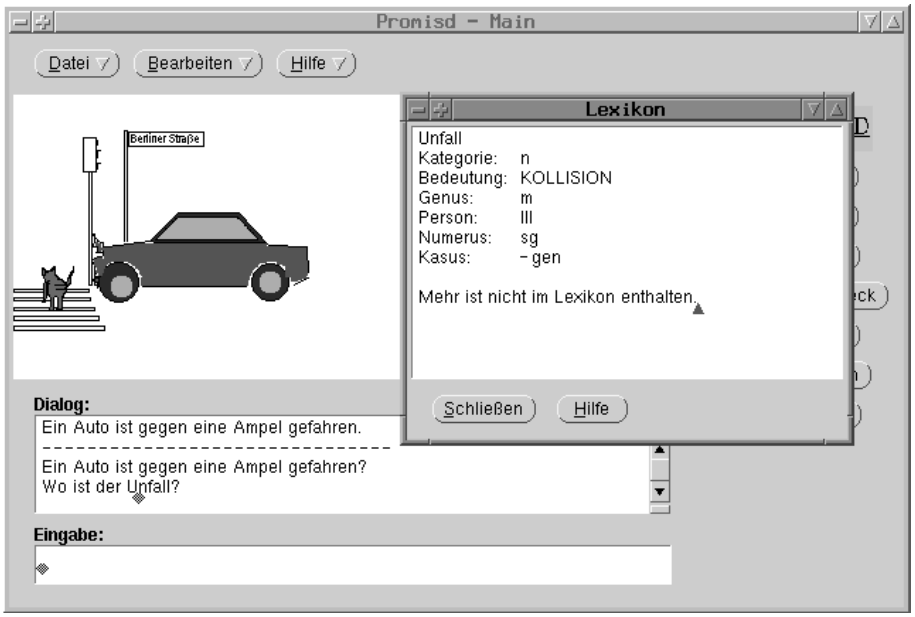


Abbildung 4.6: Nachschlagen im Systemlexikon

ermittelt und dann als zusätzlichen Hilfetext ausgibt. Das Attribut „Bedeutung“ stellt die Übersetzung des in der LFG verwendeten PRED-Attributs dar. Hier sollte für den Lerner eine Erweiterung des Lexikons mit Bedeutungsangaben erfolgen (siehe Abschnitt 4.3).

Eingehender soll die Interaktion mit dem System für den allgemeinen Fall nicht betrachtet werden. Im folgenden Unterkapitel wird nun der spezielle Fall einer fehlerhaften Eingabe beleuchtet, der ja den wesentlichen Aspekt von Promis$D$ ausmacht.

4.2 Fehlermeldungen aus Nutzersicht

Als einer der Schwerpunkte kann das System auf fehlerhafte Eingaben in den Bereichen Orthografie, Morphosyntax, Semantik und Dialogstruktur reagieren. Wie in Abschnitt 4.4 (Seite 101) erläutert, wird jeder dieser Bereiche sequentiell abgearbeitet, sodass Fehlermeldungen immer auch nur in dieser Reihenfolge auftreten können.

Wie im vorhergehenden Unterkapitel wird auch in diesem Abschnitt nur dargestellt, wie sich das System für den Benutzer verhält. In jedem Unterabschnitt ist ein Verweis auf die entsprechenden Abschnitte mit Erläuterungen zu den technischen Funktionsweisen gegeben.

4.2.1 Orthografie

Abbildung 4.7 zeigt, wie sich das Orthografie-Modul im Fall einer orthografisch fehlerhaften Eingabe dem Nutzer präsentiert. Die Korrekturkandidaten – möglicherweise auch von mehreren Wörtern – werden in einem zusätzlichen Fenster angezeigt, das entfernt werden muss, bevor die eigentliche Korrektur vom Lerner vorgenommen werden kann.

Abbildung 4.7: Meldung eines Orthografiefehlers

Von den Testpersonen wurde an dieser Stelle gewünscht, dass man die Wörter durch einfaches Klicken in den Satz einfügen kann. Durch das erneute Eingeben beziehungsweise die manuelle Korrektur, wie sie hier realisiert ist, wird aber der Umgang mit der Sprache verstärkt. Unter Umständen kann so eine stärkere Reflexion über die Äußerung erreicht werden und die Schreibweise kann sich durch das eigene Schreiben besser einprägen, als wenn der Lerner mit einem Klick die Korrektur geleistet hätte.
Bei großgeschriebenen Wörtern, die nicht im Lexikon enthalten sind, vermutet das System, dass es sich um einen Namen handelt. Der Lerner wird dazu befragt und kann das Wort gegebenenfalls korrigieren. Wenn der Lerner angibt, dass es sich um einen Namen handelt, wird das Wort als Eigenname in das Lexikon aufgenommen und steht für den Zeitraum der aktuellen Sitzung zur Verfügung. Kleingeschriebene Wörter, die vom System nicht im Lexikon gefunden wurden, werden nicht akzeptiert, da der Dialog nicht fortgesetzt werden kann, wenn ein kleingeschriebenes Wort im anschließenden Parsing, das heißt bei der syntaktischen Analyse, nicht erkannt werden kann. Wenn der Parse-Vorgang beginnt, kann also davon ausgegangen werden, dass alle

Wörter bekannt, beziehungsweise im Lexikon vorhanden sind, oder dass es sich um Eigennamen handelt.[3] Die Funktionsweise der Orthografiekontrolle wird in Kapitel 6.2.1 beschrieben.

4.2.2 Morphosyntax

Die syntaktische Analyse eines Satzes und die Identifikation eines möglicherweise enthaltenen Fehlers stellt aus computerlinguistischer Sicht den Schwerpunkt des Programms dar, obwohl sich das nicht unbedingt an der Oberfläche für den Benutzer zeigt. Zwei Fehlerklassen lassen sich präzise ermitteln: Kongruenz- beziehungsweise Rektions- sowie Konstituentenfehler (siehe Kapitel 5). Die Unterscheidung in diese zwei Fehlerklassen ist durch die in der LFG verwendeten Strukturen zur Satzbeschreibung bedingt. Kongruenz- und Rektionsbeziehungen werden wie in Kapitel 3 beschrieben in der F-Struktur kodiert, während die Wortstellung mit Hilfe von PS-Bäumen beschrieben wird. Alle typischen Formen von Kongruenz- sowie Rektionsfehlern werden vom System identifiziert und können dem Lerner erklärt sowie zumeist auch korrigiert werden. Zusätzlich werden auch einige Kategorienfehler, wie zum Beispiel der Typ des Auxiliars, in gleicher Weise behandelt. Bei den Konstituentenfehlern werden in vielen Fällen Auslassungs- und Einfügungsfehler sowie Verdrehungen identifiziert und korrigiert, da es sich bei diesen Fehlertypen um die weitaus häufigsten Konstituentenfehler handelt. Die Erklärung eines Konstituentenfehlers stellt dagegen eine besondere Schwierigkeit dar, die in Abschnitt 5.5 erläutert wird.

Falls der Parser einen Fehler im morphosyntaktischen Bereich identifiziert hat, wird zunächst nur der Fehlerknopf aktiviert, damit die Aufmerksamkeit des Lerners geweckt wird, und der Dialog fortgesetzt, wenn die Eingabe trotz des Fehlers als sinnvolle Reaktion auf die vom System gestellte Frage interpretiert werden kann. Der Satz wird mit einem Sternchen markiert und der Lerner kann das Ergebnis der Fehleranalyse mit einem Klick auf den Fehler-Knopf abrufen.[4] Abbildung 4.8 verdeutlicht dies.

Abbildung 4.8 zeigt die Rückmeldung zu einer fehlerhaften Eingabe, in der ein Rektionsfehler bezüglich des Kasus identifiziert worden ist. In Bezug auf die unterschiedlichen Stufen einer Fehleranalyse (Seite 40) kann eine Erklärung und eine Korrektur geliefert werden. Die Erklärung des Fehlers

[3] Obwohl wünschenswert, ist keine Morphologie-Komponente in das System integriert, die Auskunft über falsch flektierte Formen geben könnte. Unberücksichtigt bleibt hier auch das Problem, dass dem Lerner möglicherweise Korrekturkandidaten präsentiert werden, von denen er einen auswählt, obwohl er das Wort gar nicht gemeint hat und dann diese falsche Wahl nach dem Parsing als Fehler präsentiert bekommt. Offensichtlich ist eine Annäherung an diese Schwierigkeit nur mit großem Aufwand zu erreichen, der den Rahmen dieser Arbeit sprengt.

[4] Eine ähnliche Möglichkeit und ebenso elegant wäre sicher die direkte Markierung eines Wortes oder einer Phrase, wie es zum Beispiel in dem Textverarbeitungsprogramm Word realisiert worden ist.

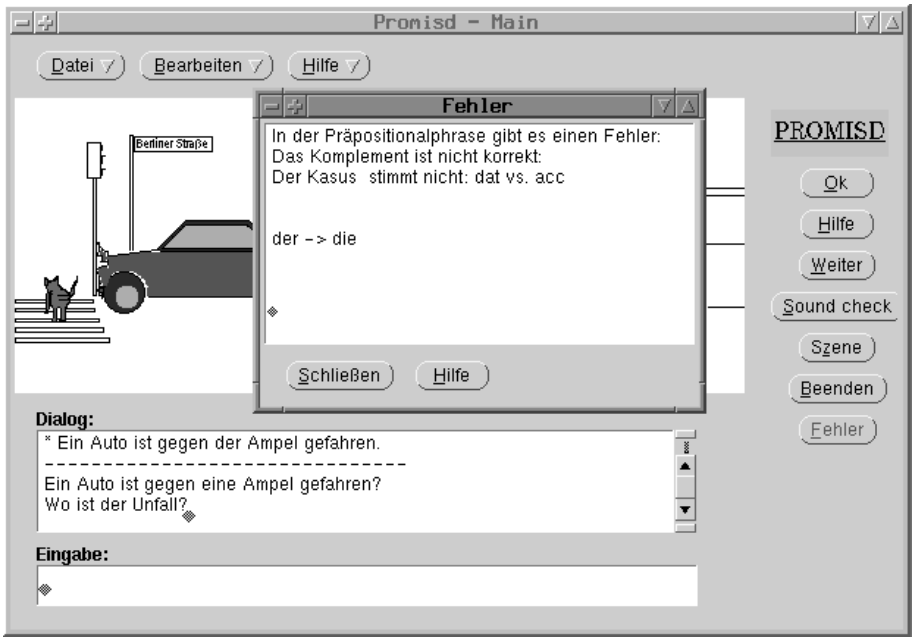


Abbildung 4.8: Meldung eines Rektionsfehlers

lautet, dass der Kasus des Präpositionalobjekts nicht mit der Forderung der Präposition übereinstimmt. Außerdem stellt die Abbildung exemplarisch die Möglichkeit der Korrektur einer Eingabe als Rückmeldung an den Lerner dar. Das Wort *der* der Lernereingabe müsste zu *die* korrigiert werden, um einen korrekten Satz zu produzieren.

In einer älteren Version des Programms wurde bei dieser Art des Fehlers der Dialog immer unterbrochen und ein Fenster mit der Fehlermeldung eingeblendet. Dieses hat sich als sehr störend für den Arbeits- beziehungsweise Kommunikationsfluss erwiesen, da der Dialog zu oft unterbrochen wurde, ohne dass der simulierte Telefondialog tatsächlich aus Sicht der Authentizität gestört gewesen wäre. Wie in Kapitel 2 ausgeführt, stellt gerade dieser Aspekt eine wünschenswerte Eigenschaft dar: Einerseits kann die Arbeitsform des Dialogs ungehindert weitergeführt werden, auch wenn ein Fehler in der Lernereingabe enthalten war und andererseits wird die Fehlermeldung festgehalten und steht dem Lerner potentiell zur Verfügung. Der Lerner kann daher nun selber entscheiden, ob er sich die Analyse des Fehlers anschauen möchte oder nicht.

Die zweite große Klasse von ungrammatischen Eingaben sind die Konstituentenfehler. Abbildung 4.9 zeigt ein Beispiel, bei dem ein Artikel ausgelassen wurde und lediglich eine Korrektur präsentiert werden kann. Wie schon bei den Kongruenzfehlern wird diese Fehlermeldung nicht sofort angezeigt, sondern erst nach einem Klick auf den Fehler-Knopf. Meldungen mit unterschiedlichen Fehlertypen können beliebig kombiniert werden, wie der

folgende Screenshot in Abbildung 4.10 zeigt.[5]

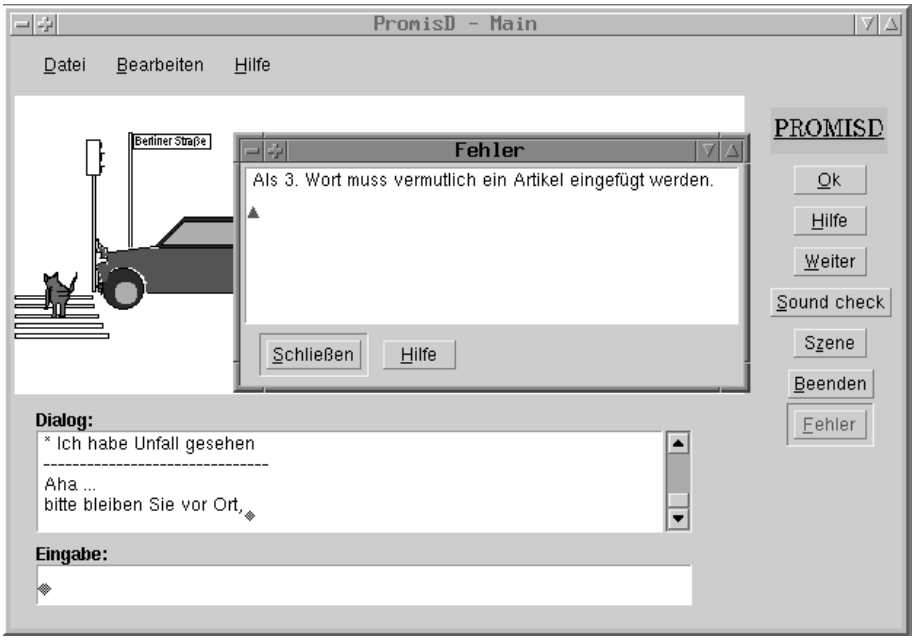


Abbildung 4.9: Meldung eines Konstituentenfehlers

Mit Hilfe dieser Fehlermeldungen kann der Lerner über seine morphosyntaktischen Fehler informiert werden, wozu offensichtlich bei den gezeigten Beispielen eine ausführliche Einführung der grammatischen Begriffe und Konzepte im Unterricht gehört. Die Form der Meldungen sollte sich daher den Kenntnissen des Lerners angepasst variieren lassen, wie exemplarisch in Abschnitt 5.5.4 gezeigt wird. Erläuterungen zu diesem Bereich insgesamt und vor allem zur unterschiedlichen Ausführlichkeit der Meldungen bei bestimmten Fehlertypen (Erklärung vs. Korrektur) erfolgen in Abschnitt 5.5.

4.2.3 Semantik

In Abbildung 4.11 kann man erkennen, wie das System reagiert, falls eine grobe Selektionsverletzung identifiziert worden ist. In der Wissensbasis des Systems ist die Information kodiert, dass „Biologie“ nicht physikalisch greifbar und daher auch nicht „gesehen werden kann“. Grobe Verletzungen dieser Art lassen sich so identifizieren und dem Lerner mitteilen.[6] Ähnlich wie auch im Fall eines morphosyntaktischen Fehlers und im Gegensatz zum Fall eines

[5] Da der Infinitiv-Eintrag zu *haben* keinerlei Angaben zum Subjekt enthält, kann ohne eine Korrektur keine Meldung ausgegeben werden, dass es sich um ein Auxiliar 1. Person Singular handeln muss (siehe Abschnitt 5.5).

[6] Eine weitergehende semantische Analyse wäre nur mit erheblichem Aufwand zu realisieren und ist daher nicht implementiert worden.

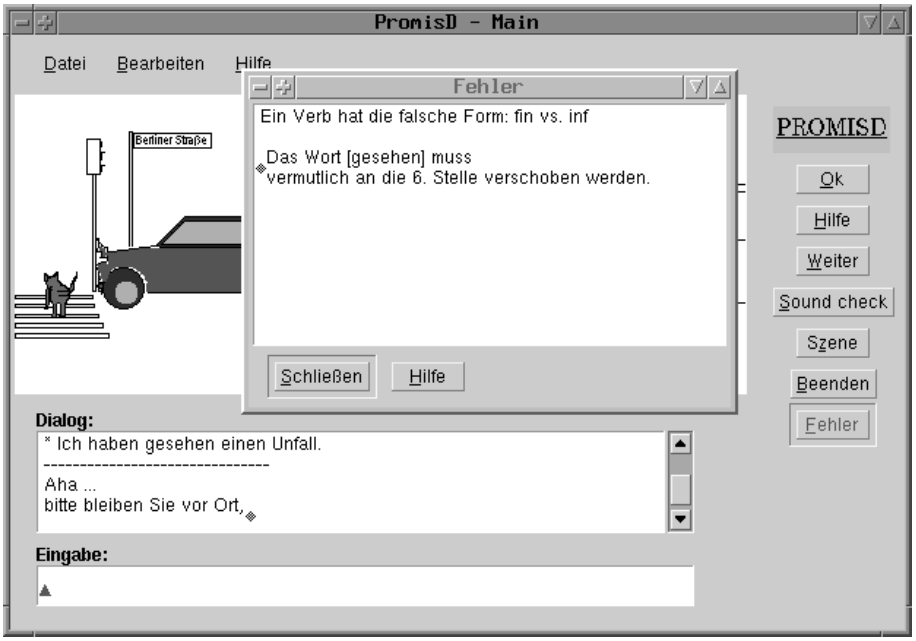


Abbildung 4.10: Meldung von mehreren Fehlern

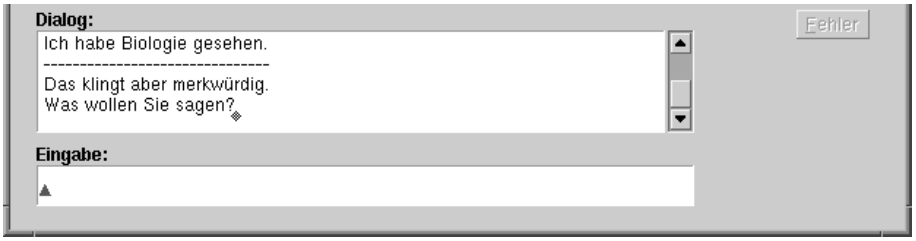


Abbildung 4.11: Meldung eines Bedeutungsfehlers

orthografischen Fehlers wird der Dialog nicht durch ein neu eingeblendetes Fenster oder Ähnliches unterbrochen, sondern das System reagiert in einer Art und Weise, wie das auch von einem realen Gesprächspartner erwartet würde. Der Lerner wird aufgefordert, den Sachverhalt bitte noch einmal zu nennen, etwa so, als ob sich der simulierte Dialogpartner „verhört hätte“. Diese Reaktion des Dialogpartners wirkt sich offensichtlich insofern auf den Dialog aus, als dass der Kommunikationsfluss doch unterbrochen wird und der Lerner zunächst eine Reaktion des Systems interpretieren muss, mit der das Thema des Dialogs zunächst einmal verlassen wird. Eine ausführlichere Beschreibung der Semantikanalyse findet sich in Kapitel 6.2.

4.2.4 Dialogstruktur

Wie im vorhergehenden Unterkapitel 4.1 schon angedeutet, wird nach einer für die Dialog-Komponente nicht interpretierbaren Eingabe eine Nachfrage an den Lerner gerichtet.

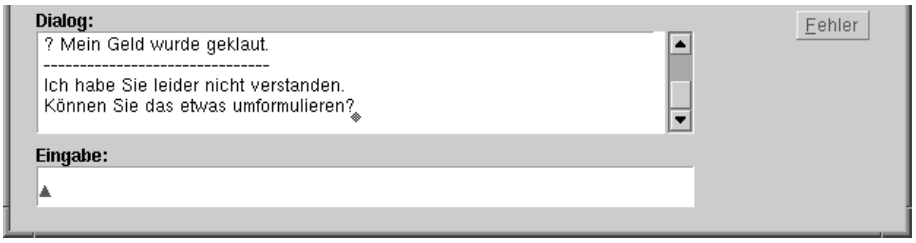


Abbildung 4.12: Meldung eines Dialogfehlers ohne vorhergehende eindeutige Frage

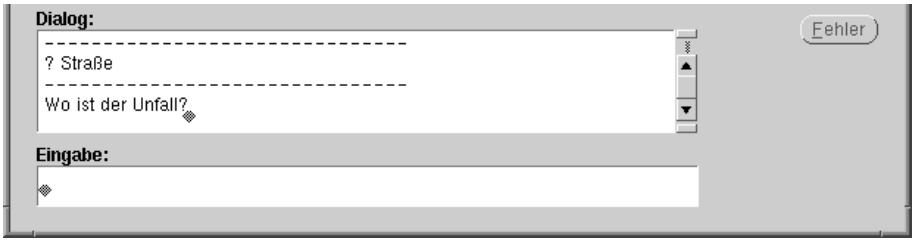


Abbildung 4.13: Meldung eines Dialogfehlers mit eindeutiger Frage

Die beiden Abbildungen 4.12 und 4.13 zeigen, in welcher Weise ein Nachfrage stattfinden kann. Im ersten Fall ist gleich zu Beginn ein zwar syntaktisch korrekter, aber sonst nicht weiter interpretierbarer Satz eingegeben worden. Bei einem Anruf bei der Polizei scheint diese Äußerung zwar Sinn zu machen, aber die Wissensbasis mit den Dialoginhalten enthält keine Informationen über ein Telefongespräch mit der Polizei, in dem es um einen Taschendiebstahl geht, sodass eine entsprechende Reaktion durch den Gesprächspartner erfolgt.

Im zweiten Fall ist auf eine konkrete Frage mit einer Eingabe reagiert worden, die weder mit Hilfe des Parsers noch mit Hilfe der Wissensbasis analysiert beziehungsweise interpretiert werden konnte. Wiederum erfolgt die Reaktion des Systems durch den simulierten Gesprächspartner und nicht durch ein eingeblendetes Fenster und die Frage wird noch einmal in ähnlicher Form wiederholt. Wenn die nächste Antwort des Lerners immer noch keinen Sinn für das System macht, wird, wie in Abbildung 4.4 dargestellt, der Dialog abgebrochen. Die Funktionsweise der Dialogführung wird in Abschnitt 6.1 genauer erläutert.

Wie in Abschnitt 2.3 und zu Beginn dieses Kapitels hervorgehoben, bie-

ten die Übungen eine gewisse Simulation eines Dialogs, der mit dem Rechner geführt wird. Um dem Ziel der kommunikativen Kompetenz näher zu kommen, erfordert die Bearbeitung eines Dialogs in Promis*D* es, eigenständig Sätze zu produzieren, die in keiner Weise durch das Programm vorgegeben sind. Damit wird eine deutliche Erweiterung eines tutoriell orientierten Systems erreicht, die eine wesentlich offenere Übungsform als bisher in einem sinnvollen didaktischen Rahmen erlaubt. Hervorgehoben werden muss hier auch die speziellere Selbstlernsituation, in der insbesondere Übungen zur Kommunikation Schwierigkeiten bereiten. Schließlich wurde gezeigt, dass der Dialog für den Lerner (nahezu) ununterbrochen weiterläuft, soweit das System in der Lage ist, eine Fortsetzung des Dialogs beziehungsweise eine entsprechende Reaktion zu bestimmen. Wenn ein morphosyntaktischer Fehler oder ein Fehler in Bezug auf eine Wissensbasis identifiziert wird, aktiviert das Programm nur den Fehlerknopf, der dann durch einen roten Rand markiert ist, was wiederum mit der auf Seite 40 erwähnten Unterscheidung von Dialogpartner und Sprachtutor einhergeht. Der Lerner kann dann selber entscheiden, ob er sich den Fehler anschauen oder lieber den Dialog weiterführen möchte.

Für den Fall eines erkannten morphosyntaktischen Fehlers wird, wenn angefordert, eine ausführliche Rückmeldung gegeben, die für eine gesonderte Auswertung verwendet werden kann. Damit wird eine zweite didaktische Anforderung erfüllt: Die Meldungen zur Morphosyntax können die geforderte Sprachbewusstheit fördern, wenn der Lerner die Meldungen in geeigneter Weise nutzt. Zusätzlich kann sich der Lerner in einer Auswertung des Dialogprotokolls einen präzisen Überblick über seinen Sprachstand, zumindest in Bezug auf die verwendeten Konstruktionen, verschaffen. Das folgende Beispiel zeigt exemplarisch einen Auszug aus einem Dialogprotokoll.

(4.1)
```
Neuendorf Polizeirevier!
Guten Tag!
Ja bitte?
---------------------
⋆ Ich habe ein Unfall gesehen.
---------------------
Im (direkten) Objekt gibt es einen Fehler:
Das Genus stimmt nicht: m vs. n[7]


Aha ...
bitte bleiben Sie vor Ort, damit wir Sie befragen
können.
Was ist denn genau passiert?
---------------------
```

[7] Die Entscheidung, ob es sich bei diesem Fehler um einen Genus- oder um einen Kasus-Fehler handelt, ist nicht trivial. In Abschnitt 5.5 wird diese Problematik diskutiert.

```
⋆ Auto ist gegen das Ampel gefahren.
---------------------
In der Präpositionalphrase gibt es einen Fehler:
Im Komplement gibt es einen Fehler:
Das Genus stimmt nicht: f vs. n

Als 1. Wort muss vermutlich ein Artikel eingefügt
werden.


Ein Auto ist gegen eine Ampel gefahren?
Wo ist der Unfall?
```

Das Protokoll in Beispiel (4.1) ist so aufgebaut, dass nach der Frage des simulierten Polizisten jeweils die Eingabe des Lerners und im Anschluss die Fehlermeldung dazu erscheint, falls ein Fehler identifiziert werden konnte.

Bei semantischen Fehlern beziehungsweise Fehlern, die durch die Limitierung des Dialoginhalts ausgelöst werden, reagiert das System nicht erster Linie mit einer „externen“ Fehlermeldung, sondern der virtuelle Gesprächspartner vermittelt das Scheitern der Interpretation der Eingabe.[8]

4.3 Mögliche Erweiterungen

Meines Erachtens ergeben sich in zwei Bereichen wünschenswerte Erweiterungen. Zum einen würde die Integration von Informationen zur lexikalischen Semantik eine wesentliche Bereicherung für den Lerner darstellen. Zum anderen ließe sich in vielen Bereichen der Systemmeldungen eine Staffelung der Informationen gestalten, die einerseits den Wissensstand des Lerners berücksichtigt (siehe auch Abschnitt 5.5) und andererseits die Aufgabengestaltung zum Beispiel durch eine Staffelung der konkreten Hilfe unterstützt. Allgemeiner gesehen erscheint es möglich, auf der Basis der relativ präzisen Informationen zur Leistung eines Lerners eine Adaptation des Systems an den Lerner zu erreichen.

4.3.1 Lexikalische Semantik

Die Informationen eines (gedruckten) Wörterbuches enthalten üblicherweise nicht nur Informationen zur Phonologie, Morphologie und zur Syntax eines Lemmas, sondern ganz wesentlich auch Informationen zur Bedeutung beziehungsweise zum Gebrauch. Gerade diese Informationen sind aber in dem hier produzierten LFG-Lexikon nicht enthalten. Eine Integration dieser Art von Daten würde die für den Bereich der Morphosyntax vermutlich ausreichenden Informationen in idealer Weise ergänzen. Allerdings erscheint die

[8] Es sei darauf hingewiesen, dass eine Analyse des kommunikativen Verhaltens des Lerners dagegen vom System nicht geleistet wird. Hier würde vermutlich auch Usermodeling eine große Rolle spielen, das aber wie erwähnt nicht Bestandteil des Projekts war.

Erweiterung des LFG-Lexikons, das in erster Linie zur automatischen morphosyntaktischen Analyse von Lernereingaben gestaltet worden ist, nicht sinnvoll. Daher sollte nicht die Erweiterung eines in erster Linie computerlinguistischen Lexikons mit Bedeutungsinformationen im Vordergrund stehen, sondern eine kohärente Darstellung der Informationen, wenn die Daten zusammengestellt aus unterschiedlichen Ressourcen dem Lerner präsentiert werden, um die volle Funktionalität eines „eletronischen Wörterbuches“ zu erreichen.

Eine Möglichkeit der Integration von Lexika mit unterschiedlichen Daten stellt die Strukturierung nach Wisisenspaketen dar, wie sie im Projekt DYLEX (Ludewig und Hötker, 1996) konzipiert worden ist. Dieses Konzept erlaubt es, konsistente Informationen in einem Wissenspaket zusammen zu fassen und inkonsistente Informationen in unterschiedlichen, hierarchisch untergeordneten Wissenspaketen zu belassen, um auf diese Weise unterschiedliche Perspektiven auf die in der Datenbank enthaltenen Wissenspakete zu erhalten. Auch wenn hier streng genommen nicht „inkonsistente“ Daten vorliegen, wären die morphosyntaktischen Daten des LFG-Lexikons dann im Wesentlichen in einem Wissenspaket enthalten, auf das auch direkt zugegriffen werden könnte, während die Daten zur lexikalischen Semantik in einem anderen gespeichert wären. Beide Wissenspakete zugleich könnten dann mit Hilfe eines hierarchisch übergeordneten Pakets erreicht werden, um auf diese Weise an alle Informationen zu einem Lexem zu gelangen. Obwohl also eine Erweiterung des LFG-Lexikons nicht sinnvoll erscheint, eröffnen sich Möglichkeiten, zusätzliche lexikalische Informationsquellen anzubinden und dem Lerner zur Verfügung zu stellen.

4.3.2 Adaptation des Systems

Der zweite angesprochene Bereich, in dem eine Erweiterung sinnvoll erscheint, ist die Adaptation des Systems an den Kenntnisstand des Lerners, die sich im Wesentlichen in zwei Aspekte unterteilen lässt.

Zusätzlich zur „natürlichen“ Ambiguität der Analysen des Parsers ergibt sich aus den angewandten Fehlerhypothesen eine weitere Vervielfachung der möglichen Interpretationen einer Lernereingabe. Beispielsweise ergeben sich aus der Analyse des Satzes „Ich sehe den Mann mit dem Auto.“ statt der üblichen 2 Analysen aufgrund der Fehlerhypothesen 10 Analysen mit einerseits unterschiedlicher Struktur und andererseits mit verschiedenen Fehlerannahmen. Beispielsweise kann mit entsprechenden Kasus- und Personfehlern auch „den Mann“ als Subjekt interpretiert werden. Wenn das System aber in der Lage ist, bestimmte Fehlertypen für einen bestimmten Lerner als wahrscheinlicher anzunehmen, dann kann auf diese Weise ein Kriterium geschaffen werden, um bestimmte Interpretationen zu bevorzugen, beziehungsweise zu disambiguieren (cf. Michaud und McCoy, 2003). Neben dieser Nutzung von Lernerdaten für interne Verfahren und Methoden sollte das

System auch in der Lage sein, die Meldungen so zu gestalten, dass eine für den Lerner verständliche und an die Übungssituation angepasste Meldung generiert wird.

Eine weitere Möglichkeit der Adaptation an den Lerner besteht in der Staffelung der Meldungen an den Lerner. Um den Lerner in der Dialogübung zu zwingen, nicht sofort die „Lösung“ anzuklicken, um damit den Dialog weiterzuführen, kann eine mehrstufige Präsentation, die nicht nach dem ersten Klick den gesamten Satz zeigt, das sofortige Nachschlagen „bremsen“. Mit Hilfe eines solchen Verfahrens lassen sich zusätzlich detailliertere Informationen über den Kenntnisstand des Lerners erzielen, um gegebenenfalls das System weiter zu adaptieren. Ein ähnlicher Informationsgewinn zur Modifikation eines Nutzer-Modells lässt auf der Grundlage des Umfangs der vom Lerner aufgerufenen Hilfefunktionen erzielen (cf. Heift, 2001). Ausführlicher wird auf diesen letzten Aspekt in Abschnitt 5.5 eingegangen, in dem gezeigt wird, welche Möglichkeiten der Adaptation sich zu einer konkreten Fehlermeldung anbieten. An dieser Stelle soll nur erwähnt werden, dass neben der Ausführlichkeit der Meldung beispielsweise auch die Terminologie modifiziert werden kann und so ein besseres Verständnis für den Lerner erzielt wird.

Offensichtlich sind zusätzliche Erweiterungen des Systems, die die Funktionalität erhöhen, denkbar. Die hier angesprochenen Bereiche stellen aber Aspekte dar, die sich unmittelbar aus den Ressourcen beziehungsweise Funktionen des Systems ergeben. Und schließlich sind diese Bereiche von Nutzern des Systems als realistische, aber fehlende Optionen eingeschätzt worden.

4.4 Systemarchitektur

In diesem Abschnitt wird kurz auf die Systemarchitektur und die interne Arbeitsweise des Programms eingegangen, wobei im ersten Teil die Systemarchitektur mit Programm- und Datenmodulen betrachtet und im zweiten die Interaktion zwischen den Modulen erläutert wird. In den Kapiteln 5 und 6 erfolgt dann eine umfassendere Betrachtung der Funktionen einzelner Module, die sich mit dem hier dargestellten Überblick leichter einordnen lassen sollten.

4.4.1 Systemarchitektur

Abbildung 4.14 zeigt zunächst die Systemarchitektur mit den verschiedenen Programmmodulen und Ressourcen.

Das Interface-Modul steuert die Kommunikation mit dem Lerner auf der Oberfläche. In diesem Modul wird zum Beispiel die Verwaltung der Fenster geregelt. Abhängig von diesem Modul sind zwei Programmteile, die mit bestimmten Aufgabentypen verbunden sind. Im ersten Modul wird die Präsentation von Dialogaufgaben sowie die Verarbeitung von Eingaben gesteuert

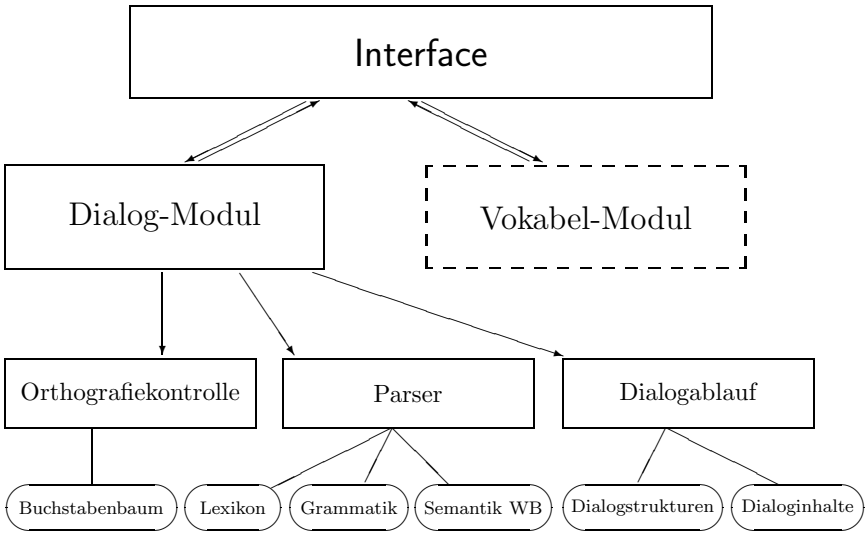


Abbildung 4.14: Architektur des Systems im Überblick

und im zweiten die Präsentation von Vokabelaufgaben. Auf Letzteres wird hier nicht weiter eingegangen, sondern nur angedeutet, wie sich das Programm um andere wichtige Übungsformen ergänzen ließe. Abhängig vom Dialog-Modul sind in erster Linie weitere Untermodule, die die linguistische Analyse übernehmen. Die Untermodule „Orthografiekontrolle“, „Parser“ und „Dialogablauf“ übernehmen in dieser Reihenfolge die Verarbeitung der Eingabe.

In der untersten Zeile der in Abbildung 4.14 dargestellten Grafik sind die wesentlichen Ressourcen aufgelistet, die zur Analyse der Eingaben und zur Fortführung des Dialogs herangezogen werden. Hierbei handelt es sich um unterschiedlich strukturierte Wissensbasen, auf die von den Analysemodulen zugegriffen wird. Für die Orthografiekontrolle wird dabei auf einen so genannten „Trie“ zurückgegriffen, der eine effiziente Speicherung der Wörter in Form eines Buchstabenbaums realisiert. Mit Hilfe des Baumes kann außerdem leicht eine mögliche Korrektur zu einer nicht enthaltenen Buchstabenkette generiert werden. Dabei wird eine Art Levenshtein-Distanz verwendet, um ein möglichst ähnliches Lexem aus dem Buchstabenbaum zu extrahieren, das dann dem Lerner als Korrekturmöglichkeit nach einer fehlgeschlagenen Analyse präsentiert wird. Es besteht außerdem eine beschränkte Möglichkeit, neue Wörter aufzunehmen, das heißt, großgeschriebene Wörter werden temporär als Namen in das Lexikon integriert, wenn der Nutzer dies in einer Abfrage bestätigt.

Der Parser nutzt die bekannten Ressourcen wie Lexikon und Grammatik und wendet im Anschluss an die morphosyntaktische Analyse eine hierarchisch aufgebaute Wissensbasis (WB) mit semantischen Informationen auf

die Analyseergebnisse an, um eventuelle Verletzungen von Selektionsrestriktionen zu entdecken. Dazu sind die semantischen Informationen in einer Ontologie strukturiert worden, in der die Knoten der Verben mit den möglichen thematischen Rollen und ihren semantischen Argumenten annotiert sind.

Der Dialogablauf wird wesentlich durch zwei Ressourcen bestimmt: 1. eine Ressource „Dialogstrukturen“, die in einem Übergangsnetzwerk Informationen über den Ablauf eines Frage-Antwort-Dialogs enthält und 2. „Dialoginhalte“, in der die Inhalte eine Dialogs wie beispielsweise die Unfallmeldung wiederum hierarchisch strukturiert gespeichert sind.

Weitere kleine Module, die hier nicht im Einzelnen aufgeführt sind, regeln zum Beispiel die Abspeicherung der Ergebnisse, die Präsentation von zusätzlichen Hilfefenstern oder die Grafik und Animation in den Aufgabenstellungen.

4.4.2 Überblick über die Eingabeanalyse

Abbildung 4.15 zeigt anhand einer Art Flussdiagramm den Ablauf einer Eingabeanalyse.

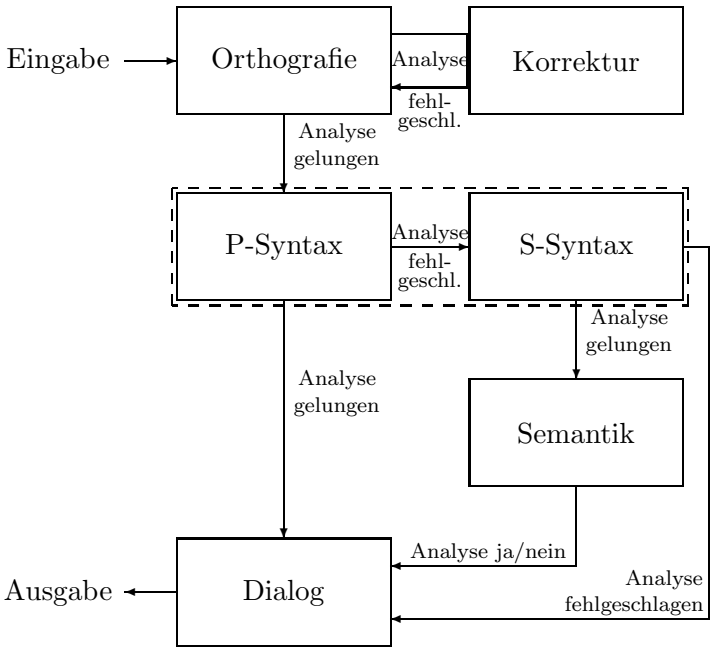


Abbildung 4.15: Verarbeitungsschritte einer Lernereingabe

Nachdem in einem gesonderten Fenster dem Nutzer die Aufgabenstellung verdeutlicht worden ist, wird der Dialog mit einer ersten Reaktion beziehungsweise einer Frage des simulierten Dialogpartner begonnen. Diese

Reaktion stammt bereits aus der Datenbasis „Dialoginhalte“, das heißt aus der Datenbasis für einen spezifischen Dialog. Wie in Abschnitt 4.1 gezeigt, wird vom Lerner im Allgemeinen die Beantwortung einer Frage erwartet.

Zu Beginn wird vom System untersucht, ob die verwendeten Wörter im Lexikon enthalten sind. Die Kontrolle der Orthografie (Abbildung 4.15) ist so aufgebaut, dass für alle unbekannten Wörter der Versuch unternommen wird, einige Korrekturkandidaten zu generieren und die Eingabe wird nicht weiterverarbeitet, bevor dem System nicht alle Wörter bekannt sind. Nähere Erläuterungen sind Abschnitt 6.2.1 zu entnehmen.

Im Anschluss an die Orthografiekontrolle wird die Eingabe an den Parser weitergereicht, mit dessen Hilfe zunächst versucht wird, die Eingabe als unvollständigen Satz, das heißt als Phrase zu parsen (P-Syntax). Zugrunde liegt die Einsicht, dass die Antwort auf eine Frage in Abhängigkeit von eben dieser Frage elliptisch sein kann. Wenn der Parsevorgang ein Ergebnis erzielt, wird es an die Dialog-Komponente weitergereicht. Falls die Analyse als Phrase fehlschlägt, wird in einem zweiten Schritt versucht, die Eingabe als vollständigen Satz zu parsen (S-Syntax). Wenn auch die morphosyntaktische Analyse als Satz scheitert, wird die Eingabe in „roher“ Form dazu genutzt, eine Fortführung des Dialogs zu erreichen. Dazu sind in den Daten der Dialoginhalte Schlüsselwörter enthalten, die als „Fall-back“ genutzt werden können. Eine genaue Beschreibung der Funktionalitäten des Parsers erfolgt in Kapitel 5.

Falls das Parsing einer Eingabe gelungen ist, wird als Nächstes die Semantik überprüft und dann das Ergebnis an die Dialog-Komponente übergeben. Falls sich aus der Kontrolle der Semantik ein Fehler ergeben sollte, wird dieses gespeichert und bei der Reaktion des Dialogmoduls berücksichtigt, indem wie im vorherigen Abschnitt erläutert mit Hilfe des Dialogmoduls eine „Nachfrage“ präsentiert wird. Die genauen Ausführungen zur Kontrolle der Semantik befinden sich in Abschnitt 6.2.2.

Das letzte wesentliche Modul ist die Steuerung des Dialogs, die schließlich die Ausgabe einer Reaktion an den Lerner bewirkt. Je nachdem, wie erfolgreich die vorherige Verarbeitung war, werden dem Dialogmodul die Ergebnisse in unterschiedlicher Form übergeben: entweder a) als semantische Struktur einer Phrase, die nicht auf ihre semantische Konsistenz überprüft werden konnte, b) als sematische Struktur eines Satzes, die möglicherweise inkonsistent ist, oder c) als unverarbeitete Wortfolge. Auf dieser Basis erfolgt dann eine angepasste Reaktion des Systems, wie es ausführlicher in Abschnitt 6.1 dargestellt wird.

Nachdem nun ein Überblick über die Funktionsweise des Systems gegeben worden ist, folgen in Kapitel 5 die theoretischen und praktischen Erläuterungen zur automatischen, morphosyntaktischen Analyse von Lernereingaben. Einerseits sollen Parse-Ergebnisse erzielt werden, auch wenn die Eingabe nicht durch die Grammatik und das Lexikon beschrieben wird und andererseits sollen im Fehlerfall aus diesen Analysen für den Lerner

verständliche Fehlermeldungen generiert werden.

Kapitel 5

Computerlinguistische Diagnose von fehlerhaften Eingaben

Dieses Kapitel ist der Hauptteil der Arbeit, in dem dargestellt wird, welche Methoden zur Fehlererkennung und Korrektur in der Vergangenheit entwickelt und welche in das hier entwickelte Promis*D*-System integriert worden sind.

Zwei generelle Ansätze zum Parsen von fehlerhaften Eingaben, das heißt Eingaben, die nicht von der internen Grammatik für grammatische Sätze beschrieben werden, lassen sich in der Computerlinguistik unterscheiden. Erstens existieren so genannte *robuste* Parsingverfahren, in denen versucht wird, das Parsing an der Fehlerstelle fortzusetzen, ohne die Art und Weise des Fehlers in Betracht zu ziehen. Mit relativ „schwachen“ Verfahren wird dabei versucht, das Parsing über die Fehlerstelle hinweg zu ermöglichen. Im Vordergrund steht dabei das Ziel, möglichst große Teile der Eingabe, so genannte „Chunks“, zu analysieren (z.B. Jensen et al., 1983, "'Parse-Fitting"'). Ähnliches gilt insbesondere auch in der Analyse von gesprochener Sprache, bei der zusätzlich zu Fehlern wie einem Abbruch der Phrase, Wiederholung von Satzteilen, Korrekturen etc. auch mit Erkennungsfehlern des Spracherkenners zu rechnen ist (z.B. Alexandersson et al., 1994, und andere Verbmobil-Reports). Zweitens wurden *sensitive* Parsingverfahren entwickelt, die von vornherein darauf ausgelegt sind, Fehler zu lokalisieren und zu identifizieren. Sie können so mit Hilfe einer Korrektur oder einer Integration des Fehlers in die Analysestruktur den Parsingprozess fortsetzen und eine Beschreibung des ganzen Satzes erreichen, die außerdem Aussagen über die Fehlerposition und den Fehlertyp machen kann. Im Rahmen eines Programms, das die speziellen Aufgaben hat, einerseits Beschreibungen für fehlerhafte Eingaben zur Fortsetzung von Aufgaben beziehungsweise Dialogen zu generieren sowie andererseits die Grammatikalität einer Eingabe

zu bestimmen und entsprechendes Feedback zu bieten, kann nur die zweite Möglichkeit des sensitiven Parsings in Betracht kommen.

Das Ziel der Analyse eines fehlerhaften Satzes in diesem Kontext ist es, eine Beschreibung und möglicherweise eine Korrektur zu liefern, obwohl das eigentlich aufgrund der linguistischen Datenbasis, das heißt aufgrund der Grammatik und des Lexikons für korrekte Sätze, nicht möglich ist. Zwei Strategien bieten sich an: Entweder wird der Analysealgorithmus derart verändert, dass er trotzdem eine Beschreibung liefern kann oder es wird die Grammatik so verändert, dass mit Hilfe von zusätzlichen, so genannten Fehlerregeln eine Beschreibung generiert werden kann. Verschiedene Aspekte dieser Strategien sollen im folgenden Abschnitt erläutert und diskutiert werden. Anschließend soll in zwei getrennten Abschnitten auf die beiden Bereiche eingegangen werden, die in der LFG den Aufbau der Beschreibungen leisten: Phrasen-Struktur-Parsing (Abschnitt 5.2) und Feature-Unifikation (Abschnitt 5.3). Für die Fehlererkennung im Bereich der Phrasen-Struktur wird eine Erweiterung des bekannten, Chart-basierten Earley-Algorithmus vorgeschlagen, in dem auf der Basis einer bestimmten Chart-Konfiguration mit Hilfe eines Triggers unterschiedliche Fehlerhypothesen als gewichtete Kanten zur Chart hinzugefügt werden. Sich widersprechende Werte in einer F-Struktur, die auf einen Fehler beispielsweise im Bereich der Kongruenz hinweisen, werden mit Hilfe eines speziellen Fehler-Features in die F-Struktur integriert, ohne dass dabei die Konsistenz einer Struktur zerstört würde. Im Anschluss an die Darstellung folgen in Abschnitt 5.4 einige Anmerkungen zur Effizienz und zur Evaluation des Verfahrens. Dazu wurden wie erwähnt 2 Korpora untersucht, die nicht nur Daten für den Abdeckungsbereich der Grammatik, sondern auch für Möglichkeiten zur Effizienzsteigerung und zur Leistungsfähigkeit des Verfahrens zur Verfügung stellen. Schließlich wird in Abschnitt 5.5 gezeigt, wie aus den Analysen der fehlerhaften Eingaben die Informationen für das Feedback an den Lerner gewonnen werden können.

5.1 Allgemeine computerlinguistische Aspekte der Analyse fehlerhafter Sätze

Da die Voraussetzung für eine Rückmeldung über den Typ des Fehlers und möglicherweise über eine Korrektur des Fehlers nur von sensitiven Systemen geleistet werden kann, konzentrieren sich die Ausführungen in den folgenden Abschnitten auf dieses Konzept.

Zur Analyse von morphosyntaktischen Fehlern mit Hilfe von Parsern lassen sich im Wesentlichen zwei Verfahren anwenden: 1. die Antizipation von Fehlern in der Grammatik und/oder dem Lexikon und 2. die „verfahrensbasierte“ Erkennung von Fehlern.

1. Unter Antizipation von Fehlern möchte ich hier Verfahren zusammenfassen, die entweder Informationen zu Fehlern in der Grammatik (so genann-

te Fehlerregeln) oder in Meta-Regeln (als bekanntes Beispiel Weischedel und Sondheimer, 1983) kodieren. Die Unterschiede zwischen diesen beiden Ansätzen scheinen mir in Bezug auf die Stärken und Schwächen eines Systems unerheblich. Im Regelfall wird die Grammatik um Elemente erweitert, die dazu geeignet sind, fehlerhafte Eingaben zu analysieren. Im ersten Ansatz wird die Grammatik um „echte“, beschreibende Regeln erweitert, die zusätzlich eine Markierung zur Art des Fehlers in unterschiedlicher Ausführlichkeit enthalten. Im zweiten wird die Grammatik im allgemeinen so erweitert, dass im Fall eines Fehlers Programmfunktionen außerhalb des Parsers aufgerufen werden, die den Fehler analysieren und markieren. Dort wird dann von anderen Modulen im Programm versucht, den Fehler zu erkennen und zu verarbeiten.

2. Verfahrensbasierte, antizipationsfreie Ansätze dagegen belassen die linguistischen Daten im System nahezu unverändert und versuchen, Fehler mit Hilfe des Analyseverfahrens, das heißt des Parsings zu diagnostizieren (Menzel, 1992; Fouvry, 2003). Dabei muss im Wesentlichen zwischen Ansätzen zur Fehleranalyse mit Hilfe von Phrasenstruktur-Grammatiken und von unifikationsbasierten Grammatiken unterschieden werden, da in fast allen Systemen zur Fehlererkennung Grammatiken mit Hilfe des einen oder anderen Mechanismus formuliert sind.

Schließlich existieren gemischte Ansätze, die je nach Strukturtyp unterschiedliche Verfahren anwenden (z.B. Schwind, 1988; Schneider und McCoy, 1998). Zumeist werden hier Fehler in Merkmalsstrukturen mit Hilfe eines antizipationsfreien Verfahrens analysiert, während das Problem im Bereich der PS-Regeln mit Hilfe von zusätzlichen Regeln beziehungsweise Erweiterungen von Regeln gelöst wird.[1]

Zunächst erfolgt nun eine allgemeinere Darstellung zu diesem Bereich, um anschließend auf die Fehleranalyse auf der Basis der LFG-Grundstrukturen K- und F-Struktur eingehen zu können.

5.1.1 Antizipation von Fehlern

Wenn kommerzielle Sprachlernprogramme in der Lage sind, auf fehlerhafte Eingaben mit mehr als nur einer FALSCH-Meldung zu reagieren, basieren die Methoden meines Wissens immer auf der Antizipation von Fehlern. Für bestimmte Aufgabentypen lassen sich damit zum Teil erstaunliche Ergebnisse erzielen, da insbesondere in Lückentexten und ähnlichen Aufgabenstellungen die sinnvollen Möglichkeiten, eine Lücke zu füllen, durch den Kontext beziehungsweise die Aufgabenstellung selber meist sehr eingeschränkt sind. Die verbleibenden Möglichkeiten sind dann leicht von einem Entwickler vorherzusehen. Eine unwesentlich erweiterte Fehleranalyse sieht das kommerzielle Programm *Interaktive Sprachreise 5* (2001) von Digital

[1] Andere Klassifikationen mit anderen Parametern zum Beispiel in Menzel (1992, S. 22f) oder Heift (1998, S. 8ff).

Publishing vor. Dem Übungsmodul ist ein Lexikon hinzugefügt worden, das auf dieser Basis die Unterscheidung von orthografischen und „inhaltlichen“ Fehlern in Lernereingaben erlaubt, was aber keineswegs zur Erweiterung der Aufgabenstellung genutzt wird. Damit sind nach meinen Erkenntnissen die Möglichkeiten zur Generierung von Rückmeldungen in kommerziellen Programmen im Wesentlichen erschöpft.

Ein relativ einfaches Verfahren zur Erkennung von Fehlern, das die Antizipation von Mustern erweitert und automatisiert, wird in Hu et al. (1998) kurz erläutert. Die Autoren haben ein Lerner-Korpus zusammengestellt und auf Fehler hin annotiert, sodass die Ergebnisse generalisiert und in einer Datenbank zusammengefasst werden konnten. Ein Pattern-Matching-Algorithmus erkennt nun einen Fehler, wenn es einen ähnlichen Fall in der Datenbank gibt, und kann daraufhin sinnvolle Rückmeldungen dazu geben. Damit wurde ein System entwickelt, dass in gewisser Weise analog zu einen „memory-based“ Übersetzungssystem funktioniert, da in beiden Fällen ein Ähnlichkeitmaß zur Bewertung der Eingaben im Vergleich mit Elementen in der Datenbank von entscheidender Bedeutung ist. Die Autoren machen aber keine Angaben über die tatsächliche Leistungsfähigkeit, weshalb eine Bewertung nicht möglich ist.

Im Folgenden werden Ansätze angesprochen, die im Gegensatz zu den vorhergehenden Beispielen Methoden der Computerlinguistik im engeren Sinne zur Erkennung von Fehlern verwenden, um einen Vergleich mit dem hier entwickelten Konzept zu ermöglichen. Das folgende Beispiel (5.1) von PS-Regeln in Prolog-Notation stammt aus Schwind (1988), wo ein Konzept für französische Deutschlerner vorgestellt wird.

(5.1)
```
np(X,X0) :- art(X,X1),ng(X1,X0,F),F.
ng(X,X0,correct) :- ag(X,X1),noun(X1,X0);noun(X,X0).
ng(X,X0,error(noun,ag)) :- noun(X,X1),ag(X1,X0).
```

Diese Regeln beschreiben die Verwendung eines Adjektivs in einer Nominalphrase: Im Deutschen folgt das Substantiv dem Adjektiv, während im Französischen das Adjektiv normalerweise dem Substantiv folgt, was hier als Fehler markiert ist. Das `error`-Prädikat löst dann die Fehlermeldung aus. Hier zeigt sich schon eine wesentliche Schwierigkeit des Verfahrens. Die Fehlerregel macht nur dann Sinn, wenn davon ausgegangen werden kann, dass es sich bei den Muttersprachen der Benutzer um Sprachen handelt, in denen die Nachstellung des Adjektivs eine gültige Wortstellung ist.

Die allgemeinere Frage dabei ist, ob ein Linguist, der die Grammatik und die darin enthaltenen Fehlerregeln verfasst, die entscheidenden Fehlertypen berücksichtigt. Dieses ist insbesondere bei der Verarbeitung von freien Eingaben von Bedeutung. Hier muss mit den verschiedensten Fehlertypen und -positionen gerechnet werden. Ohne Hinweise auf mögliche Fehler zum Beispiel über die Muttersprache der Lerner wäre sinnvoller Weise keine Einschränkung in der Modellierung der Fehlerregeln in der Grammatik erlaubt.

In der Konsequenz hieße das, beliebige Wortstellungen müssten berücksichtigt werden.

Ein ähnliches Konzept der Antizipation wird auch in Schneider und McCoy (1998) vertreten, in dem sich folgendes Beispiel findet.

(5.2)
```
DP(agr ?a) -> Det(agr ?a) NP(agr ?a)
DP(agr s)(error +) -> Det(agr (?!a s)) NP(agr s)
DP(agr p)(error +) -> Det(agr (?!a p)) NP(agr p)
```

Die erste Regel wird gebraucht, um eine korrekte DP zu analysieren, während die zweite und dritte Regel dazu verwendet wird, eine Numerus-Inkongruenz zu beschreiben. Der Negations-Operator „!“ erlaubt es, festzulegen, dass das Numerus-Feature des Det nicht mit dem der NP übereinstimmen darf. In beiden dargestellten Beispielen würde ein Parser zuerst versuchen, die fehlerhafte Eingabe mit Hilfe der „korrekten“ Regel zu analysieren, um dann im Fehlerfall eine Beschreibungsstruktur mit Hilfe der Fehlerregel zu liefern.

Im Beispiel (5.2), das für das Englische gilt, zeigt sich als zweite wesentliche Schwierigkeit, dass eine Grammatik, die ein größeres Fragment einer Sprache beschreiben will, sehr viele Regeln enthalten muss, und deren Anzahl würde sich noch einmal vervielfachen, wenn außerdem noch andere Features wie Kasus, Genus, Definitheit usw. in separaten Regeln berücksichtigt werden müssten. Üblicherweise enthalten umfassende Systeme zur Beschreibung korrekter Sätze mehrere tausend Regeln, die dann jeweils an die zu erwartenden Fehler angepasst werden müssten. Aus diesen Überlegungen ergibt sich ein wesentlicher Nachteil dieses Typs der Fehlererkennung, bei dem die Grammatiken gewaltig aufgebläht würden, um in möglichst vielen Fällen ein präzises Feedback geben zu können, was wiederum zu großen Effizienzproblemen führt.

Neben den aufgeführten Nachteilen besteht ein Vorteil der Antizipation von Fehlern in der Möglichkeit zur Verwendung von bekanntermaßen effizient arbeitenden Verfahren zur Analyse von natürlicher Sprache. Einzige Bedingung ist, dass sich der Ort der Markierung von Fehlermöglichkeiten eindeutig bestimmen lässt. Krüger-Thielmann (1992) zum Beispiel beschreibt eine präzise umrissene Domäne, nämlich die der französischen Zahlwörter. Hier lässt sich genau bestimmen, in welcher Art und Weise die Struktur der französischen Numerale von denen im Deutschen abweicht. Darauf aufbauend lassen sich dann bestimmte Fehlertypen und -positionen vorhersagen und in die Beschreibung integrieren.

Ein ähnlicher Fall liegt auch in Heift (1998) vor, deren System zwar nicht explizit ein antizipationsbasiertes ist, in deren Ansatz meines Erachtens aber nur relativ einfache grammatische Konstruktionen analysiert werden können. Der auf der HPSG basierende Ansatz regelt die Wortstellung mit Hilfe von Listen als Werte von Attributen. Die einzelnen Positionen in diesen Listen sind für bestimmte Wortarten markiert. Wenn also zum Beispiel ein

finites Verb nicht an der zweiten Position im Satz auftaucht, wird diesem Element ein Fehler-Feature hinzugefügt. Damit können dann Aussagen über die Fehlstellung einzelner Wortarten gemacht werden, allerdings nicht über die tatsächliche vorgefundene Wortstellung. Dass dieses trotzdem didaktisch sinnvoll sein kann, zeigt sich in der in Heift (2001) vorgestellten Aufgabenstellung, in der es um die Konstruktion einfacher Sätze auf der Basis von vorgegebenen Grundformen geht.

Schließlich kann auch Foster und Vogel (2004) angeführt werden, bei denen ein Ansatz präsentiert wird, der eine Fehlergrammatik explizit auf einem Fehlerkorpus aufbaut. Allerdings ist dieser Ansatz wenig umfassend: Es werden nur Sätze analysiert, die einen einzigen Fehler beinhalten und es werden nur die drei Operationen „Einfügung“, „Auslassung“ und „Ersetzung“ eines Wortes ausgeführt. Die Operation „Ersetzung“ erweitert das Verfahren allerdings, da damit auch Kongruenzfehler korrigiert werden. Das Verfahren arbeitet ähnlich wie Mellishs bekannter Ansatz (Mellish, 1989) in zwei Phasen, indem zunächst eine Chart bottom-up mit den Regeln für eine korrekte Grammatik aufgebaut wird und anschließend nacheinander verschiedene Fehlerregeln angewandt werden, um eine vollständige Analyse zu erreichen. Falls eine Fehlerregel nicht zum Erfolg führt, werden alle Kanten in der Chart gelöscht, die auf dieser Kante aufgebaut haben, und somit eine Erkennung von multiplen Fehlern unmöglich gemacht. In einer Evaluation wurden nicht Sätze von Fremdsprachenlernern, sondern Sätze von Muttersprachlern verwendet, woraus sich unter anderem auch die hohe Erfolgsquote der Fehlererkennung (84%) ergibt, da nur 5 Sätze Mehrfachfehler enthielten. Für die Verwendung des Ansatzes in einem CALL-Kontext reicht es aber nicht aus, nur einen Fehler pro Satz zu erkennen, obwohl sich Fosters und Vogels Ansatz sicher erweitern ließe, wenn zum Beispiel wie in Promis*D* die Chart mit Fehlerhypothesen nicht nach jeder fehlgeschlagenen Hypothesenanwendung mit den entsprechenden Konsequenzen wieder gelöscht würde.

Ein weiterer Vorteil eines antizipationsbasierten Ansatzes gegenüber der Nicht-Antizipation ist die Möglichkeit zur Unterscheidung von fehlerhaften und nicht von der Grammatik abgedeckten Eingaben. Wenn solch ein System also einen Fehler identifiziert hat, der auf einer Fehlerregel basiert, kann der Fehler mit der entsprechenden Sicherheit an den Lerner weitergegeben werden. Aus didaktischer Perspektive handelt es sich hierbei um einen ganz entscheidenden Aspekt, da ein Lerner mit größtmöglicher Sicherheit über seine Leistungen informiert werden sollte. Wenn das System nicht in der Lage ist, eine Beschreibung zu generieren, kann allerdings offensichtlich nicht gesagt werden, dass es sich um eine fehlerhafte Äußerung handelt.

5.1.2 Antizipationsfreie Verfahren

Ein erster Vorteil aus dem Einsatz von antizipationsfreien Verfahren ergibt sich in der Hinsicht, dass bestimmte Fehlertypen, wenn sie generell vom System erkannt werden, an potentiell beliebigen Positionen identifiziert werden können. Bezogen auf ein Beispiel wie (5.2) bedeutet das, dass beispielsweise eine Inkongruenz des Verbs mit dem Subjekt im Deutschen nicht nur erkannt wird, wenn das Subjekt vor dem finiten Verb steht, sondern auch, wenn der Satz ein topikalisiertes Element enthält und das Subjekt erst nach dem finiten Verb folgt und damit eine andere PS-Regel angewendet wird.

Mit Hilfe sehr genereller Verfahren zum Parsen mit einfachen PS-Regeln können auch Konstituentenfehler an beliebigen Positionen erkannt und gegebenenfalls korrigiert werden.[2] Als Beispiel sei Kato (1994) (aufbauend auf Mellish, 1989) genannt, mit dessen Verfahren Einfügungen, Auslassungen und Ersetzungen erkannt werden können.[3] Ein weiteres Beispiel ist Lee et al. (1995), deren Verfahren auf einem gänzlich anderen Konzept als Katos beruht, in dem aber von einer Evaluation berichtet wird, in der eine relativ kleine Grammatik (192 PS-Regeln) zur Analyse eines Korpus von 400 Sätzen diente. Diese Evaluation ergab immerhin ca. 68% Sätze mit „non crossing brackets“ verglichen mit dem von Hand analysierten Korpus. Dazu muss angemerkt werden, dass das Verfahren zur Evaluation nicht eindeutig interpretiert werden kann, da auf der Basis von „non crossing brackets“ keine Aussagen darüber gemacht werden können, ob es sich bei Klammerfehlern um „schwere“ Fehler oder nur um „leichtere“ Fehler zum Beispiel des falschen Attachments einer Präpositionalphrase handelt, obwohl der Schwerpunkt bei diesem Konzept auf robustem Parsing und nicht auf der Erkennung von Fehlern lag. Das Verfahren in diesem Ansatz ist dem hier entwickelten ähnlich und wird daher im nächsten Abschnitt 5.2 genauer erläutert und bewertet. In beiden Ansätzen sind also relativ uniforme Methoden entwickelt worden, was als zusätzlicher Vorteil gesehen werden kann.

Ein weiterer wichtiger Vorteil eines antizipationsfreien Verfahrens besteht in der Möglichkeit zur unabhängigen Entwicklung einer Grammatik und eines Lexikons. Die Grammatik und das Lexikon sind dann so konstruiert, dass sie tatsächlich nur korrekte Sätze beschreiben. Hieraus ergibt sich die Möglichkeit zur Integration von „fremden“ Daten, die zum Beispiel in anderen Zusammenhängen entwickelt wurden.

Während wie erwähnt bei antizipationsbasierten Systemen der Aufwand zur Erkennung von fehlerhaften Eingaben in der Grammatik und den entsprechenden Fehlerregeln liegt, muss in einem antizipationsfreien System die-

[2] Hier sei noch einmal angemerkt, dass ich zu Konstituentenfehlern nicht nur Wortstellungsfehler im engeren Sinne zähle, das heißt Sätze, bei denen die Reihenfolge vertauscht ist, aber alles lexikalische Material vorhanden ist. Zusätzlich möchte ich zu diesem Fehlertyp, den ich Verschiebungsfehler nenne, auch Auslassung-, Einfügungs- und Ersetzungsfehler zählen.

[3] Nähere Erläuterungen zu diesem Verfahren folgen auf Seite 117.

ser Aufwand in das Erkennungsverfahren übertragen werden. Der in Menzel (1992) beschriebene Ansatz zur so genannten modellbasierten Fehlerdiagnose zeichnet sich zum Beispiel durch einen relativ hohen Aufwand aus, der zur Diagnose eines Fehlers betrieben werden muss. Jede für die Kongruenz beziehungsweise Rektion relevante syntaktische Eigenschaft eines Wortes wird mit jeder korrespondierenden Eigenschaft eines anderen Wortes innerhalb einer Phrase verknüpft. Um eine umfassende und äußerst präzise Fehleranalye zu ermöglichen, muss für jede Wortform ein eigener Lexikoneintrag vorhanden sein, wobei keine Eigenschaft wegfallen darf, das heißt, dass jedes Homonym als eigener Eintrag verzeichnet sein muss. Für ein Wort wie *Mann* bedeutet dies beispielsweise, dass nicht mit einer Verkürzung wie ([CASE $\neq$ genitiv]) gearbeitet werden darf, sondern alle möglichen Kasus ausspezifiziert werden. Für andere Wortarten wie zum Beispiel Adjektive, bei denen es einen noch stärkeren Zusammenfall von Wortformen gibt, müssen dann sehr viele Lexikoneinträge gestaltet werden (siehe zum Beispiel auch die Diskussion in Schwind, 1994).[4]

In dem Ansatz von Mellish (1989) werden für das Verarbeiten von fehlerhaften Eingaben zwei Parsevorgänge benötigt. Der erste arbeitet bottom-up und erzeugt dabei soviele Kanten in einer Chart, wie es auf der Basis der „korrekten“ Grammatik und der Standard-Parseregeln möglich ist. Danach folgt ein zweiter Parsevorgang Top-Down, der versucht, möglichst nur mit Hilfe der Kanten in der Chart eine Strukturbeschreibung für den Satz zu generieren. Dazu sind natürlich die Parseregeln verändert worden, um zum Beispiel Grammatikregeln zu verwenden, falls sich keine geeigneten Kanten für die Fortsetzung des Parsevorgangs in der Chart finden.

Aufgrund der komplexen Verfahren, die relativ viele Daten produzieren und damit die Systeme potentiell ineffizient machen, stellt die Eingrenzung des Suchraums einen besonders schwierigen Aspekt bei der antizipationsfreien Analyse dar. Da solche Systeme den Suchraum an (fast) beliebigen Positionen für potentielle Fehler öffnen müssen, sollten geeignete Verfahren entwickelt werden, die den Suchraum von vornherein in möglichst restriktiver Weise einschränken. Gefunden werden soll ja im Allgemeinen eine möglichst global minimale Fehlerhypothese. Dieses Problem kann an dem oben erwähnten Beispiel Lee et al. (1995) verdeutlicht werden. Hier werden bei nur 192 Regeln in der Grammatik eine Menge von durchschnittlich 12 000 Kanten mit und 25 000 Kanten ohne Heuristik pro Satz in der Chart erzeugt. Es werden also zwar beliebige Sätze analysiert, aber die Kosten sind dafür sehr hoch. Auch Krüger-Thielmann (1992, S. 117) berichtet von einem antizipationsfreien Verfahren, das eine um mindestens 20% längere Analysezeit gegenüber einem Verfahren mit Markierung der Fehlerstellen in der Grammatik hat, wobei zu betonen ist, dass es sich hier wie erwähnt „nur“ um ein Grammatikfragment zur Beschreibung von Numeralen handelt.

[4]Dieser spezielle Aspekt wird noch in Abschnitt 5.5.3 (S. 170) zu diskutieren sein.

Heuristiken

Ein Verfahren, das sehr komplexe Heuristiken zur Integration von lokalen und globalen Beschränkungen vorsieht, wurde in Mellish (1989) entwickelt. Verwendet wurden unter anderem die folgenden, vereinfacht dargestellten Heuristiken:

- Es werden Kanten bevorzugt, die mit Hilfe der so genannten fundamentalen Regel erzeugt worden sind.
- Kanten, die aus der Anwendung bestimmter anderer Parsing-Regeln entstehen, werden mit einer Strafe belegt und dadurch benachteiligt.
- Kanten können einen geschätzten Strafwert bekommen, der auf der hypothetischen Erfüllung aller offenen Elemente beruht.
- Kanten können einen Strafwert bekommen, je nachdem, wie viele Wörter der Eingabe sie überspannen.
- Die Kante mit dem geringsten Strafwert wird zuerst berücksichtigt.
- Die Kante, die die geeignetste Anzahl von Wörtern für die Eingabe vorsieht, wird bevorzugt.

Hier zeigt sich als Beispiel, welche Möglichkeiten man in Bezug auf eine Chart hat, Kanten zu bewerten und so das Parsing zu beschleunigen, obwohl damit offensichtlich nur die Identifizierung der Lösung mit dem geringsten Strafwert (oder auch Fehlerwert) erleichtert wird; die Generierung von Lösungsalternativen bleibt trotzdem sehr aufwändig. Die Verbesserung der Heuristiken war auch ein wesentlicher Aspekt in dem auf Mellishs Ansatz basierenden Vorschlag von Kato (1994). Er schlägt vor, jeder Kante (der zweiten Top-Down-Phase) nur noch zwei Fehlerwerte zuzuordnen, die sich aus den Regeln zur Ableitung einer den Satz überspannenden Kante und damit einer Fehlerhypothese ergeben. Nach seinen Angaben werden damit in der zweiten Phase weniger Kanten produziert, was eine schnellere Identifizierung im Rahmen einer A$*$-Suche zur Folge hat.

In anderen Ansätzen müssen zum Teil zusätzliche „linguistische“ Anstrengungen unternommen werden, um die geeignete Hypothese als Resultat herauszufiltern. Es werden nicht numerische Werte, die von den linguistischen Einheiten abstrahieren, berücksichtigt, sondern es werden die linguistischen Elemente der Grammatik und des Lexikons herangezogen. Als Beispiel sei noch einmal Schwind (1988) erwähnt, die bei einer bestimmten fehlerhaften Konstruktion einen Kasus-Filter einführen muss, damit das intuitiv „plausibelste“ Ergebnis identifiziert wird. In dem hier vorgestellten Konzept wird der Suchraum auf ähnliche Weise eingeschränkt, sodass eine Fehleridentifikation nur an den Elementen möglich oder sinnvoll ist, die sich auch in Belegen als häufige Fehlerquelle erwiesen haben.

Ein Ansatz, der ein grundsätzlich anderes Verfahren und damit einhergehend ein anderes Grammatikmodell verwendet, wird in Menzel und Schröder (1998a) vorgestellt. Als Grundlage für die Modellierung von sprachlichen Phänomenen wird eine so genannte Constraint Dependency Grammar

(CDG) verwendet. Der Parsingprozess wird als ein Constraint-Satisfaction-Problem betrachtet, wobei zur Analyse von fehlerhaften Äußerungen Constraints grundsätzlich verletzbar sind. Zur Bestimmung einer optimalen Lösung durch den Disambiguierungsprozess können die Constraints zusätzlich gewichtet werden. Constraints, die als „harte“ Beschränkungen angesehen werden, bekommen einen Straffaktor 0, der bewirkt, dass Analysen, die diesen Constraints nicht genügen, für die weitere Disambiguierung nicht mehr berücksichtigt werden. Constraints, die zwar eigentlich für die Wohlgeformtheit eingehalten werden müssten, aber vom Lerner verletzt werden, bekommen einen Straffaktor nahe 0. Hierbei handelt es sich also um „relativ“ harte Constraints. Constraints, die lediglich Präferenzen für eine bestimmte Analyse darstellen, zum Beispiel für nicht topikalisierte Formen, bekommen einen Straffaktor von nahe 1. Durch die Multiplikation dieser Strafmaße kann am Ende des Parsings unter Umständen eine eindeutige Beschreibung der Äußerung mit dem geringsten Strafmaß bestimmt werden.

Ein wesentlicher Vorteil dieses Verfahrens besteht in dem umfassenden Ansatz zur Erkennung von Fehlern in sprachlichen Äußerungen. Da sich die Constraints zur Beschreibung von zum Beispiel Kongruenz und Konstituenz in Bezug auf die Identifikation von Fehlern nicht unterscheiden, können mit Hilfe eines einzigen, uniformen Verfahrens beliebige Fehlertypen analysiert werden. Zusätzlich lassen sich auf weiteren Ebenen auch thematische beziehungsweise semantische Constraints integrieren, die im Zusammenspiel mit den syntaktischen Constraints die Auswahl einer Beschreibung weiter unterstützen. Damit wird eine außerordentliche Robustheit erreicht, da das Verfahren in der Lage ist, nahezu jeder Eingabe eine Beschreibung zuzuordnen.

Ein Nachteil der Verwendung einer gewichteten CDG, insbesondere verglichen mit dem hier vorgestellten Verfahren, ist die Beschränkung auf maximal binäre Constraints. Diese Beschränkung wurde eingeführt, um das Problem der Komplexität der Disambiguierungsaufgabe auf eine handhabbare Aufgabe reduzieren zu können. Für die linguistische Beschreibung müssen daher zum Teil recht artifizielle Konstruktionen gewählt werden, um in diesem Rahmen zu bleiben. In Schröder et al. (2000) wird daher vorgeschlagen, für bestimmte, sehr beschränkte Bereiche ternäre Constraints zuzulassen. Ein weiterer Nachteil könnte in der starken Erweiterung des Hypothesenraums liegen, wenn gewichtete Contraints zugelassen werden, da ohne eine geeignete Heuristik für das sogenannte „Pruning“ keine Analyse unberücksichtigt bleiben darf. Zu Evaluationzwecken wurde ein einfacher Subjekt-Objekt-Satz systematisch fehlerhaft verändert, bei dem die Robustheit des Verfahren deutlich wird (Menzel und Schröder, 1998b). In den Ausführungen bleibt allerdings unklar, was mit „percentage of correct analyses“ im Verhältnis zu einem globalen Fehlermaß bezeichnet wird, sodass eine Interpretation der Ergebnisse in Bezug auf eine zu generierende Fehlermeldung schwierig ist.

Fehlerhaft vs. nicht abgedeckt

Schließlich muss erwähnt werden, dass antizipationsfreie Systeme nur schlecht zwischen inkorrekten sowie zwar korrekten, aber von der Grammatik nicht abgedeckten Eingaben unterscheiden können. Es kann also der Fall auftreten, dass zwar ein korrekter Satz eingegeben wurde, dieser Satz aber auf Grund der Grammatik und/oder des Lexikons nicht geparst werden konnte. Im Gegensatz zu antizipationsbasierten Ansätzen, die relativ einfach zwischen nicht parsbar und nicht beschreibbar unterscheiden können, ist das bei diesen Systemen aufgrund des generellen Konzeptes nicht möglich. Gerade für natürlichsprachliche Systeme stellt dieser Aspekt ein gravierendes Problem dar, da grundsätzlich von einer unvollständigen Beschreibung der Daten ausgegangen werden muss. „There is no obvious solution to this problem, so the best we can do in developing ICALL is probably to help learners become aware of why and how they may have to limit their expressive attempts in certain language activities.“(Garrett, 1995, S. 348) Eine Rückmeldung an den Nutzer sollte daher eine entsprechende „Vagheit“ mit einbeziehen.

Obwohl die fehlende Unterscheidungsmöglichkeit in antizipationsfreien Systemen als Nachteil gesehen werden kann, da dem Lerner die Grenzen des Systems nicht klar dargelegt werden können, kann dieser Aspekt für bestimmte Aufgabenstellungen als Vorteil gewertet werden. In dialogorientierten Übungen, wie sie in Promis*D* realisiert sind, bietet es sich an, Parser zu verwenden, die in vielen Fällen eine Analyse für die Fortsetzung des Dialogs liefern können, auch wenn damit eine womöglich korrekte Eingabe eines Lerners als falsch erklärt wird. Die abstrakte Beschreibung eines Satzes durch das Analyseergebnis ist für die weitere Verarbeitung beziehungsweise Interpretation in nachfolgenden Modulen wie zum Beispiel der Semantikkontrolle oder der Dialogsteuerung wünschenswert. Nur so kann von der Nutzung von Schlüsselwörtern zur Interpretation der Eingabe mit deutlich reduzierter Sicherheit bezüglich der Korrektheit der Interpretation abgesehen werden und es können auf der Basis der abstrahierenden Beschreibung erweiterte Möglichkeiten genutzt werden (siehe Kapitel 6). Trotz der Notwendigkeit einer linguistischen Beschreibung darf meines Erachtens nicht daraus folgen, dass der Parser in *jedem* Fall ein wie auch immer geartetes Ergebnis erzeugen sollte. Im Gegensatz zum *robusten* Parsing, bei dem diese Eigenschaft unter Umständen erwünscht wäre, steht hier die Identifizierung und Rückmeldung von Fehlern als Anforderung im Vordergrund.

Zusammenfassung

Der vorliegende Ansatz folgt zusammengefasst also der zweiten Möglichkeit der antizipationsfreien Methode aus folgenden Gründen.

- Da in diesem ICALL-System ein Dialog simuliert werden soll, in der der Lerner beliebige Eingaben machen kann, ist eine (reine) Antizipation

von Fehlern meiner Meinung nach nicht geeignet. Die Gestaltung der Aufgaben beziehungsweise der Dialoge sollte keine Rücksicht auf die Anzahl und Form der identifizierbaren Fehler nehmen, wie das zum Beispiel in Krüger-Thielmann (1992) der Fall ist.

- Umgekehrt sollte es aber auch nicht so sein, dass zwar die Aufgabenstellung relativ weit gefasst, aber die linguistische Abdeckung nur beschränkt ist. Wenn möglich, sollte das System in der Lage sein, eine umfassende Grammatik zu verwenden.

- Wenn die Fehlererkennung auf dem Verfahren und nicht auf der Grammatik basiert, besteht die Möglichkeit, linguistische Daten, das heißt die Grammatik und das Lexikon unabhängig vom Einsatz in einem ICALL-System entwickeln zu können.

Dem Einwand, dass antizipationsfreie Systeme üblicherweise ineffizient und daher praktisch nicht nutzbar sind, wird in Promis*D* mit einer gezielten und klar definierten Einschränkung der Fehlererkennung begegnet. Das „Standard"-Parsingverfahren wurde so umgebaut, dass es im Prinzip in der Lage sein sollte, an beliebigen Positionen beliebige Typen von Fehlern im Rahmen der Grammatik zu erkennen. Allerdings werden zur Beschränkung des Suchraums nicht nur Fehlerbewertungen beziehungsweise Korrekturhypothesen verwendet, sondern es werden auch linguistische Informationen dazu herangezogen. Sinnvollerweise sollten zum Beispiel die Informationen, in welchen Bereichen der Grammatik Fehler auftreten dürfen, an klar definierbaren Stellen eingesetzt werden.

Nach diesen allgemeineren Ausführungen wird nun in den nächsten Abschnitten näher erläutert, wie die Verarbeitung der zwei wesentlichen Beschreibungsebenen der LFG, der PS-Regeln und der F-Strukturen, abläuft, damit der Parsevorgang relativ effizient und sensitiv, also mit der Erhaltung der Fehlerinformationen arbeiten kann.

5.2 Phrasenstruktur-Parsing: Fehlererkennung in der K-Struktur

In diesem Abschnitt soll ein Verfahren vorgestellt werden, das Fehler in der Konstituenz erkennt und gegebenenfalls korrigiert. Dabei möchte ich mich auf die Verwendung von PS-Regeln zur Beschreibung der Wortstellung beschränken, da ja im Rahmen der LFG, wie in Kapitel 3 erläutert, dieses Mittel eingesetzt wird. Dass durch die Verwendung von PS-Regeln neben Aussagen zur Linearisierung auch Aussagen über die Dependenzbeziehungen zwischen Kategorien gemacht werden können, soll hier keine wesentliche Rolle spielen. Meines Erachtens sind die für einen Nutzer relevanten Rektionsbeziehungen wie zum Beispiel die Kasusrektion eines Objekts in der F-Struktur enthalten.

Zu Beginn erfolgen die Erläuterungen zu bisherigen vergleichbaren Arbeiten und nachfolgend zum tatsächlich realisierten Verfahren. Verschiedene Aspekte sind dabei von Relevanz.

1. Die Parsing-Strategie sollte einerseits effizient arbeiten und andererseits ein möglichst breites Spektrum an Fehlern erkennen können. Erforderlich ist dazu ein Abwägen zwischen der Öffnung des Fehlerhypothesenraums zur Identifizierung möglichst vieler Fehler und der geeigneten Einschränkung des Raums zur effizienten Identifizierung der relevanten Fehler.[5]

2. Es stellt sich die Frage, ob Fehler unter Umständen in der ein oder anderen Form antizipiert werden sollten, um damit den Parseprozess zu vereinfachen beziehungsweise zu beschleunigen. Das kann sich zum Beispiel auch in der Gewichtung von Elementen ausdrücken, sodass beispielsweise so genannte „harte“ Constraints markiert werden und dann als Fehlerpunkt nicht in Frage kommen. Der Parser kann mit Hilfe dieser Constraints bestimmte Möglichkeiten vernachlässigen und damit den Parseprozess beschleunigen. Offensichtlich kann auf diese Weise auch eine Gewichtung der identifizierten Fehler erfolgen, um in der Rückmeldung auch tatsächlich einen „präferierten“ Fehler zu präsentieren.

3. Ein weiterer Aspekt, der mit dem vorhergehenden eng verknüpft ist, ist die Frage nach den tatsächlich auftretenden Fehlertypen. Als Konsequenz aus der Analyse annotierter Lernerkorpora fand einerseits bei der Entwicklung der Verfahren insbesondere eine Berücksichtigung der häufigen Fehlertypen statt, während andererseits in Abschnitt 5.4.2 eine Evaluation präsentiert wird, in der sich zeigt, wieviele fehlerhafte Sätze tatsächlich erkannt werden.

4. Zusätzlich ist im vorliegenden Konzept zu berücksichtigen, dass eine bestimmte Grammatiktheorie, die sich als pädagogische Grammatik eignet, als linguistische Grundlage verwendet werden soll. Damit sind dann verschiedene Rahmenbedingungen vorgegeben.

Vier verschiedene Typen von Konstituentenfehlern lassen sich grundsätzlich unterscheiden (Beispiele aus dem HU/FU-Korpus, siehe Abschnitt 5.4.2):

- Einfügungen: Mein Auto rast gegen die Ampel *gewesen*.
- Auslassungen: Das Auto ist gegen __ Ampel gefahren.
- Ersetzungen: Es *gebt* von Unfall.
- Verdrehungen beziehungsweise Verschiebungen: Der Unfall ist *passiert* am Berliner Straße.

[5] Hier besteht offensichtlich eine Korrespondenz zu den Evaluierungsmaßen „Precision“ und „Recall“.

In den mir bekannten Ansätzen haben sich die Autoren fast immer auf die Behandlung der ersten drei Typen von Fehlern konzentriert. Diese Tatsache hängt in erster Linie mit den bei der Fehlerkorrektur berücksichtigten Kontexten zusammen. *Lokale* Korrekturen lassen sich relativ leicht bestimmen, da nur ein beschränkter Kontext berücksichtigt werden muss. Es kann allerdings der Fall eintreten, in dem eine lokale Korrektur nicht zur *global* optimalen Lösung führt. Globale Korrekturen sind aber unter Umständen nur mit großem Aufwand zu berechnen.

Der vermutlich bekannteste Ansatz wurde 1989 von Mellish vorgestellt. Darauf aufbauend wurde in Kato (1994) ein verbessertes Verfahren vorgestellt, das einige Schwierigkeiten von Mellishs Ansatz vermeidet. In beiden Fällen wird der Input-String zuerst bottom-up geparst, sodass am Ende eine Chart mit den Teilen vorhanden ist, die sich aus der Verwendung der Grammatik berechnen lassen. In einem zweiten Schritt wird dann versucht, eine den gesamten Satz überspannende Kante zu generieren. Dieser Schritt besteht aus einem Top-Down-Parsing, das so weit wie möglich auf Informationen aus der schon bestehenden Chart beruht, und das bestimmte Korrekturmechanismen verwendet. Um in diesem Teil ein möglichst effizientes Parsing zu erreichen, experimentiert Mellish mit sechs verschiedenen Heuristiken, die alle zu einer Bewertung von neuen Kanten herangezogen werden (siehe Seite 112).

Die Verbesserung durch Kato besteht vor allem im Einsatz eines bidirektionalen Bottom-Up-Parsers in der ersten Phase und in einer Veränderung der Struktur der Kanten und deren Plausibilitätsbewertung. Nach Katos Angaben wird mit Hilfe der neuen Plausibilitätsberechnung ein Fehler schneller gefunden, da sowohl lokale als auch globale Parameter mit einfließen. Außerdem werden die Regeln zur Integration von aktiven Kanten in der zweiten Phase verändert. Die Verwendung der „Top-Down-Regel“, die Regeln aus der Grammatik integriert, wird in der zweiten Phase reduziert, da alle relevanten Kanten schon in der ersten Phase in die Chart aufgenommen worden sein sollten. Jede Kante der zweiten Top-Down-Phase bekommt zwei Werte zugeordnet, g und h. g ist der Wert, den die Korrekturen bis zur aktuellen Position gekostet haben, das heißt, wie viele Fehler bisher korrigiert worden sind. Der Wert h dagegen entspricht dem, was eine Korrektur vermutlich noch kostet, das heißt, wie viele offene Kategorien noch vorhanden sind. Die Kante, die den kleinsten $g + h$-Wert hat und damit bisher am wenigsten Korrekturen erfordert hat, wird dann als erstes zur Korrektur in der zweiten Phase herangezogen. Die Berechnung der Werte erfolgt innerhalb der Parsing-Regeln der zweiten Phase, weshalb ein Wert an jede neu produzierte Kante weitergegeben werden kann.

Der wesentliche Vorteil eines solchen Verfahrens ist, dass es sehr generell funktioniert. Nahezu beliebige Fehler, das heißt Einfügungen, Auslassungen und Ersetzungen können auf diese Art und Weise erkannt werden. Ein Nachteil besteht in der Anforderung des doppelten Parses. Allerdings wer-

den korrekte Sätze ohne zusätzlichen Aufwand geparst. Schwerer noch wiegt der Nachteil, dass das Parsing-Verfahren nicht in der Lage ist, verschobene Phrasen zu erkennen und an geeigneter Stelle wieder einzufügen. Diese Art von Fehler hat sich in den von mir untersuchten Belegen als relativ häufig erwiesen (Abschnitt 5.4.2). Der Lerner scheint sich der benötigten Elemente in einem Satz bewusst zu sein, kann sie aber nicht in die richtige Reihenfolge bringen. Folgende Beispiele aus dem HU/FU-Korpus zeigen dies.

(5.3) a. *Meines Auto einen Unfall machen.
b. *Ich denke daß mein Deutsch ist ziemlich gut.

In beiden Fällen lassen sich die Konstituentenfehler leicht durch das Verschieben von Elementen beheben. Die zusätzlichen Fehler im ersten Satz (5.3 a) sollen hier nicht betrachtet werden, da es sich nicht um Konstituentenfehler handelt.

(5.4) a. Meines Auto __ machen [einen Unfall].

b. Ich denke daß mein Deutsch __ ziemlich gut [ist].

Heift (1998) markiert, wie erwähnt, jede Linearisierungsregel (Listen als Werte von Attributen) mit Werten, die festlegen, ob eine Wortart an einer bestimmten Position platziert werden darf. Damit können aber in erster Linie nur sehr einfache Sätze analysiert werden, und es müssen potentiell für alle Wortarten Markierungen an allen möglichen Positionen vorgenommen werden. Zusätzlich können bei einer inkorrekten Eingabe keine Aussagen über die tatsächliche Position von Wörtern gemacht werden, sondern nur allgemein über eine Fehlpositionierung eines bestimmten Wortes. Die Informationen, die ein System liefern kann, sollten aber so umfassend wie möglich sein, sodass in diesem Fall unter Umständen dem Lerner relevante Informationen nicht geliefert werden können.

Alle vier Fehlertypen kommen in den von mir untersuchten Fehlerkorpora vor, wobei es allerdings bei den unterschiedlichen Typen deutliche quantitative Unterschiede gibt. Nach dem Heringer-Korpus (siehe Abschnitt 5.4.2) sind nur 5,5% aller Fehler überflüssige Einfügungen im Gegensatz zu 16,5% Auslassungen und 14% Verdrehungen, also in absoluten Zahlen 3-mal mehr Auslassungen und Verdrehungen. Ersetzungen wurden in diesem Korpus nicht gesondert markiert. Es ist allerdings davon auszugehen, dass solche Fehler eher selten auftreten. Ein Lerner müsste ein Wort orthografisch so falsch schreiben, dass sich daraus ein neues, korrektes Wort einer anderen Wortart ergäbe, was intuitiv am ehesten bei Adverb und Adjektiv passieren kann. Das hier entwickelte Verfahren beschränkt sich zur Zeit darauf, Auslassungen, Einfügungen und Verdrehungen zu identifizieren und zu korrigieren.

5.2.1 Verfahren und Implementation

Der Lösungsansatz zur Analyse von Konstitutenzfehlern auf der Grundlage von PS-Regeln basiert auf dem bekannten Algorithmus von Earley (1970) und ist dem in Lyon (1974) vorgestellten Vefahren ähnlich. Eine Erweiterung von Lyon (1974) wurde in Lee et al. (1995) vorgeschlagen. Weiter unten wird darauf eingegangen, worin die Unterschiede bestehen und welche Ähnlichkeiten es gibt. Das Verfahren von Earley hat sich unter anderem als relativ effizient erwiesen, da kein Backtracking notwendig ist. Mit dem Aufbau der Hypothesen, so genannten „states“, in einer Tabelle („Chart“) werden alle Möglichkeiten zur Expandierung gespeichert, die die Grammatik zur Verfügung stellt, und beim Erreichen eines lexikalischen Elements sukzessive „aufgefüllt“. Mit Hilfe der Speicherung aller Zwischenergebnisse entfällt insbesondere das aufwendige Backtracking, bei dem Kanten neu berechnet werden müssten, auch wenn ein Teilbaum schon einmal expandiert worden ist. Weitere grundsätzliche Vorteile dieses Algorithmus bestehen in der Verarbeitung von rekursiven Regeln und der relativ einfachen Möglichkeit zur Integration von „linguistischen“ Merkmalen wie der Optionalität von Kategorien oder dem in Abschnitt 3.2 erwähnten Kleene-Stern.

Das Verfahren von Earley kann wie folgt beschrieben werden.[6] Es werden üblicherweise drei Prozeduren verwendet (in Klammer alternative Bezeichnungen) und so oft angewendet, wie es relevante Kanten in der Chart gibt:

- *Predictor* (Expand): Diese Prozedur schlägt eine neue, aktive Kante für die Chart vor und fügt sie gegebenenfalls hinzu. Diese Operation wird auf alle Kanten der Chart angewandt, die ein noch nicht expandiertes nicht-terminales Element enthalten.

  Für jede Kante der Form $< i, j > A \rightarrow \alpha \cdot B\gamma$ und für jede Regel der Form $B \rightarrow \beta$ füge eine Kante der Form $< j, j > B \rightarrow \cdot\beta$ zur Chart hinzu. $< i, j >$ stellt den Bereich der Eingabekette dar, für den die Kante eine Beschreibung liefern kann, und der Punkt „$\cdot$“ in den Regeln markiert die Position, bis zu der Elemente gefunden wurden. Dieses ist der so genannte Top-Down-Teil des Verfahrens, da neue, so genannte aktive Kanten Top-Down hinzugefügt werden, in denen noch nicht erkannte Kategorien enthalten sind.

- *Scanner* (Shift): Der Scanner akzeptiert ein Wort der Eingabe.

  Für das Wort w_j der Kategorie C füge die Kante $< j, j + 1 > C \rightarrow w_j \cdot$ zur Chart hinzu. Auf dieser so genannten passiven Kante basierend können nun im nächsten Schritt passende weitere Kanten bottom-up gefüllt werden, das heißt eventuell passiv gemacht werden, wenn alle Kategorien der rechten Seite einer Regel gefunden worden sind.

[6] Siehe zum Beispiel Naumann und Langer (1994, S. 125). Verschiedene andere Möglichkeiten sind denkbar.

- *Completer* (Closure): Eine Kante mit offenen Elementen wird mit erkannten Elementen passiv gemacht.

  Für jede passive Kante der Form $< j, k > B \rightarrow \beta\cdot$ und die aktive Kante $< i, j > A \rightarrow \alpha \cdot B\gamma$ füge die Kante $< i, k > A \rightarrow \alpha B \cdot \gamma$ zur Chart hinzu.

Falls eine so genannte S-Kante gefunden wurde, die den gesamten Input überspannt, gibt es es eine Beschreibung für die Eingabekette. Als Grundlage für die vorliegende Arbeit diente die Implementierung aus Naumann und Langer (1994), die wiederum auf Dörre (1987) basiert. Dabei handelt es sich ursprünglich um eine Realisierung des Algorithmus mit *Expand-*, *Shift-*, *Complete-*, und *Closure*-Prozedur für unannotierte PS-Regeln. Das System verwendet eine unterteilte Chart mit aktiven und passiven Kanten in unterschiedlichen Formaten. In einem Teil der Chart werden die aktiven Kanten mit Hilfe der *Closure*-Prozedur gespeichert, die nur die jeweils unmittelbaren Töchterknoten enthalten und im zweiten Teil werden die passiven Kanten mit Hilfe der *Complete*-Prozedur gespeichert, die die vollständigen Teilbäume enthalten. Das Lexikon ist von den Syntaxregeln getrennt.

Der Parser wurde für Promis*D* so erweitert, dass auch mit F-Strukturen annotierte PS-Regeln verarbeitet werden können. Die F-Strukturen werden parallel zum Aufbau des Baumes konstruiert. Dieses ist nicht selbstverständlich, wie zum Beispiel in Maxwell und Kaplan (1996) gezeigt wird, in deren Ansatz in einem Parser zur LFG-basierten Analyse die Trennung von K- und F-Struktur vorgenommen wurde, um die effizienten Methoden im PS-Parsing ausnutzen zu können. Zuerst werden dabei die K-Strukturen aufgebaut und anschließend die F-Strukturen berechnet. Da es in dem hier beschriebenen Verfahren allerdings zu Situationen kommen kann, in denen ein Feature die Anwendung einer PS-Regel beschränkt, ist die Anwendung des „geteilten“ Parsings von Maxwell und Kaplan nicht geeignet, eine Lösung zu finden. Beispielsweise würde die NP-Regel für artikellose Substantive bei der Verwendung des Ansatzes von Maxwell und Kaplan (1996) ohne die Beachtung der F-Struktur zum Aufbau einer Chart führen, die keine Fehlerhypothese für einen fehlenden Artikel enthält und daher auch keine Korrektur liefern könnte, wenn ein Look-Ahead verwendet wird (siehe Abschnitt 5.3.2, Seite 145). Ohne den Einsatz eines Look-Aheads würde aber der Einsatz der Trennung der Berechnung keinen Sinn machen, da auf diese Weise die Anzahl der Kanten in der Chart erheblich reduziert werden kann. Diese Schwierigkeit gilt insbesondere auch für das Deutsche mit freierer Wortstellung als dem Englischen, wenn man annimmt, dass die Wortstellung dann eher durch Features, das heißt durch lexikalische Eigenschaften beschränkt wird. Mit Hilfe der sofortigen Analyse der Feature-Strukturen sollte ein Teil der zu generierenden Kanten ausgeschlossen werden können.

Fehlerhypothesen

Um nun Fehler in der Eingabe diagnostizieren zu können, wird hier wie auch in Lyon (1974) und später in Lee et al. (1995) eine Änderung der *Scan*-Prozedur vorgenommen, sodass Kanten zur Chart hinzugefügt werden, die zwar nicht durch die verwendete Grammatik lizensiert sind, die aber eine mögliche, lokale Fehlerquelle beschreiben. Dabei wird der *Scan*-Schritt so erweitert, dass zusätzlich zur passiven Kante mit der Kategorie des Wortes ($< j, j+1 > C \rightarrow w_j \cdot$) weitere Kanten hinzugefügt werden, die die Hypothese eines Fehlers realisieren. Die ursprüngliche Darstellung wurde an die hier verwendete angepasst. Der zusätzlich in einer Kante enthaltene Wert e stellt einen mit der Kante assoziierten Fehlerwert dar, der im Fehlerfall als Strafmaß erhöht wird.

- Für Ersetzungen („mutation-error hypothesis“) füge die Kante $< j, j+1 > C \rightarrow w_x \cdot, e+1$ mit $w_j \neq w_x$ zur Chart hinzu.
- Für Einfügungen („insertion-error hypothesis“) füge die Kante $< j, j+1 > C \rightarrow \cdot w, e+1$ zur Chart hinzu.
- Für Auslassungen („deletion-error hypothesis“) füge die Kante $< j, j > C \rightarrow w \cdot, e+1$ zur Chart hinzu.

Nach der vollständigen Integration einer terminalen Kante in die Chart kann die Fortsetzung des Aufbaus der Chart auf die Kanten mit den niedrigsten Fehlerwerten beschränkt werden. Auf dieser Basis werden dann Lösungsstrukturen mit möglichst geringen Fehlerwerten ausgewählt. Aufbauend auf diesem Verfahren wurde von Lee et al. (1995) eine Erweiterung vorgeschlagen, in der dieselbe Methode der Speicherung von Fehlerhypothesen zusätzlich auf die *Completer*-Phase angewandt wird. Dadurch ist der Parser in der Lage, auch den fehlerhaften Strukturen auf Phrasenebene eine Beschreibung zuzuweisen, beziehungsweise sie zu korrigieren.

Wie oben ausgeführt, ergibt sich dadurch zwar die Möglichkeit zum Parsen eines beliebigen Textes mit einer kleinen Grammatik, aber die Kosten zur Berechnung einer minimalen Lösung sind außerordentlich hoch, da für jeden *Scan*-Schritt ungefiltert Kanten mit Fehlerhypothesen zur Chart hinzugefügt werden. Die von Lee et al. (1995) vorgeschlagene Ausdehnung auf nicht terminale Knoten vergrößert die Chart zusätzlich. Dieses sehr allgemeine Verfahren mag für relativ kleine und „reine“, das heißt annotationslose Phrasenstruktur-Grammatiken erfolgreich angewendet werden. Allerdings gibt es durch die Annotationen an den Regeln in der LFG-Theorie eine zusätzliche Erweiterung, die berücksichtigt werden muss (cf. Maxwell und Kaplan, 1996). Außerdem ist das Ziel dieser Arbeit nicht die Entwicklung eines Verfahrens zum *robusten* Parsing, sondern zum *sensitiven* Parsing. Sowohl über die Registrierung der Fehler beziehungsweise Korrekturen, die zu einer Strukturbeschreibung eines Satzes führen, als auch über Möglichkeiten zur Reduzierung der Kantenanzahl, wird in Lee et al. nichts gesagt.

Schließlich muss die Erweiterung zur Erkennung von Verdrehungs- beziehungsweise Verschiebungsfehlern angeführt werden, die sich in den Korpora wie erwähnt als häufig herausgestellt haben. Dieser Typ von Fehlern wurde sowohl in Lyon (1974) als auch in Lee et al. (1995) nicht berücksichtigt.

Die grundlegende Idee ist also, dass nach dem Abschluss des *Expand*-Schrittes alle aus der Grammatik verfügbaren Hypothesen generiert worden sind. Diese Hypothesen legen fest, welche terminalen Kategorien als nächstes in der Eingabe vorhanden sein sollten. Wenn die Hypothesen nicht erfüllt werden, können dann Maßnahmen zur Korrektur beziehungsweise zur Hypothesenerfüllung folgen. Das bedeutet auch, dass keine Erreichbarkeitstafel für den Einsatz eines Look-Ahead eingesetzt werden kann, da ja gerade dadurch die Generierung von Hypothesen auf das tatsächlich vorhandene nächste Wort in der Eingabe eingeschränkt wird.

Um die Generierung von beliebigen Fehlerhypothesen einzuschränken, kann es sich also anbieten, die in der Chart enhaltenen Hypothesen zu betrachten, damit nur für die Fälle eine Hypothese generiert wird, für die die Grammatik auch eine Beschreibung liefern kann. Darüber hinaus kann zusätzlich die Generierung an sich beschränkt werden, in dem nur in bestimmten Fällen überhaupt Hypothesen generiert werden. Die allgemeine Idee ist, dass die Struktur der Chart Hinweise darauf geben kann, wann die Einführung einer Fehlerhypothese sinnvoll ist, sodass zwar auf diese Weise unter Umständen bestimmte Fehler nicht erkannt werden können, die Anzahl der Kanten in der Chart sich aber erheblich reduzieren lässt. In Abschnitt 5.4.1 wird darauf noch einmal näher eingegangen. Dieser zweite Aspekt soll im Folgenden näher beleuchtet werden.

Zwei wesentliche Fälle, in denen ein Fehler angenommen werden kann, können unterschieden werden. Allgemein gesehen wird die Chart daraufhin untersucht, ob nach der Integration des aktuellen Wortes (*Shift*) und der darauf aufbauenden Hypothesengenerierung (*Closure + Expand*) eine Möglichkeit zur Integration des folgenden Wortes besteht.

1. Das aktuelle Wort konnte nicht in die Chart integriert werden, das heißt, es gibt keine aktive Kante in der Chart, die ausgehend vom aktuellen Wort eine offene Hypothese enthält.

   $(< j, j > A \rightarrow \cdot B) \notin CHART_j$

2. Es gibt zwar aktive Kanten, die das aktuelle Lexem überspannen, aber keine Hypothese lässt sich mit dem folgenden Wort verbinden.

   $(< n, j > A \rightarrow \alpha \cdot B) \in CHART_j,\ not(LINK(B, w_{j+1}))$

Die folgenden zwei konkreten Beispiele können das Konzept verdeutlichen.

(5.5) Beispielregeln: S → NP VP S → adv Sinv Sinv → v NP
NP → prn VP → v (NP)

Beispieleingabe 1: $_0$ *sehe* $_1$ *dich* $_2$
Das Wort *sehe* ($< 0, 1 >$ v $\rightarrow$ sehe $\cdot$) wurde zwar als terminales Element in die Chart aufgenommen, aber auf dieser Annahme konnten keine weiteren Hypothesen aufgebaut werden, das heißt die Kante ($< 1, 1 >$VP $\rightarrow \cdot$ v NP) wurde nicht in die Chart aufgenommen.

Beispieleingabe 2: $_0$ *heute* $_1$ *ich* $_2$ *lache* $_3$
Die Kanten ($< 0, 1 >$ S $\rightarrow$ adv $\cdot$ Sinv) und ($< 1, 1 >$ Sinv $\rightarrow \cdot$ v NP) wurden zwar in die Chart aufgenommen, aber es besteht kein Link von „Sinv“ beziehungsweise „v“ auf „prn“, der Kategorie von *ich*.

Damit ist ein „Trigger“ eingerichtet, der eine potentiell fehlerhafte Eingabe erkennt. Es muss aber noch geklärt werden, in welcher Form und vor allem, an welcher Position eine Korrektur der Konstituentenabfolge geschieht, damit eine Analyse erfolgreich ist, da ein Fehler in der Konstituenz möglicherweise auch bereits vor der aktuellen Position, an der sich der Parser befindet, angenommen werden kann. Wie zu Beginn des Abschnitts angemerkt, findet in dieser Arbeit eine Beschränkung auf die Erkennung und Korrektur von Auslassungs-, Einfügungs- und Verschiebungsfehlern statt, da sich alle drei Fehlertypen in den Korpora als relativ häufige Konstituentenfehler gezeigt haben (siehe Abschnitt 5.4.2).

Auslassungsfehler

Um einen möglichen Auslassungsfehler zu korrigieren, wird eine passive Kante zur Chart hinzugefügt, die durch eine Hypothese, das heißt eine aktive Kante und außerdem durch eine von der Grammatik unabhängige Liste von möglichen Kategorien lizensiert sein muss. Diese Liste ist wiederum aus den erwähnten Korpora gewonnen worden, wobei sich gezeigt hat, dass die bei weitem am häufigsten ausgelassenen Elemente im Heringer-Korpus Subjekte[7] und im HU/FU-Korpus Artikel sind. Dieser Fehlertyp kann durch die Einfügung eines „virtuellen“ Wortes korrigiert werden, das heißt, es wird eine Kante ($< j, j >$ prn $\rightarrow$ _ $\cdot$) oder ($< j, j >$ det $\rightarrow$ _ $\cdot$) eingefügt (siehe Seite 121). Das Wort wird allerdings nicht an der aktuellen Position eingefügt, sondern es wird die Position gewählt, die sich aus dem Beginn der längsten passiven Kante ergibt, die an der aktuellen Position endet. Gesonderte Erläuterungen zu dieser empirisch bestimmten Heuristik folgen unten, da ein identischer Mechanismus auch für Einfügungsfehler genutzt wird. Weitere Elemente außer den von der Valenz geforderten Aktanten und Artikeln sind alle Arten von konjunktionalen Elementen, wie zum Beispiel *dass* und *um*.

[7] ca. 12% aller Auslassungen. Die Erkennung von Valenzfehlern läuft tatsächlich in den meisten Fällen über die Anwendung des Vollständigkeits- und des Kohärenzprinzips: siehe Seite 128f.

Insbesondere Substantive werden dagegen nicht als Korrekturhypothese eingefügt. Angemerkt werden kann hier außerdem, dass mit der Speicherung einer Kante als Fehlerhypothese in der Chart zwei verschiedene Versionen des Satzes analysiert werden: einmal mit der Phrase an der ursprünglichen Position und einmal in der modifizierten Form. Das gilt auch für die Identifizierung der nachfolgend erläuterten Einfügungs- und Verschiebungsfehler.

Einfügungsfehler

Für diesen Fall wird die Hypothese aufgestellt, dass in einem Satz ein Wort zu viel enthalten ist, was sich beispielsweise in den häufiger auftretenden Fällen wie kontrahierte Präpositionen plus Artikel oder mehrere Auxiliare in einer VP äußert. Wie auch bei den Auslassungfehlern spielt hier zur Bestimmung der Korrekturposition wieder die längste passive Kante, die an der aktuellen Position endet, die entscheidende Rolle. Für jedes Wort, das von der Kante ($< j - max, j > A \rightarrow \alpha\beta\gamma \cdot$) überspannt wird, wird eine Hypothese in die Chart aufgenommen, die dieses Wort „löscht“. Dazu wird die Kante für das vorhergehende Wort entsprechend dem auf Seite 121 erläuterten Verfahren verlängert.

Ermittlung der Korrekturposition

Auf die längste passive Kante wird, wie erwähnt, in unterschiedlichen Fällen zurückgegriffen. Erstens kann es vorkommen, dass zwar ein lexikalisches Element mit einer aktiven Kante kombiniert werden kann, aber trotzdem für das folgende Wort keine Hypothese besteht. Das ist in den hier untersuchten Sätzen vor allem bei NPn der Fall, denen ein Artikel fehlt. Der Artikel aber befindet sich immer am Beginn der Phrase, die nicht nur aus einem Substantiv bestehen muss, sondern auch komplex sein kann, also einschließlich Adjektiven, PPn etc. Ein Beispiel aus dem HU/FU-Korpus ist der Satz (5.6 a.), bei dem keine vollständige NP-Kante aufgebaut werden kann. Dieser Fehler tritt auch im Heringer-Korpus relativ häufig auf: Neben fehlendem Subjekt, also Pronomen sowie Präpositionen, die sich aber oft nur aus der Bedeutung ergeben und daher hier nur wenig Relevanz haben, zählen fehlende Artikel zur dritten großen Klasse.

(5.6) a. *Ich habe ein Unfall gesehen auf Berliner Straße.

b. *Nach dem Essen ging ich mit meinem Freund im den Wald spazieren.

c. *Du weißt, dass ich liebe Sky.

d. Du weißt, dass ich glaube, Sky lacht.

Zweitens wird auf dieses Maß zurückgegriffen, wenn der Bereich bestimmt werden soll, in dem vermutlich ein Wort zu viel enthalten ist. Ein

Beispiel ist die Kombination von kontrahierter Präposition und Artikel, wobei entweder die kontrahierte Präposition aufgelöst oder der Artikel gelöscht werden muss. Der Satz (5.6 b.) stammt aus dem Heringer-Korpus, in dem die Löschung von *den* als Korrektur ausreicht, damit eine Analyse des Satzes gelingt. An der Position *im* deutet der 2. Trigger auf einen Fehler hin, da es keine aktive Kante gibt, die den Artikel vorhersagen würde. Nach dem Algorithmus werden alle Wörter der längsten passiven Kante einschließlich des folgenden Wortes gelöscht, sodass im konkreten Fall *dem* gelöscht wird, da die längste passive Kante lediglich die kontrahierte Präposition *im* umfasst.

Drittens wird schließlich die längste Kante gesucht, wenn zwar für alle Elemente im Satz eine passive Kante gefunden worden ist, eine den gesamten Satz überspannende Kante aber fehlt. Ein Beispiel dafür ist der Satz in (5.6 c.) wiederum aus dem Heringer-Korpus, bei dessen Analyse zwar alle Elemente zusammengefasst werden können, aber eine Satzkante sich nicht generieren lässt. Der Satz in (5.6 d.) zeigt, welche Konstruktion, die auch von der Grammatik abgedeckt werden soll, dafür verantwortlich ist.[8] Erst wenn das letzte Verb nicht auftritt, kann auf einen Fehler geschlossen werden. Die längste vorhandene Kante (von *Du* bis *liebe*) lässt es zu, das Verb *liebe* als fehlplatziert zu erkennen. Diese Beispiele zeigen, dass es unterschiedlichste Situationen geben kann, in denen die Ausnutzung einer längsten Kante Hinweise auf die Fehlerposition liefert.

Verschiebungsfehler

Ein Verschiebungsfehler liegt dann vor, wenn eine Phrase zwar zum Satz gehört, sich aber an der falschen Postion befindet und verschoben werden muss. Ein Problem, dass vor allem gelöst werden muss, ist die Bestimmung der Elemente, die verschoben werden müssen, wobei die Beobachtung gemacht werden kann, dass zumeist nicht nur einzelne Wörter, sondern komplette Phrasen verschoben beziehungsweise vom Lerner falsch positioniert werden, was auch intuitiv plausibel ist. Daraus folgt, dass mit Hilfe eines partiellen Parses, der an der Fehlerposition beginnt, eine Phrase bestimmt werden muss und aus der Menge der so neu erzeugten Kanten die längste passive Kante, die nicht ein Satz, das heißt eine Kante mit der Kategorie `ms` ist, ausgewählt wird. Die Phrase wird dann auf eine „Halde“ gelegt, damit sie zur späteren Einfügung zur Verfügung steht. Die mit Hilfe des partiellen Parses festgelegte Lücke wird gefüllt, in dem die Kante mit dem vor der Lücke stehenden Wort so weit „verlängert“ wird, wie die Lücke reicht. Ähnlich wie für einen Einfügungsfehler wird also für eine zu verschiebende Phrase an der Position j der Länge n eine Kante $(< j-1, j+n > C \rightarrow w_{j-1} \cdot)$ zur Chart hinzugefügt. Voraussetzung für die Speicherung einer Phrase in einer „Halde“ ist eine Verwaltung, die einerseits den gesonderten Parse-Vorgang

[8] Kommata sind in der Grammatik grundsätzlich als optional kodiert.

von dem ursprünglichen trennt und die andererseits die Elemente der Halde auch nur in Sätze einfügt, aus denen dasselbe Element vorher entfernt worden ist.

Das Beispiel „heute ich lache“ aus Beispiel (5.5) soll zur Verdeutlichung dienen. Hier wird ein Fehler an der Position 1 erkannt, da es keine Regel gibt, die ein Adverb und eine NP zu Beginn eines Satzes lizensiert. Allerdings sollte nicht einfach das dieser Position folgende Wort verschoben werden, sodass im vorliegenden Ansatz an dieser Stelle ein neuer, partieller Parse-Vorgang gestartet wird, mit dessen Hilfe die zu entfernende Kette bestimmt werden kann. In diesem Fall handelt es sich dabei nur um das Pronomen *ich*. Nach der Rückkehr zum eigentlichen Parse-Vorgang wird dann versucht, diese Phrase an geeigneter Stelle, das heißt durch eine aktive Kante lizensiert, wieder in den Satz einzufügen.

Darstellung als Pseudocode

In Abbildung 5.1 auf Seite 127 wird das gesamte Verfahren der SHIFT-Modifikation noch einmal in Pseudocode, also etwas formaler dargestellt. Einige Prädikate, die in diesem Zusammenhang weniger wichtig sind, wie zum Beispiel zur Verwaltung der Halde und zum partiellen Parse, sind weggelassen.

Der Fall 1 ist das Standard-Prädikat: Ein passives Element mit den Angaben zum aktuellen Wort w_j wird zur Chart hinzugefügt (ASSERT) und das CLOSURE-Prädikat testet, ob die Kategorie B des Wortes in einer aktiven Kante verwendet werden kann. Im Anschluss daran wird für den Fall, dass ein Element auf der Halde liegt ($<$`PHRASE`$(P \rightarrow Q\cdot, h) >$) und es eine Hypothese für dieses Element gibt, der gleiche Vorgang für dieses Element wiederholt. Beachtenswert ist, dass dieses Element P, das auch aus Phrasen bestehen kann, „nur“ von j bis j reicht, sodass damit die ursprüngliche Länge des Ausgangssatzes nicht verändert wird.

Anschließend wird die Chart daraufhin untersucht, ob relevante Hypothesen aufbauend auf dem aktuellen Wort gebildet werden konnten, sodass für den negativen Fall der Korrekturmechanismus angewandt werden kann. Für die Hypothese eines Auslassungsfehlers wird zunächst die längste passive Kante gesucht, die an der aktuellen Position endet. Daraufhin wird am Beginn der Kante ein Element der Liste T eingefügt, wenn es an dieser Position eine Hypothese dafür gibt, und es wird in die Chart eingebunden.

Ein Einfügungsfehler wird korrigiert, indem die Kante eines möglicherweise überflüssigen Wortes ($< j, j+1 > B \rightarrow w_j\cdot$) von der vorhergehenden terminalen Kante ($< j-1, j+1 > B \rightarrow w_{j-1}\cdot$) überspannt wird, damit auch eine Interpretation möglich ist, die das überflüssige Wort nicht enthält. Dieses Verfahren wird für jedes in der längsten passiven Kante enthaltene Wort angewandt, da nur eine ungenaue Bestimmung möglich ist, welches Wort gelöscht werden muss, um einen der Grammatik entsprechenden Satz zu

DATEN: Ein Lexikon L; eine Chart C mit Elementen $(< i, j > A \rightarrow \alpha\beta\gamma)$; eine Liste mit einfügbaren lexikalisch-funktionalen Kategorien T, z.B. *prn, det, part* und *cop-v*.
EINGABE: Ein Wort w_j des Eingabesatzes der Länge n mit $1 \leq j \leq n$.
METHODE: Für alle lexikalischen Kategorien B mit $w_j \in B$:

```
                                        % Standard
<ASSERT(< j − 1, j > B → w_j ·) >
<CLOSURE(< j − 1, j > B → w_j ·) >

                                        %Elemente auf der Halde?
Wenn <PHRASE(P → Q ·, h) > mit h < j
 und (< m, j > A → α · P) ∈ C_j
dann <ASSERT(< j, j > P → Q ·) >
     <CLOSURE(< j, j > P → Q ·) >

                                        % 1. Fehlerfall/Trigger
Wenn entweder
    (< j, j > A → · B) ∉ C_j
                                        % 2. Fehlerfall/Trigger
oder
    (< m, j > A → α · B) ∈ C_j, not(LINK(B, w_{j+1}))
                                        % Auslassungsfehler
dann
    <FIND_LONGEST_EDGE(i, j) >
    <ASSERT(< i, i > B → w_i ·) > mit B ∈ T
    <CLOSURE(< i, i > B → w_i ·) >
    Für alle (< i, j > X → Y ·)
          <COMPLETE(i, j, X, Y) >

                                        % Einfügungsfehler
    Für alle (< k, k + 1 > B → w_k ·) mit i − 1 ≤ k ≤ j − 1
        <ASSERT(< k, k + 2 > B → w_k ·) >
        <CLOSURE(< k, k + 2 > B → w_k ·) >
        Für alle (< k, j > X → Y ·)
              <COMPLETE(k, j, X, Y) >

                                        % Verschiebungsfehler
    Wenn j < n − 1
    dann
        <STORE(C_j) >
        <PARSE_&_STORE_PHRASE(P → Q ·, l) >
        <RESTORE(C_j) >
        <ASSERT(< j − 1, l > B → w_j ·) >
        <CLOSURE(< j − 1, l > B → w_j ·) >
```

Abbildung 5.1: Erweitertes SHIFT-Prädikat

erhalten.

Schließlich muss auch der Fall eines Verschiebungsfehlers berücksichtigt werden. Im ersten Teil wird an der aktuellen Position ein partieller Parse-Vorgang wie oben beschrieben gestartet (PARSE_&_STORE_ PHRASE). Damit wird versucht, eine Phrase zu ermitteln, die sich unter Umständen an der falschen Position befindet. Sie wird dann gespeichert, damit sie zu einem späteren Zeitpunkt wieder in den Satz eingefügt werden kann (auf die Halde gelegt). Im zweiten Schritt wird schließlich die vorhergehende terminale Kante wie im Fall des Einfügungsfehlers um die Spannweite der Phrase „verlängert“ $(j-1, l)$, um eine mögliche Analyse des Satzes ohne die Phrase an dieser Position erreichen zu können.

Damit ist der Teil zur Erkennung von Konstituentenfehlern mit Hilfe der Chartmanipulation abgeschlossen. Es werden drei wesentliche Bereiche der Konstituentenfehler – Auslassungen, Einfügungen und Verschiebungen – abgedeckt, die zudem noch mit einer kontrollierten Erweiterung der Chart einhergehen. Ein Trigger löst dabei den Korrekturmechanismus aus, der im Wesentlichen die aktuelle Phrase verändert oder sie auf eine Halde legt, um sie später wieder in den Satz einzufügen. In jedem Fall wird eine Fehlerhypothese nur als Alternative zur Chart hinzugefügt, sodass am Ende des Parse-Vorgangs aus einer Anzahl von unterschiedlich bewerteten Lösungen ausgewählt werden kann. Obwohl es sich hier um eine lokale Methode der Korrektur handelt, kann doch eine Vielzahl von Fehlertypen erkannt werden, was in Abschnitt 5.4.2 näher betrachtet wird.

Interaktion von Konstituentenfehlern und F-Struktur

Es gibt allerdings spezielle Konstituentenfehler im weiteren Sinne, die auf die beschriebene Weise nicht erkannt werden können. Es handelt sich um Auslassungen von ganzen Aktanten, also eher Valenzfehler, wobei aber die tatsächliche Satzstruktur nicht den PS-Regeln widerspricht. Dieser Typ von Fehler kann aber in der F-Struktur deutlich gemacht werden, sodass er einerseits als solcher identifiziert und andererseits dem Lerner gemeldet werden kann.

Valenz wird in der LFG mit Hilfe von zwei wichtigen Prinzipien zur Form von F-Strukturen geregelt: dem Vollständigkeits- und dem Kohärenzprinzip (Wiederholung von Seite 50).

- Vollständigkeit: Eine F-Struktur ist vollständig, gdw. alle vom Prädikat regierten Argument-Funktionen als Attribute vorhanden sind.
- Kohärenz: Eine F-Struktur ist kohärent, gdw. alle als Attribute vorhandenen grammatischen Argument- Funktionen von einem Prädikat regiert werden.

Auch im Bereich der Valenz von Prädikaten werden häufig Fehler gemacht, die das System idealerweise erkennen können sollte. Wenn es ein

Prädikat gibt, das bestimmte grammatische Funktionen subkategorisiert und außerdem bestimmte Funktionen in die F-Struktur integriert werden sollen, die nicht in der Funktionsliste enthalten sind, kann ein Fehler in der Kohärenz umgehend erkannt werden. Die Vollständigkeit dagegen kann erst zum Abschluss des Parsings bestimmt werden. Am folgenden Beispiel (5.7) lässt sich dieses erläutern.

(5.7)
$$\begin{bmatrix} \text{PRED} & = \text{‘ÖFFNEN <(SUBJ,}\boxed{\text{DIROBJ}}\text{)>’} \\ \text{TENSE} & = \text{pres} \\ \text{SUBJ} & = \begin{bmatrix} \text{NUM} & = \text{sg} \\ \text{PRED} & = \text{‘Mann’} \end{bmatrix} \end{bmatrix}$$

*Der Mann öffnet.

In der F-Struktur in Beispiel (5.7) lässt sich erkennen, dass es kein Attribut DIROBJ mit einer eingebetteten F-Struktur gibt. Aus dieser Erkenntnis heraus lässt sich leicht die Fehlermeldung produzieren, dass das Verb *öffnen* ein direktes Objekt verlangt, das aber anscheinend nicht realisiert worden ist. Ein ähnlicher Fehler, der häufiger in den Korpora auftauchte, wird durch das folgende Beispiel aus dem Heringer-Korpus verdeutlicht.

(5.8) *Plötzlich geht zum Verkäufer.

Durch die Verwendung eines Adverbs zu Beginn des Satzes verschiebt sich die Stellung des Subjektes auf die Position nach dem finiten Verb. In der in Promis*D* konstruierten Grammatik ist die Position des Subjekts nach dem Verb optional, sodass bei der Analyse des Beispiels (5.8) kein Element nach dem Verb eingefügt wird und trotzdem – für den Lerner verständlich – am Ende des Parses bei der Überprüfung der F-Struktur mit Hilfe der Vollständigkeits- und Kohärenzprinzipien das Fehlen des Subjekts festgestellt wird. Dasselbe Verfahren dient zur Erkennung von allen subkategorisierbaren Einheiten wie zum Beispiel auch PPn, wenn sie nicht als obligatorisch durch die Regeln vorgegeben werden.

Fehlerspeicherung und -bewertung

Schließlich muss betont werden, dass bei diesem Verfahren der Chartmanipulation die finale Strukturbeschreibung nur die korrigierte Version darstellt, sodass im Gegensatz zur Fehlererkennung in den F-Strukturen, in denen die Fehlerursache und damit die Erklärung festgehalten wird, an der abschließenden Beschreibung nicht der „Fehler“ abgelesen werden kann. Aus diesem Grund ist eine Erweiterung des Kanteninhalts notwendig, damit die Änderungen in der Konstituenz nach dem Parsen zur Verfügung stehen. Außerdem muss eine Bewertung der Korrektur erfolgen, damit nach dem Parse eine Disambiguierung der Ergebnisse aufgrund dieser Bewertung erfolgen kann und im einfachsten Fall die Kante am besten bewertet wird, die die

wenigsten Korrekturen enthält. Jedes Element der Chart enthält außer den üblichen Angaben also auch noch eine Position, in der ein Fehlerwert, das heißt eine Bewertung der Korrektur gespeichert wird, sowie eine Position, in der die Änderungen in einer Liste festgehalten werden. Der folgende Satz mit der dazugehörigen Kante einer Fehlerhypothese kann das verdeutlichen.

(5.9) $_0$ Mann $_1$ lacht $_2$
($< 0, 0 > det \rightarrow \_ \cdot$, 2, [[ins,0,det]])

Der Satz benötigt an der ersten Position einen Artikel, um syntaktisch korrekt zu sein. Dazu wird eine Kante zur Chart hinzugefügt, die den Teil von 0 bis 0 überspannt, die die Kategorie *det* hat, die den Fehlerwert 2 zugewiesen bekommen hat und die eine Angabe mit sich führt, aus der hervorgeht, dass an der Position 0 ein *det* eingefügt worden ist ([ins,0,det]). Die letzteren Angaben werden beim Parsen der Eingabe, die dieses Element umfasst, im Baum nach oben weitergereicht, sodass die Informationen schließlich beim Erreichen der Kante, die den gesamten Satz überspannt, zur Verfügung stehen. Falls weitere Fehler im Satz enthalten sind, wird der Fehlerwert je nach Fehlertyp entsprechend erhöht. Die Speicherung von Informationen über Verschiebungs- und Einfügungsfehler geschieht in der gleichen Art und Weise, wobei die Korrektur eines Verschiebungsfehlers geringer bestraft wird als die Korrektur durch Einfügung oder Auslassung, um auf diese Weise eine Bevorzugung des Satzes zu erhalten, der mit den wenigsten Änderungen korrigiert werden konnte. Da zur Speicherung der Angaben eine Liste verwendet wird, können auch Informationen über multiple Fehler aufgenommen werden.

5.2.2 Problematische Fehlertypen

In diesem Abschnitt soll ein Bereich angesprochen werden, der sich bei genauem Hinsehen als nicht leicht handhabbar erwiesen hat. Es handelt sich um Sätze, in denen eine Phrase bei Betrachtung des gesamten Satzes dem Sinn nach offensichtlich falsch platziert ist, an der tatsächlichen Position aber morphosyntaktisch korrekt ist. Konstruierte Beispiele dafür sind:

(5.10) a. ?Die Frau auf dem Marktplatz steht.[9]
b. Die Frau steht auf dem Marktplatz.

(5.11) a. ?Der Mann der Frau hilft.
b. Der Mann hilft der Frau.

Die Behebung eines Konstituentenfehlers, bei dem eine Phrase sich an einer Position befindet, die syntaktisch zulässig ist aber trotzdem bei der Betrachtung des ganzen Satzes erkennbar an eine andere Position verschoben

[9]Beispiel von W. Menzel pc.

werden müsste, lässt sich nur mit Schwierigkeiten innerhalb eines Konzeptes realisieren, das auf einem einmaligen Parse beruht. Zwei Aspekte spielen dabei meines Erachtens eine Rolle.

Erstens ist nicht klar, ob syntaktische oder semantische Wohlgeformtheitsbedingungen relevant sind. In dem Beispiel (5.10) handelt es sich um einen „Fehler“, der nur mit präzise ausgestalteten Weltwissen korrigiert werden kann. Hier ist eine Situation vorstellbar, in der der Satz voll gültig und korrekt ist: nämlich dann, wenn betont werden soll, dass die Frau im Gegensatz zu einer laufenden Frau steht. Da es in dem hier vorgestellten Ansatz allein um die Behandlung von morphosyntaktischen Fehlern geht, ist eine Fehleridentifizierung daher so nicht möglich. Wenn man bei der ursprünglichen LFG-Perspektive und seiner modularen Sichtweise bleibt, müsste nach der Diagnose des Fehlers in einem Semantikmodul in einer getrennten Behandlung von morphosyntaktichen und semantischen Fehlern ein Feedback in das Morphosyntaxmodul erfolgen. Als zweite Option zur Behandlung dieses Fehlers widerspricht die Integration von semantischem Wissen in die F-Struktur der linguistischen Intuition, die mit der F-Struktur verknüpft ist, obwohl diese Möglichkeit natürlich prinzipiell gegeben ist. Menzel und Schröder (1998a) deuten genau diese Strategie für den Umgang mit derartigen Fehlern an.

Der zweite Aspekt betrifft die Möglichkeiten, ein Verfahren zu entwickeln, dass diese Art von Fehler korrigiert, wenn man davon ausgeht, dass es sich insbesondere im konstruierten Beispiel (5.11) im weitesten Sinne noch eher um einen morphosyntaktischen Fehler handelt. In diesem Beispiel liegt vermutlich ein Valenzfehler vor, da das Verb noch ein Objekt als Ergänzung verlangt. Aber auch hier ist eine Situation vorstellbar, in der der Mann einer bestimmten Frau existiert, der in einer aus dem Kontext bekannten Situation hilft, wobei das Objekt, dem geholfen wird, in einer Art Ellipse weggelassen werden kann. Eine Möglichkeit zur Behandlung mit Hilfe des Parsers wäre die „prophylaktische“ Verschiebung der verschiedenen Phrasentypen auf die Halde, auch wenn kein Korrekturprozess mit Hilfe des gezeigten Triggers ausgelöst wird. Ein Satz hätte dann verschiedene Realisierungsmöglichkeiten entsprechend den Mutationen, die der Parser liefern kann. Der Satz (5.11 a.) ohne das Objekt von *helfen* wäre dann stärker fehlermarkiert als der Satz (5.11 b.) mit dem Dativ-Objekt und daraus ergäbe sich dann die Korrektur. Basierend auf dem funktionalen Konzept der LFG kann eine Auswahl der zu mutierenden Phrasen auf den grammatischen Funktionen aufsetzen, die in der F-Struktur ausgedrückt werden. Eine zweite Möglichkeit besteht im Versuch der Korrektur, nachdem der eigentliche Parse-Prozess abgeschlossen ist. Jeder Satz würde entsprechend den grammatischen Funktionen permutiert, bis eine Beschreibung mit einem geringen Fehlerwert erreicht worden ist. Allerdings kann im Zweifelsfall auch hier wieder nur dezidiertes Weltwissen zu einer Entscheidung über die Auswahl eines korrigierten Satzes führen. Abschließend muss betont werden,

dass nur sehr wenige Sätze in dem hier betrachteten Korpus zu den in dieser Art zweifelhaften Sätzen zählen.[10]

In diesem Abschnitt wurde gezeigt, wie auf der Basis eines modifizierten Earley-Algorithmus Konstituentenfehler durch gezielte Chartmanipulation erkannt werden können. Die hier gezeigte Methode zeichnet sich insbesondere dadurch aus, dass sie gegenüber unrestringierten Ansätzen wie Lyon (1974) oder Lee et al. (1995) gezielt Maßnahmen zur Einschränkung der Fehlerhypothesen einsetzt und dass zusätzlich auch falsch positionierte Phrasen erkannt werden können, wenn der Satz durch ein Verschieben der Phrase nach „hinten“ korrigiert werden kann. Die Effizienz ergibt sich aus der Annahme eines Fehlers nur dann, wenn es tatsächlich Hinweise auf einen Konstituentenfehler in der Chart gibt, das heißt, wenn ein Trigger die Einführung einer Hypothese auslöst. Unter Zuhilfenahme des Vollständigkeits- und des Kohärenzprinzips, die sich allerdings auf die F-Struktur beziehen, werden außerdem Valenzfehler erkannt. Eine Evaluation folgt in Abschnitt 5.4, in der genauer geklärt wird, welche Fehlertypen erkannt werden und welche nicht.

5.3 Unifikation: Fehlererkennung in der F-Struktur

In diesem Unterkapitel soll es um die Erkennung von Fehlern in der Kongruenz, Rektion, sowie Bereichen der Verbeigenschaften wie zum Beispiel der Finitheit gehen. Aus der strukturellen Perspektive geht es um Fehler, die beim Aufbau der F-Struktur auftreten. Es soll ein Verfahren gezeigt werden, bei dem sich widersprechende Werte von Attributen in die F-Struktur integriert werden, damit sie am Ende des Parse-Vorgangs zur Erzeugung einer präzisen Fehlermeldung genutzt werden können. Zu Beginn werden einige Vorschläge aus der einschlägigen Literatur vorgestellt und an diesen Beispielen demonstriert, welche Schwierigkeiten sich aus der Anwendung dieser Verfahren in dem hier gewählten Rahmen ergeben. Anschließend wird das entwickelte Verfahren erläutert.

Es hat verschiedene Vorschläge zur Behandlung von sich widersprechenden Werten in Feature-Strukturen gegeben. Sågvall Hein (1998) stellt ein Verfahren im Rahmen des SCARRIE-Projektes vor, in dem Attribute mit atomaren Werten relaxiert werden können. Wenn sich zwei Werte widersprechen, wird nur einer von beiden in das Ergebnis der Unifikation eingefügt. Dabei wird der Feature-Struktur zusätzlich ein Fehler-Attribut hinzugefügt, das einen Wert bekommt, der auf einen Typ von Fehlermeldung hinweist. Mit Hilfe der Instantiierung des Wertes ist das System dann in der Lage, eine allgemeine, vorformulierte Fehlermeldung auszugeben. Über die Werte des sich

[10] Siehe auch die Anmerkungen zu dem Satz „Ich gehe zum Fuß.“ in der Diskussion in Abschnitt 5.4.2, Seite 161.

widersprechenden Attributs können somit keine Aussagen gemacht werden. Für einen Lerner wäre es allerdings sicherlich leichter, den Fehler zu verstehen, wenn er zusätzlich zur Information „Numerusfehler“ auch die Werte mitgeteilt bekommen würde. Außerdem kann immer nur ein einziger Fehler pro Feature-Ebene registriert werden, da auch hier das Fehler-Attribut wie üblich in einer Feature-Struktur mit nur einem einzigen Wert auftreten darf. Auf diese Weise lässt sich vermutlich ein sehr effizientes Parse-Verhalten realisieren, wobei die in der Feature-Struktur enthaltenen Informationen über den Fehler allerdings sehr gering ausfallen.

In Schwind (1988) wird ein Verfahren vorgestellt, bei dem die Werte als Disjunktionen in einer Feature-Struktur zusammengefasst werden (auch in Schwind, 1994). Die Unifikation von zwei Feature-Strukturen ergibt dann als Resultat immer nur eine Feature-Struktur, die unter Umständen an verschiedenen Positionen Disjunktionen als Werte enhält. Die Feature-Strukturen sind auf einfache Strukturen beschränkt, das heißt, Werte können nur atomar sein, was zwar das Verfahren praktikabel macht, aber auch die Kodierung von komplexeren Eigenschaften verhindert. Funktionale Beziehungen wie in der LFG können zum Beispiel nur mit Schwierigkeiten ausgedrückt werden. Außerdem fehlt in dieser Konzeption die Möglichkeit der Bewertung von unterschiedlichen Fehlerquellen, sodass sich also im Fall von mehreren Alternativen auch keine Lösung auswählen lässt. Unter anderem aus diesem Grund muss Schwind im Fall des Satzes „Der Götter zürnen“ einen „Kasus-Filter“ einführen, der die gewünschte Lösung auswählt, da man schließlich davon ausgehen kann, dass es sich bei diesem Satz eben nicht um einen Kasusfehler (Subjekt im Genitiv) handelt, der gefiltert wird, sondern um einen Numerusfehler (Artikel hat falschen Numerus).

Desweiteren wird in den Feature-Strukturen nicht vermerkt, ob es sich bei der Disjunktion tatsächlich um einen Fehler handelt. Wenn die Möglichkeit zur Verwendung von Disjunktionen im Lexikon vorgesehen ist, stellt das sicher auch eine geeignete Methode zur Vermeidung von Redundanzen im Lexikon dar. Dann wäre aber nicht klar, ob eine spezifische Disjunktion von Werten aus dem Lexikon stammt, oder ob es sich um eine Disjunktion als Ursache aus einem Feature-Clash handelt. Ein ähnliches Verfahren wird auch in Fouvry (2003) für die Verwendung in getypten Merkmalsstrukturen (HPSG, Pollard und Sag (1987, 1994)) vorgeschlagen. Hier wird allerdings berücksichtigt, dass die Feature-Strukturen außer mit den sich widersprechenden Werten zusätzlich noch mit Fehlerbewertungen versehen werden müssen.

Ein weiterer Vorschlag zur Behandlung von sich widersprechenden Werten in Feature-Strukturen wird in Carpenter (1993) gemacht. Zwei Typen von Unifikation werden präsentiert, wobei die eine „Credulous Default Unification“ und die andere „Skeptical Default Unification“ genannt wird. Die erste Form der modifizierten Unifikation versucht, im Resultat der Unifikation soviel wie möglich an Informationen zu erhalten. Der Unifikationsme-

chanismus wird so verändert, dass mehrere Feature-Strukturen entstehen, die jeweils so beschaffen sind, dass sie von den ursprünglichen Strukturen subsumiert werden. Die resultierenden Feature-Strukturen sind außerdem in sich konsistent (siehe Beispiel (5.13)), allerdings nicht miteinander kompatibel: Eine Unifikation würde fehlschlagen. Mit Hilfe dieses Verfahrens lassen sich auch komplexe, im Normalfall nicht unifizierbare Feature-Strukturen unifizieren.

(5.12) The result of credulously adding the default information in G to the strict information in F is given by:
$F \overset{<}{\sqcup}_c G = \{F \sqcup G' \mid G' \sqsubseteq G \; is \; maximal \; such \; that \; F \sqcup G' \; is \; defined\}$

Dieser Weg kann im vorliegenden Fall nicht beschritten werden, da die resultierenden Feature-Strukturen unter Umständen vollständig unterschiedliche Informationen enthalten würden. Das folgende Beispiel aus Bouma (1990), zitiert nach Carpenter (1993), verdeutlicht dieses.

(5.13) $$\left[\begin{array}{ll} \text{F} & : \text{a} \end{array}\right] \overset{<}{\sqcup}_c \left[\begin{array}{ll} \text{F} & : \boxed{1}\ \text{b} \\ \text{G} & : \text{b} \end{array}\right] = \left\{ \left[\begin{array}{ll} \text{F} & : \text{a} \\ \text{G} & : \text{b} \end{array}\right], \left[\begin{array}{ll} \text{F} & : \boxed{1}\ \text{a} \\ \text{G} & : \boxed{1} \end{array}\right] \right\}$$

Alle resultierenden Strukturen müssten dann für die weitere Analyse genutzt werden, was zu einem erheblichen Rechenaufwand führen würde. Die resultierenden Strukturen enthalten zudem keinerlei Informationen über die sich widersprechenden Werte. Ein Fehler könnte nur über den Vergleich von Feature-Strukturen erkannt werden, was einen zusätzlichen Aufwand bedeutet. Die „skeptische Unifikation“ beruht auf der Generalisierung der Menge der Resultate der „credulous unification“.

(5.14) $$\left[\begin{array}{ll} \text{F} & : \text{a} \end{array}\right] \overset{<}{\sqcup}_c \left[\begin{array}{ll} \text{F} & : \boxed{1}\ \text{b} \\ \text{G} & : \text{b} \end{array}\right] = \left[\begin{array}{ll} \text{F} & : \text{a} \\ \text{G} & : \bot \end{array}\right]$$

In diesem Fall folgt aus der Unifikation nur eine einzige Feature-Struktur; diese enthält aber nur die nicht konfligierenden Informationen. Beachtenswert ist außerdem die Tatsache, dass die Unifikation in allen Fällen gerichtet ist, das heißt, dass die Symmetrie verloren geht, woraus möglicherweise schwerwiegende Konsequenzen folgen.

Ein weiterer Vorschlag zum Umgang mit nicht klassisch unifizierbaren Strukturen stammt von Vogel und Cooper (1995). Dieser Vorschlag basiert ähnlich wie Carpenters „skeptical unification“ auf der Idee, die sich widersprechenden Werte aus der resultierenden Struktur ganz herauszunehmen und so nach der Unifikation eine einzige Feature-Struktur zu erhalten. Wenn bei der Unifikation sich widersprechende atomare Werte festgestellt werden, wird als neuer Wert für das Attribut der Wert $\bot$ (bottom) eingesetzt. An dieser Stelle weist die resultierende Feature-Struktur nun eine Lücke auf und alle weiteren möglichen Belegungen dieses Attributs mit Werten werden immer in $\bot$ enden. Auf diese Weise wird gewährleistet, dass bei allen Unifikationen sich jeweils nur eine Struktur ergibt. Die Strukturierung

innerhalb von HPSG (Pollard und Sag, 1987, 1994) macht es notwendig, Structure-Sharing zu berücksichtigen. Hier ergeben sich nun zwei Möglichkeiten: Entweder werden die nicht unifizierbaren Werte über die gesamte Struktur verteilt oder sie werden lokal in einzelnen Sub-Feature-Strukturen belassen. Die erste Möglichkeit erlaubt es nicht mehr, die Herkunft der sich widersprechenden Werte zu rekonstruieren. Die zweite Möglichkeit dagegen erzeugt Feature-Strukturen, für die gilt:„the logic of the resulting feature structure is not well understood“ (Vogel und Cooper, 1995, S. 213).

Das Beispiel in Abbildung 5.2 auf Seite 136 zeigt das Ergebnis der Unifikation von zwei HPSG-artigen Feature-Strukturen für den Satz „the drivers uses the seatbelts“. Der Numeruswert der COMP-DTR stimmt nicht mit dem Numeruswert der HEAD-DTR überein. Für die Verwendung in einem System, in dem es gerade auf die Registrierung von Fehlertypen und -positionen ankommt, ist dieser Ansatz aus ähnlichen Gründen wie bei Carpenter (1993) nicht anwendbar. Der Wert des entscheidenden Attributes wird in den Erläuterungen von Vogel und Cooper mit $\perp$ angegeben. Im Beispiel tauchen allerdings die sich widersprechenden Werte in der unifizierten Feature-Struktur nach dem $\perp$ als eine Art Subskript auf, sodass nicht klar ist, ob die Werte erhalten bleiben und für eine Auswertung verwendet werden können, oder ob der Erhalt der Werte nur zur Verdeutlichung angegeben ist. Es ist meines Erachtens notwendig, zusätzlich eine Struktur einzuführen, die die sich widersprechenden Werte und ihre Attribute nach der Herkunft für das Feedback speichert, wenn man die bekannten Eigenschaften des Structure-Sharing erhalten möchte, da ansonsten die erwähnten ungeklärten logischen Eigenschaften definiert werden müssten.

Foster (2000) kritisiert diesen Vorschlag auch und liefert einen eigenen Ansatz. Ähnlich wie im vorliegenden Versuch werden auch bei Foster die sich widersprechenden Werte als eine Art Annotation an die Ursprungs-Feature-Struktur angefügt. Die Relation, die die Werte miteinander verknüpft, nennt sie „inconsistent identity“, die in Abbildung 5.3 (Seite 137) dargestellt ist.

Die Beschränkung besteht hier allerdings in der Behandlung von Kongruenzfehlern in HPSG-ähnlichen Feature-Strukturen. Anzumerken ist, dass in Fosters Ansatz die Behandlung von Widersprüchen in „structure-shared“ Werten erfolgt, wobei dieses Structure-Sharing sich nach der Unifikation verändert. Das widerspricht allerdings den allgemeinen Eigenschaften, wonach bei der Unifikation eigentlich nur Informationen hinzugefügt werden sollen, das heißt, diese Methode ist nicht monoton. Außerdem wird nicht deutlich, was mit eventuell schon vorhandenen Indizes, die ein „structure-sharing“ darstellen, geschehen soll. Ein zweiter, zu kritisierender Aspekt ist die Beschränkung auf Kongruenzfehler, da die Kodierung von Nicht-Kongruenz-Informationen in der HPSG auf eine andere Art gelöst wird. Zur Behandlung von fehlerhaften Eingaben von Fremdsprachenlernern ist das Erkennen weiterer Fehler aber unabdingbar.

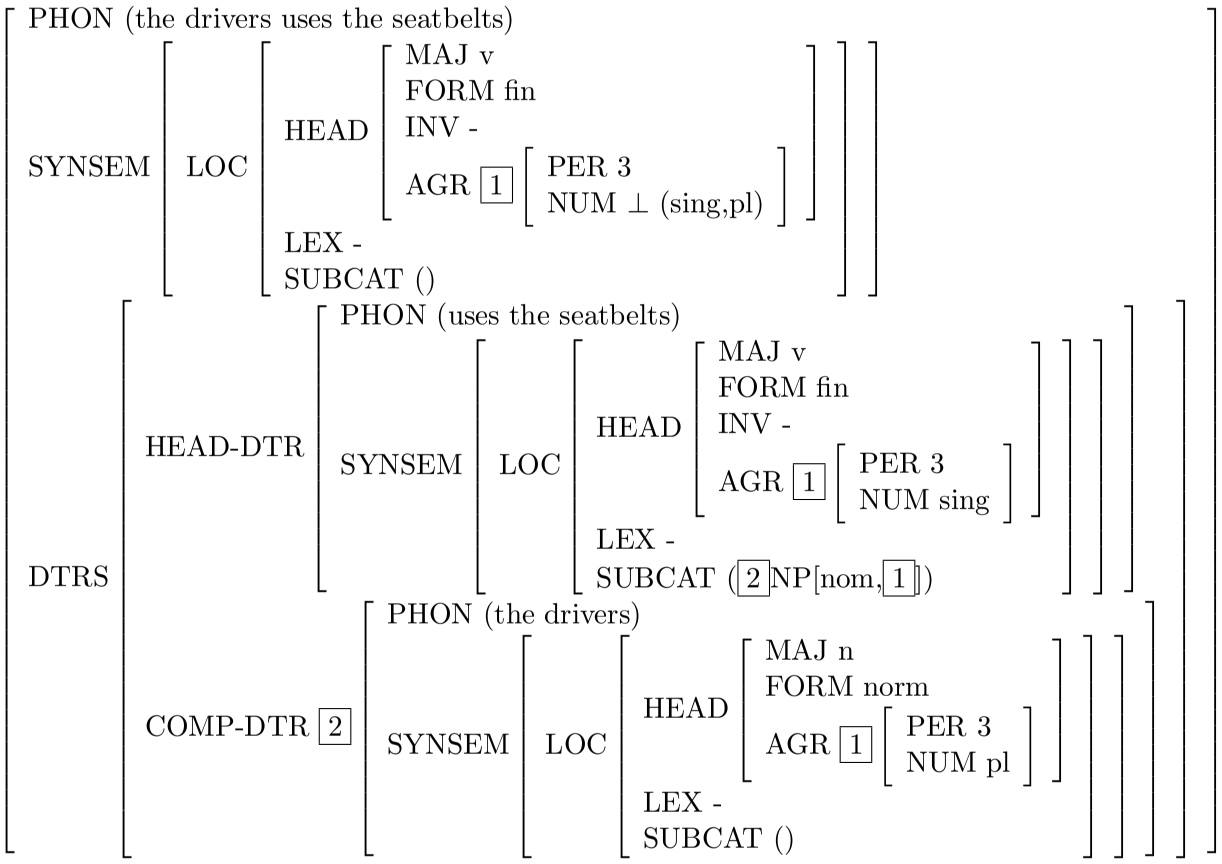


Abbildung 5.2: HPSG-Analyse des Satzes „the drivers uses the seatbelts“ nach Vogel und Cooper (1995)

$$\left[\begin{array}{ll} \text{HEAD-DTR:} & _{noun}\left[\begin{array}{ll}\text{NUM:} & \boxed{1}\ \text{plur}\end{array}\right] \\ \text{SPR-DTR:} & _{det}\left[\begin{array}{l}\text{NUM:}\ \ \boxed{2}\ \text{sing} \\ \boxed{1} \Leftrightarrow_{\top} \boxed{2}\end{array}\right]\end{array}\right]$$

Abbildung 5.3: Feature-Struktur für „this women“ (Foster, 2000)

5.3.1 Verfahren und Implementation

Um einerseits die Verwendung von konsistenten Feature-Strukturen zu gewährleisten und andererseits so wenig wie möglich an Informationen bei der Unifikation von sich widersprechenden Attributen zu verlieren, soll in dem hier vorgestellten Konzept eine zusätzliche Markierung in eine Feature-Struktur eingefügt werden, die diese Informationen aufnehmen kann.[11] Jede komplexe Feature-Struktur, die nicht nur aus Atomen besteht, kann mit einer Markierung, das heißt mit einer Fehlerliste versehen werden, die einen Teil der konfligierenden Attribute und deren atomare Werte enthält. Der andere Teil bleibt bei der Unifikation innerhalb der klassischen Feature-Struktur erhalten. Die folgenden Beispiele (5.15) und (5.16) können dieses verdeutlichen.

$$(5.15)\quad \left[\begin{array}{ll}\text{F} & :\text{a}\end{array}\right] \overset{<}{\sqcup}_{err} \left[\begin{array}{ll}\text{F} & :\text{b} \\ \text{G} & :\text{c}\end{array}\right] = \left[\begin{array}{ll}\text{F} & :\text{a} \\ \text{G} & :\text{c} \\ \hline err & :\left\{\left[\begin{array}{ll}\text{F} & :\text{b}\end{array}\right]\right\}\end{array}\right]$$

$$(5.16)\quad \left[\begin{array}{ll}\text{F} & :\text{a} \\ \text{G} & :\text{c}\end{array}\right] \overset{<}{\sqcup}_{err} \left[\begin{array}{ll}\text{F} & :\text{b} \\ \text{G} & :\text{d}\end{array}\right] = \left[\begin{array}{ll}\text{F} & :\text{a} \\ \text{G} & :\text{c} \\ \hline err & :\left\{\begin{array}{l}\left[\begin{array}{ll}\text{F} & :\text{b}\end{array}\right] \\ \left[\begin{array}{ll}\text{G} & :\text{d}\end{array}\right]\end{array}\right\}\end{array}\right]$$

Die atomaren Werte der Ausgangsstrukturen widersprechen sich zwar bei identischen Attributen, aber in beiden Ergebnissen sind Fehlerlisten als Werte des speziellen Features *err* enthalten, die die widersprechenden Attribut-Wert-Paare aufnehmen, wobei sich für den Fall von mehreren sich widersprechenden Attributen entsprechend mehrere Feature-Strukturen in der Liste ergeben können (Beispiel (5.16)).

Diese Strukturierung hat Ähnlichkeit mit dem in Schwind (1988) vorgeschlagenen Konzept zur Verwendung von Disjunktionen. Allerdings werden im vorliegenden Ansatz die bekannten Schwierigkeiten mit der Behandlung von Disjunktionen vermieden, da es sich den folgenden Definitionen nach eher um eine Konjunktion handelt. Wenn eine Feature-Struktur also eine Fehlerliste enthält, dann wird sie von einer anderen subsumiert, wenn die

[11] Eine erste Ausarbeitung des folgenden Ansatzes mit besonderer Berücksichtigung der verwendeten Grammatiktheorie findet sich in Reuer (2000).

Attribut-Wert-Paare entweder in der Feature-Struktur selber oder in der Fehlerliste enthalten sind. Zudem ist die Definition für Subsumption hier so erweitert, dass auch komplexe Feature-Strukturen erfasst werden.

Subsumption und Unifikation

Notwendig ist zunächst die Definition einer Substitutionsfunktion, die ein Attribut-Wert-Paar gegen ein eventuell in der Fehlerliste enthaltenes Feature austauscht und damit eine Bestimmung der Subsumption ermöglicht. Außerdem muss das Fehler-Feature als ein spezielles Attribut ausgezeichnet werden. Formal gesehen ist die Darstellung hier an Smolka (1992) als Grundlage angelehnt, der einen graphentheoretischen Ansatz zur Definition von Feature-Strukturen verfolgt. Eine umfassende Darstellung der mathematischen Aspekte findet sich in Reuer und Kühnberger (2005), sodass hier nur eine eher intuitive Erläuterung erfolgt.

Definition 5.1 *Eine Substitution Θ angewandt auf eine Feature-Struktur F_{err} ist ein kontravariantes Paar von Ersetzungsfunktionen $\Theta = \langle \Theta_i^{\wedge}, \Theta_i^{\vee} \rangle$, sodass die folgenden Bedingungen gelten:*

(i) $\Theta_i^{\wedge} : [err : \{f_1(x_1), \ldots, f_n(x_n)\}] \mapsto f_i(x_i)$ mit $i \in \{1, \ldots, n\}$, $x_i \in Const$, und $f_i(x_i)$ wird in err gelöscht.

(ii) $\Theta_i^{\vee} : f_i(y) \mapsto [err : \{f_i(y)\}]$ mit $i \in \{1, \ldots, n\}$, $y \in Const$, und $f_i(y)$ wird in der Feature-Struktur F_{err} gelöscht.

Definition 5.2 (Informationserhaltende Subsumption $\sqsubseteq_{err}$) *Eine Feature-Struktur F subsumiert eine Feature-Struktur F' genau dann, wenn gilt:*

- F und F' sind atomar und $F = F'$ oder
- F und F' sind komplex und es gilt:
  - jeder Pfad in F ist auch ein Pfad in F' und entweder
    - der Wert des Pfades in F ist gleich dem Wert des korrespondierenden Pfades in F'

    oder
    - der Wert des Pfades in F ist gleich dem Wert des korrespondierenden Pfades in F' nach Anwendung der Substitution Θ.
  - Alle in der Fehlerliste err von F enthaltenen Elemente sind auch in der korrespondierenden Fehlerliste err' von F' enthalten.
  - Alle Pfade, die sich in F einen gemeinsamen Wert teilen, teilen sich auch in F' einen Wert.

Diese Definition von Subsumption unterscheidet sich im Wesentlichen nur in einem Punkt von den Definitionen beispielsweise in Naumann und Langer (1994): Es muss explizit auf die Elemente der Fehlerliste Bezug genommen werden.

Um eine partielle Ordnung mit Hilfe der Subsumptions-Relation zu erhalten, ist es notwendig, eine Äquivalenz-Relation zu definieren:

Definition 5.3 *Gegeben eine Feature-Struktur F_{err} und eine Feature-Struktur F'_{err} kann eine Äquivalenz-Relation wie folgt definiert werden:*

$$F_{err} \sim F'_{err} \quad \Leftrightarrow \quad F_{err} \sqsubseteq F'_{err} \wedge F'_{err} \sqsubseteq F_{err}$$

Die Äquivalenzklasse einer Feature-Struktur F_{err} wird geschrieben als $[F_{err}]_{\sim}$. Damit lässt sich dann eine Subsumptions-Relation definieren, die eine partielle Ordnung über Äquivalenzklassen von Feature-Strukturen ergibt.

$$[F_{err}] \sqsubseteq_{\sim} [F'_{err}] \quad \Leftrightarrow \quad F_{err} \sqsubseteq F'_{err}$$

Die Definition hat die gewünschten Eigenschaften wie Reflexivität, Transitivität und Antisymmetrie. Die Standarddefinition der Unifikation muss nicht geändert werden, und reicht aus, neue Strukturen zur Analyse zu erzeugen.

Definition 5.4 *Das Ergebnis der Unifikation von zwei Feature-Strukturen $[F_{err}]$ und $[F'_{err}]$ ist definiert als die größte untere Schranke in der Menge der Feature-Strukturen geordnet nach der Subsumptionsrelation.*

Ein wichtiger Aspekt ist, dass die Monotonizität der Unifikation erhalten bleibt, was nicht nur eine erwünschte Eigenschaft der Unifikation ist, sondern auch in der LFG-Theorie angestrebt wird (Falk, 2001, S. 9). Die Definition hat zur Folge, dass prinzipiell jede Feature-Struktur mit jeder Struktur unifiziert werden kann, obwohl die Unifikation im fehlerfreien Fall der Standard-Unifikation entspricht. Mit Hilfe dieses Verfahrens lässt sich nun jeder Widerspruch zwischen atomaren Werten in eine einzige Feature-Struktur integrieren.

Fehlerbewertung

Um eine Gewichtung der resultierenden Strukturen zu erreichen, bietet es sich an, die Kanten in der Chart wie auch bei Konstituentenfehlern mit Fehlerwerten zu markieren, die in erster Linie der Anzahl der sich widersprechenden Werte entspricht. Damit lässt sich neben der reinen Bewertung einzelner Lösungen ein Parsing realisieren, das Kanten mit niedrigen Fehlerwerten präferiert und es lässt sich ein maximaler Schwellwert festlegen,

über den die Anzahl der Fehler in einer Struktur nicht hinausgehen darf. Feature-Strukturen, die „sinnlose“ Kombinationen von Lexemen darstellen, werden auf diese Weise von vornherein unterdrückt.

Zusätzlich zur Disambiguierung der Strukturen für die Rückmeldung und zur Einschränkung des Suchraums mit Hilfe der Anzahl der Fehler lassen sich auch die Attribute selbst einschränken. In der Grammatik von Promis$D$ ist eine Liste der Features enthalten, bei denen überhaupt ein Fehler auftreten darf, sodass die Fehlererkennung auf die Eigenschaften eingeschränkt wird, die aus linguistischer beziehungsweise sprachdidaktischer Sicht sinnvoll sind. Die folgende Tabelle 5.1 stellt die 12 relaxierbaren Features dar.

CASE	Kasus	MOOD	Finitheit
NUM	Numerus	MODUS	Modus
PERS	Person	COMPTYPE	Verbkomplementtyp
GEN	Genus	MOOD2	Finitheit Komplement
AUXTYPE	Auxiliartyp	PCASE	gram. Funktion einer
SPECTYPE	Artikeltyp für		subkat. PP
	Adjektivflexion		(LFG-spezifisch)
		MODE	Präpositionalkasus

Tabelle 5.1: Liste der zulässigen Fehler-Features

Ein weiterer Vorteil ergibt sich aus der Ausnutzung der hierarchischen Struktur der F-Struktur mit der Markierung von Kongruenz- und Rektionsfehlern, die in der tatsächlichen Implementation realisiert ist. Wenn ein Fehler bei der Unifikation einer eingebetteten Feature-Struktur auftritt, wird der Fehler nach außen propagiert, das heißt, in die Rahmen-F-Struktur des Unifikationsergebnisses wird eine Markierung eingebaut, die auf einen Fehler in der Substruktur hinweist. Das Verfahren zur Unterscheidung des Auftretens des Fehlers wird durch die rekursive Abarbeitung der Features möglich. Beispielsweise kann der oben angeführte Satz „Der Götter zürnen“ (Seite 133) ohne Hilfsmittel wie Kasus-Filter analysiert werden.

Die Analyse mit einem Numerusfehler innerhalb der NP erhält nur einen Fehlerwert von 1: Nur bei der Unifikation der F-Strukturen von Artikel [NUM = sg] und Substantiv [NUM = pl] wird der Fehlerwert erhöht, sodass diese Analyse bei der Generierung des Feedbacks bevorzugt wird und damit eine sinnvolle Fehlermeldung ausgegeben werden kann (Beispiel (5.17)).

Die Analyse, nach der es sich um ein Subjekt im Genitiv handelt, erhält dagegen nach dem erläuterten Konzept der Fehlerpropagierung „nach außen“ bei der Unifikation von Subjekt [SUBJ = [CASE = gen]] und Verb [SUBJ = [CASE = nom]] einen Fehlerwert von 2: einen Fehlerpunkt für Markierung des falschen Kasus und einen Fehlerpunkt für die Markierung der Inkongruenz zwischen Verb und Subjekt, da in beiden Fällen ein zusätzliches

err-Feature eingefügt werden muss (Beispiel (5.18)).

(5.17) Fehlerwert 1:

$$\left[\begin{array}{lll} \text{SUBJ} & = & \left[\begin{array}{ll} \text{NUM} & = \text{sg} \\ \text{CASE} & = \text{nom} \\ \text{PRED} & = \text{`GOTT'} \\ \hline \textit{err} & = \left\{ \left[ \text{NUM} \quad = \text{pl} \right] \right\} \end{array}\right] \\ \text{PRED} & = \text{`ZÜRNEN <SUBJ>'} & \end{array}\right]$$

(5.18) Fehlerwert 2:

$$\left[\begin{array}{lll} \text{SUBJ} & = & \left[\begin{array}{ll} \text{NUM} & = \text{pl} \\ \text{CASE} & = \text{gen} \\ \text{PRED} & = \text{`GOTT'} \\ \hline \textit{err} & = \left\{ \left[ \text{CASE} \quad = \text{nom} \right] \right\} \end{array}\right] \\ \text{PRED} & = \text{`ZÜRNEN <SUBJ>'} & \\ \hline \textit{err} & = \left\{ \left[\text{SUBJ} \right] \right\} & \end{array}\right]$$

Auf einen wesentlichen Unterschied zum Parsen mit PS-Regeln sei schließlich hingewiesen.

In einer F-Struktur wird zwar ein Werte-Mismatch und damit eine Fehlererklärung festgehalten, es wird aber insbesondere nicht ermittelt, wie eine Korrektur der Lexeme aussieht, sodass die Korrektur in einem anschließenden Schritt ausgeführt werden muss, zu dem das Verfahren in Abschnitt 5.5 erläutert wird. Im Gegensatz dazu wird beim Parsen mit PS-Regeln eine Korrektur sofort durchgeführt, das heißt, in der vollständigen Beschreibung der Eingabe (der Ableitungsbaum) befindet sich kein Hinweis mehr auf die ursprünglich fehlerhafte Form. Damit trotzdem ein Feedback gegeben werden kann, wird die Kante wie erwähnt so erweitert, dass sie auch Informationen über die Korrektur enthält.

5.3.2 Diskussion

Vier Aspekte sollen in diesem Abschnitt diskutiert werden. Zunächst sollen einige Überlegungen angestellt werden, auf welche Weise unter Umständen zusätzliche Informationen über die Herkunft einzelner sich widersprechender Werte in der F-Struktur kodiert werden können, um so an der vollständigen F-Struktur mehr Informationen für die Präferierung einer Analyse direkt ablesen zu können. Ein weiterer Aspekt ist die Tatsache, dass auch in dem hier präsentierten Szenario zwischen lokalen und globalen Korrekturen unterschieden werden kann und sollte. Im Anschluss daran soll darauf eingegangen werden, welche Auswirkungen zusätzliche Gleichungstypen wie beispielsweise die Negation in Attribut-Wert-Paaren, die in erster Linie zur Reduzierung lexikalischer Einträge genutzt wird, auf die Möglichkeiten der Fehleranalyse hat, und schließlich wird ein Fehlertyp erläutert, der sowohl

durch eine Korrektur der K-Struktur als auch durch einen Markierung in der F-Struktur identifiziert werden könnte.

Präferenzkonstruktion

Da in dem hier vorgestellten Konzept der „sensitiven Unifikation“ im Prinzip mehrere Feature-Strukturen als Ergebnisse einer Unifikation in der Äquivalenzklasse zur Auswahl stehen (siehe Seite 137), stellt sich die Frage, ob und welche Parameter zur Selektion einer der Ergebnisse mit entsprechenden zusätzlichen Interpretationsmöglichkeiten eingesetzt werden könnten. Zu Denken ist hier insbesondere an die zusätzliche Information, die durch das Enthaltensein eines Features direkt in der F-Struktur oder in der Fehlerliste gegeben ist. Eine Möglichkeit besteht so beispielsweise in der Auswahl derjenigen Struktur, in der sich die Features der Kopf-F-Struktur (mit der Annotation ↑=↓) direkt in der F-Struktur und nicht in der Fehlerliste befinden. Damit ließe sich die zusätzliche Information über die Herkunft der unterschiedlichen Werte von unterschiedlichen Lexemen aus der Anordnung der Features und der dazugehörigen Fehlerfeatures für das Feedback ableiten.

Beispielsweise sind dann in solch einem Konzept die Informationen aus der VP eines Satzes direkt in der F-Struktur enthalten, während die Informationen des Subjekts sich in der Fehlerliste befinden, wenn es einen Feature-Clash gegeben hat.

(5.19)
$$\left[\begin{array}{ll} \text{SUBJ} & = \left[\begin{array}{ll} \text{PERS} & = 3 \\ \hline err & = \left\{ \left[\text{PERS} \quad = 1 \right] \right\} \end{array}\right] \\ \text{PRED} & = \text{‘PRED <SUBJ>’} \\ \hline err & = \left\{ \left[\text{SUBJ} \right] \right\} \end{array}\right]$$

In der Feature-Struktur in Beispiel (5.19) ließe sich ablesen, welches Feature, das zum Mismatch beigetragen hat, vom Verb stammt und welches vom Subjekt. Das Feature [PERS = 3] vom Verb ist demnach direkt im komplexen Wert des SUBJ-Attributes enthalten und das Feature [PERS = 1], das vom Subjekt geliefert worden ist, befindet sich im err-Attribut. Eine Untersuchung der Lexeme zur Herkunft der sich widersprechenden Werte entfällt und ein präziseres Feedback kann unmittelbar gegeben werden. Diese Interpretation der Fehlerkodierung setzt allerdings als Konsequenz voraus, dass die Unifikation nicht in beliebiger Reihenfolge erfolgen darf, das heißt, der funktionale Kopf muss dann immer mit dem funktionalen Argument unifiziert werden und nicht umgekehrt.

Allerdings wäre damit die Suche nach der Herkunft der Werte im Ableitungsbaum nicht völlig überflüssig, wenn eine Phrase aus mehr als einem funktionalen Kopf besteht. Wenn sich ein Fehler innerhalb eines funktionalen Arguments ergibt, also zum Beispiel ein Mismatch innnerhalb einer NP

zwischen den Werten eines Artikels und des Substantivs, kann dieses Verfahren nicht angewendet werden. Da in diesem Fall in der F-Struktur keine Möglichkeit der Unterscheidung der Lexeme besteht, muss doch eine Untersuchung der einzelnen Lexeme erfolgen, um eine genaue Rückmeldung mit der Herkunft der sich widersprechenden Werte zu erhalten.

Ein zweiter Aspekt, der möglicherweise gegen die Einschränkung der Unifikation auf diese Weise spricht, ist die Ermittlung der zu korrigierenden Lexeme für eine Rückmeldung. Wenn dem Lerner zusätzlich zur Art des Fehlers auch eine Korrektur geliefert werden soll, dann ist eine genaue Untersuchung der verwendeten Lexeme unerlässlich, da zum Beispiel auf Grund des Synkretismus nicht alle Wortformen korrigiert werden müssen. Wenn aber die Analyse der einzelnen Lexeme unumgänglich ist, kann auf die erläuterte Einschränkung der Unifikation verzichtet werden.

Lokale und globale Korrekturen

Ein zweiter Bereich, der hier Erwähnung finden soll, ist die Unterscheidung von lokalen und globalen Korrekturen. Im Prinzip sind minimale lokale Korrekturen wünschenswert, damit eine minimale globale Korrektur erreicht wird. Die Analyse mit dem geringsten Fehlerwert für die NP „den Mann“ in dem Satz „den Mann lacht“ ist die Analyse als topikalisiertes Objekt, da die Phrase im Akkusativ steht (kein Fehler). Lokal gesehen handelt es sich dabei um eine sinnvolle Hypothese. Wenn nur mit dieser Analyse weitergeparst wird, ergibt sich am Ende allerdings ein unerwünschtes Ergebnis, nämlich dass global gesehen die Analyse unsinnig ist und sie eigentlich Subjekt mit Kasusfehler lauten sollte. Um das zu erreichen, ist es notwendig, nicht nur mit der Kante weiterzuarbeiten, die den geringsten Fehlerwert hat, sondern auch weitere Kanten in Betracht zu ziehen, die zwar lokal schlechtere Hypothesen darstellen, global aber besser geeignet sind, sinnvolle Analysen zu produzieren.

Meines Erachtens kann ein gewisser Schwellwert als Fehlerwert angenommen werden, der die weitere Integration von passiven Kanten zulässt, sodass also nicht nur die Kanten mit dem niedrigsten Fehlerwert berücksichtigt werden, sondern auch alle Kanten, deren Fehler maximal 2 Punkte schlechter ist. Für das Beispiel „den Mann lacht“ bedeutet dies, dass nicht nur die Kante für die NP mit der Interpretation Objekt aufgenommen wird, sondern auch die Kante mit der Interpretation Subjekt (Fehlerwert 2: falscher Kasus des Subjekts).[12] Der Schwellwert von 2 hat sich bei der Untersuchung der Korpussätze als geeignet herausgestellt, was vermutlich mit der Anzahl der bei der Kongruenz und Rektion relevanten Attribute zusammenhängt. Entscheidend für die Korrektheit sind im Deutschen für die Kongruenz von Verb und Subjekt die Attribute Numerus, Person und Kasus. Dass alle drei

[12] Abgesehen wird hier von der Interpretation „NP-interner Fehler“, wobei *den* als Dativ Plural gesehen wird.

Eigenschaften simultan fehlerhaft belegt sind, kommt praktisch nicht vor, sodass in allen Fällen eine Analyse erreicht werden kann.

Zusätzliche Relationen

Zu den üblichen Methoden zur Reduzierung der Komplexität der Informationen gehört in Feature-Strukturen auch die Nutzung weiterer Relationen wie zum Beispiel der Negation oder der aus der LFG bekannten „constraining equation“, die ähnlich wie das Vollständigkeitsprinzip zu grammatischen Funktionen das Vorhandensein eines bestimmten Attribut-Wert-Paares fordert. Damit lässt sich vor allem auch die Anzahl der lexikalischen Einträge reduzieren wie das folgende Beispiel zeigt.

(5.20) Mann, N,

$$\begin{bmatrix} \text{NUM} & = \text{sg} \\ \text{CASE} & = \text{nom} \\ \text{PRED} & = \text{‘MANN’} \end{bmatrix} \vee \begin{bmatrix} \text{NUM} & = \text{sg} \\ \text{CASE} & = \text{dat} \\ \text{PRED} & = \text{‘MANN’} \end{bmatrix} \vee \begin{bmatrix} \text{NUM} & = \text{sg} \\ \text{CASE} & = \text{acc} \\ \text{PRED} & = \text{‘MANN’} \end{bmatrix}$$

(5.21) Mann, N,

$$\begin{bmatrix} \text{NUM} & = \text{sg} \\ \text{CASE} & \neq \text{gen} \\ \text{PRED} & = \text{‘MANN’} \end{bmatrix}$$

Während sich bei der Ausspezifizierung der Kasus-Features drei separate Lexikoneinträge zu *Mann* ergeben, lässt sich die Zahl der Einträge durch die Nutzung der Negation („$\neq$“) auf einen Eintrag reduzieren. Diese Form der Kodierung lexikalischer Informationen hat allerdings deutliche Auswirkungen auf die Möglichkeiten der Fehlermeldungen. Beispielsweise wird für den Satz „ich sehe der Mann“ bei der Nutzung der ausspezifizierten Lexikoneinträge der Fehler mit dem geringsten Fehlerwert als Kasus-Mismatch zwischen dem Artikel *der* und dem Substantiv *Mann* identifiziert, was intuitiv einer sinnvollen Meldung entspricht. Im Gegensatz dazu wird bei der Nutzung der Negationsrelation der Fehler mit dem geringsten Fehlerwert als Kasusfehler zwischen dem Verb und der Objekt-NP identifiziert. Das zweite Ergebnis ist zwar nicht völlig unplausibel, entspricht aber der Intuition nach nicht einer „minimalen“ Fehlerhypothese.

Relevant sind hier die Ausführungen in Menzel (1992), in dem eine vollständige Ausmultiplizierung der Einträge zur präzisen Fehleridentifikation genutzt wird, die aber auch zur erwähnten Komplexität des Ansatzes führt. Im Gegensatz dazu wird im vorliegenden Ansatz die Strategie verfolgt, möglichst wenig Einträge für jede homographische Form im Lexikon zu haben,

um so den Parsingprozess zu beschleunigen, auch wenn auf diese Weise weniger präzise Fehlermeldungen möglich sind, da wie in der Evaluation (Seite 148) gezeigt, insbesondere diese Synkretismen zu Performanzeinbußen führen.

Interaktion von F- und K-Struktur

Ein spezielles Problem der Interaktion zwischen K- und F-Struktur entsteht bei der Analyse eines Satzes, der einen Artikelfehler enthält, wie zum Beispiel

(5.22) *Das Auto ist gegen Ampel gefahren.

Falls sich aus der Analyse ergibt, dass Wertebelegungen wie [NTYPE = norm] und [NTYPE = mass] nicht miteinander korrespondieren, kann dem Lerner zunächst nur mitgeteilt werden, dass es sich bei dem Substantiv *Ampel* um ein normales und nicht um ein Stoffsubstantiv handelt. Die erwartete Rückmeldung, dass es sich bei *Ampel* um ein Substantiv handelt, das mit einem Artikel verwendet werden muss, kann entweder speziell mit dieser Wertebelegung verknüpft werden, was aber zunächst unplausibel erscheint. In einer zweiten Möglickeit aber kann das Feature NTYPE zu einem „harten“ Feature erklärt werden, bei dem ein Fehler nicht auftreten darf. Im diesem Fall muss dann der Fehler mit Hilfe des Mechanismus für Konstituentenfehler erkannt werden.[13] Die Lösung besteht in Promis$D$ in eben der Unterdrückung einer Fehlermöglichkeit auf der Basis des NTYPE-Attributs. Dem Lerner wird daher mitgeteilt, dass ein Artikel fehlt und nicht, dass das artikellose Substantiv ein „normales“ Substantiv und nicht ein Stoffsubstantiv ist. Eine Erläuterung für den Lerner, dass nur Stoffsubstantive und Ähnliches ohne Artikel verwendet werden können, kann also nicht aus der Grammatik extrahiert werden, auch wenn die Präsentation einer entsprechenden F-Struktur verständliche Informationen für diesen Bereich bietet. Allgemeiner gesehen besteht also in manchen Fällen die Möglichkeit, zwischen der Identifizierung eines Fehlers in der F-Struktur oder in der K-Struktur abzuwägen.

Zusammenfassend lässt sich festhalten, dass erstens die Verteilung der Features im Fall von sich widersprechenden Werten ausgenutzt werden kann, um daraus ein wenig mehr Informationen für die Diagnose eines Fehlers und die Rückmeldung zu extrahieren. Allerdings besteht diese Möglichkeit nur für eine begrenzte Anzahl von Fehlern, sodass eine Realisierung in der Implementation nicht vorgenommen wurde. Zweitens muss der Einsatz von lokalen Korrekturen so gestaltet werden, dass einerseits global die Analysen mit den geringsten Fehlerwerten für das Feedback berücksichtigt werden

[13] Es scheint mir nicht sinnvoll, grundsätzlich andere Formen der Modellierung wie zum Beispiel eine Art Valenzforderung für einen Artikel zu wählen, um lediglich der relativen Häufigkeit dieses Fehlertyps gerecht zu werden.

können und andererseits nicht beliebige Lösungen mit einer beliebigen Anzahl an Fehlern aus dem Parse-Vorgang hervorgehen und anschließend wurde gezeigt, in welcher Art und Weise die Nutzung von zusätzlichen Relationen die Möglickeiten der Fehleridentifizierung beeinflusst. Schließlich wurde ein Beispiel gezeigt, in dem eine gewisse Interaktion zwischen der F-Struktur und der K-Struktur im Rahmen der Fehlererkennung gegeben ist und der „Ausschluss“ eines Features von der Fehlererkennung die präferierte Fehleranalyse auf der Basis der K-Struktur produziert.

Im ersten Teil dieses Unterkapitels wurde eine Methode zur Erhaltung von sich widersprechenden Informationen in Feature-Strukturen vorgestellt, die sich auf die Integration eines zusätzlichen Features stützt. Notwendig ist dazu im Wesentlichen nur die Umformulierung der Definition der Subsumption mit Hilfe einer Substitutionsfunktion, wobei die Integration auf atomare Werte von Attributen beschränkt wurde. Damit lassen sich einerseits linguistisch motivierte F-Strukturen handhaben, die nur wenig Abweichung voneinander zeigen und andererseits sind konkrete Maßnahmen zur Limitierung der Ergebnisse anwendbar, wie zum Beispiel die Einschränkung auf einen bestimmten Satz an Features, bei denen überhaupt ein Fehler auftreten darf. Im folgenden Unterkapitel soll nun eine quantitative und qualitative Bewertung der Maßnahmen zur Fehleridentifizierung erfolgen und insbesondere anhand eines kleinen Korpus von 150 fehlerhaften Sätzen gezeigt werden, welche Auswirkungen sich für den Parsing-Prozess und für die Möglichkeiten zur Rückmeldung eines Fehlers an den Lerner ergeben.

5.4 Praktische Ergebnisse: Effizienz und Evaluation

In diesem Abschnitt werden zwei Bereiche angesprochen, die eng miteinander verbunden sind: Zu Beginn geht es allgemein um die Effizienz dieses Verfahrens und um die Abwägungen, die zwischen Effizienz und Fehlerabdeckungsbreite getroffen werden können. Im Anschluss daran wird gezeigt, inwieweit Fehler in Sätzen eines Korpus erkannt werden können.

5.4.1 Effizienz

Es ist klar, dass ein Verfahren, bei dem eine Form der Relaxierung angewandt wird, deutlich an Effizienz einbüßt. In Langer (2001) sind dazu beispielsweise Experimente mit einer unterschiedlichen Anzahl von relaxierten Attributen unternommen worden, die deutlich zeigen, wie beim Parsing die Anzahl der analysierten Wörter pro Sekunde sinkt und die Ambiguität steigt, je mehr Features relaxiert werden. Schließlich sinkt auch die Anzahl der tatsächlich analysierten Sätze, die innerhalb der festgelegten Maximalwerte erkannt werden, von 11.025 auf 8.109 Sätze des Korpus. Die Performanzeinbußen zeigen

sich in der folgenden Tabelle.

relaxierte Attribute	Wörter/Sek.
-	3.181143
NUM CASE PREPOSITION PREPTYPE GENDER	0.81619

Tabelle 5.2: Performanzeinbußen durch Merkmalrelaxierung (Langer, 2001, S. 116)

Die Relaxierung von fünf Attributen verlangsamt also den Parser auf ca. 25% der ursprünglichen Leistung. Ähnliche Ergebnisse werden auch in Vandeventer (2001, S. 118) berichtet, in denen die Relaxierung von drei Attributen den Parser um 23,2% verlangsamt. Selbstverständlich zeigen sich auch bei dem hier implementierten Parser diese Effekte, die durch die relevanten Werte für den Parser aus Promis*D* belegt werden können. Um realistische Bedingungen zu simulieren, wurden zusätzlich zur Chartmanipulation deutlich mehr Features für eine Relaxierung freigegeben, als es bei den erwähnten Experimenten in der Literatur der Fall ist. An den folgenden 12 Features konnten bei den Durchläufen Fehler auftreten (Wiederholung von Tabelle 5.1 auf Seite 140).

CASE	Kasus	MOOD	Finitheit
NUM	Numerus	MODUS	Modus
PERS	Person	COMPTYPE	Verbkomplementtyp
GEN	Genus	MOOD2	Finitheit Komplement
AUXTYPE	Auxiliartyp	PCASE	gram. Funktion einer
SPECTYPE	Artikeltyp für		subkat. PP
	Adjektivflexion		(LFG-spezifisch)
		MODE	Präpositionalkasus

Tabelle 5.3: Liste der zulässigen Fehler-Features

In diesen Features sind Informationen enthalten, die einerseits für einen Lerner relevant sind und andererseits die Fehlertypen in dem 150 Sätze großen Korpus abdecken, wie im folgenden Abschnitt gezeigt wird. Von den korrigierten Sätzen können 9 (6%) mit eingeschalteter Fehlererkennung nicht erkannt werden, wenn ein Maximalwert von 20.000 aktiven Kanten in der Chart festgelegt wird.

In Prozentpunkten ausgedrückt bedeutet dieses, dass der Parser nur ca. 1,7% der ursprünglichen Leistung erreicht. Auch die Anzahl der Lösungen pro Satz steigt offensichtlich beträchtlich, wenn die Fehlererkennung eingesetzt wird. Für die korrigierten Sätze ergeben sich bei ausgeschalteter Feh-

relaxierte Attribute + Chartmodifikation	Wörter/Sek.
-	2,52
12 relaxierte Attribute (Tabelle 5.3) + Chartmodifikation (Abbildung 5.1, S. 127)	0,043

Tabelle 5.4: Performanzeinbußen durch Merkmalrelaxierung und Chartmodifikation in Promis$D$

lererkennung durchschnittlich 1,8 Ergebnisse pro Satz, während mit angeschalteter Fehlererkennung durchschnittlich 5,8 Ergebnisse pro Satz generiert werden, wobei sich für diesen Fall auch Lösungen ergeben, die „fehlerhafte“ Analysen darstellen.

Die folgende Abbildung 5.4 zeigt die Effizienz des Parsers mit Hilfe der generierten Anzahl von passiven Kanten nach Sätzen einzelner Wortlängen in logarithmischer Darstellung. Dazu wurden alle korrigierten Versionen der Sätze der im nächsten Abschnitt beschriebenen Evaluationssatzmenge geparst. Wie aus der Abbildung 5.4 leicht abzulesen ist, werden für die Sätze, für die tatsächlich ein Ergebnis erzielt wird, mit eingeschalteter Fehlererkennung deutlich mehr passive Kanten in der Chart (max. 3874) generiert als bei ausgeschalteter Fehlererkennung (max. 551).

Hervorgehoben werden kann schließlich, dass für die Werte mit eingeschalteter Fehlererkennung eine wesentlich größere Standardabweichung in Bezug auf die Zahl der generierten passiven Kanten besteht.

$n = 27$	Fehlererkennung an	Fehlererkennung aus
Mittelwert	361,74	138,67
Varianz	59516	2660,89
Standardabweichung	243,96	51,58

Tabelle 5.5: Mittelwert, Varianz und Standardabweichung zur Anzahl der passiven Kanten für Satzlänge = 8 (korrigierte Sätze)

Eine nähere Untersuchung der Sätze, bei denen der Parser besonders lange gebraucht hat, zeigt, dass das vor allem auf den so genannten Synkretismus zurückzuführen ist. Da in der Implementation keine Disjunktion sondern nur die Negation zur Modellierung der morphosyntaktischen Eigenschaften verwendet werden kann, existieren zum Beispiel für Adjektive bis zu sechs Einträge im Lexikon für identische Wortformen. Fast alle Features, in denen sich diese Einträge unterscheiden, gehören zu den relaxierbaren Features. Daraus ergibt sich, dass bei einer Eingabe mit Adjektiv alle sechs Formen zur Konstruktion von umfassenderen Kanten mit sehr geringen Unterschieden in die Chart eingetragen werden, obwohl der lokale Kontext eine Disambiguierung erlauben würde. Ein zweiter Bereich sind Präpositional-

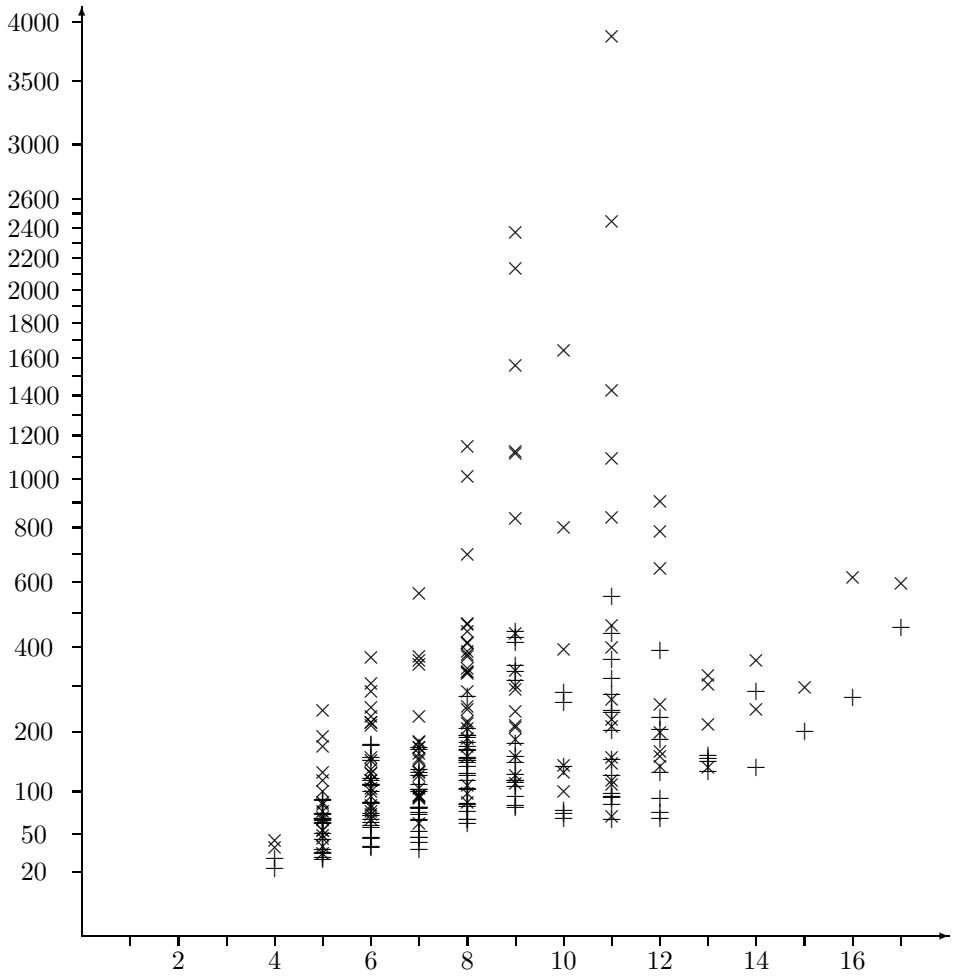


Abbildung 5.4: Anzahl der Kanten/Satzlänge in Anzahl der Wörter; + = Relaxierung aus- und Look-ahead angeschaltet; × = Relaxierung an- und Look-ahead ausgeschaltet

phrasen, bei denen zusätzlich zu unterschiedlicher Kasusrektion auch die Variabilität der Anbindung bekanntermaßen viele Kanten in der Chart erzeugt.

Die Zahlen aus Abbildung 5.4 können natürlich nur einen Anhaltspunkt darstellen. Verschiedene andere Faktoren wie Programmiersprache, Kompilierung etc. beeinflussen die tatsächliche Effizienz erheblich. Zum Beispiel wird die Grammatik des Gepard-Parsers (Langer, 2001) vor dem Parsing kompiliert, sodass die Featurenamen in Integer-Werte umgewandelt werden. Kuhn und Rohrer (1997) berichten dagegen von Parsingergebnissen, bei denen für einen bestimmten Satz die Geschwindigkeit fast um den Faktor 50 schneller wird, wenn die PS-Regel, die eine NP nur mit einem Artikel zulässt, entfernt wird. Auch für den vorliegenden Ansatz kann ein ähnliches

Beispiel konstruiert werden. Die Analyse des korrekten Satzes „Ich sehe den schrecklichen Mann.“ mit einer Relaxierung der üblichen Features CASE, NUM, GEN etc. einschließlich des Attributs SPECTYPE, das die Form des Adjektivs je nach Artikeltyp steuert, dauert fast doppelt so lange wie ohne Relaxierung des einen Attributs SPECTYPE und es werden ca. 1,5-mal so viele Kanten in die Chart aufgenommen. Die Relaxierung nur allein des SPECTYPE-Attributes macht dagegen keinen Unterschied, da die anderen Kongruenz-Features eine Fehler-Unifikation der F-Strukturen weitestgehend verhindern. Dieses Beispiel zeigt, in welchem Ausmaße auch scheinbar nebensächliche Attribute den Suchraum verändern können und welchen erheblichen Einfluss somit die Struktur der Grammatik hat. Für ein ICALL-System allerdings, das möglichst umfassend über aufgetretene Fehler Rückmeldung geben soll, eignen sich Einschränkungen im linguistischen Bereich nicht, da in jedem Fall das Spektrum der identifizierbaren Fehler mit unerwünschten Folgen eingeschränkt würde.

5.4.2 Evaluation

In den vorhergehenden Abschnitten wurde ein Verfahren vorgestellt, das in der Lage sein soll, einen größeren Teil der Fehler in Lernereingaben in einem ICALL-System zu erkennen. Um das Verfahren effizienter zu machen, wurden verschiedene Möglichkeiten auf der Basis von Annahmen über die tatsächlich auftretenden Fehler ausgenutzt. Zum Beispiel werden die Features, bei deren Unifikation ein Mismatch auftreten darf, auf bestimmte Typen begrenzt, und der Schwerpunkt der Erkennung von Konstituentenfehlern liegt auf Auslassungs-, Verschiebungs- und Einfügungsfehlern. In diesem Abschnitt wird nun eine genauere Analyse der schon in Unterkapitel 3.4.1 erwähnten Korpora vorgestellt, die diese Annahmen untermauern. Im zweiten Teil wird ausgeführt, welche Fehlertypen in einem kleinen Testsatzkorpus von 150 Sätzen vom Parser erkannt und welche nicht erkannt wurden. In der Literatur gibt es meines Wissens nach nur sehr wenige Untersuchungen, in denen Lernerkorpora zur Evaluation von sensitiven Parsingverfahren eingesetzt wurden (zum Beispiel Granger et al., 2001). In Vandeventer (2001) wird ein umfassendes Parsingsystem, das eigentlich zur Analyse von korrekter Sprache verwendet wird, so modifiziert, dass einzelne Constraints relaxiert werden können. Dabei hat sich erstens herausgestellt, dass, wie im vorhergehenden Abschnitt ausgeführt, schon die Relaxierung von nur drei Constraints einen erheblichen Performanzverlust zur Folge hatte. Zweitens generierte der Parser mit eingeschalteter Relaxierung eine Reihe von „false-positive“ Ergebnissen, worin sich das schon erwähnte Problem (Seite 114) eines antizipationsfreien Parsers zeigt, nicht zwischen einem fehlerhaften und einem nicht von der Grammatik abgedeckten Satz unterscheiden zu können. Ein wichtiger Aspekt muss also bei der Entwicklung eines tatsächlich umfassenden sensitiven Parsers die Vermeidung solcher „False-Positives“ sein,

obwohl dieser Bereich hier nicht weiter berücksichtigt werden soll, da es sich bei der Grammatik und dem Lexikon in Promis*D* nicht um „umfassende“ Modellierungen handelt. Zu kritisieren ist an den Ergebnissen allerdings, dass der Parser tatsächlich nur Fehler in Bezug auf lediglich drei Features erkennen konnte, was in keiner Weise den von Lernern produzierten Fehlertypen entspricht, wie sich zeigen wird. Die folgende Tabelle, die auch auf Seite 72 zu finden ist, gibt noch einmal einen Überblick über die hier verwendeten Korpora.

	HU/FU-Korpus	Heringer-Korpus
fehlerhafte Sätze	103	7107
Wörter	683	82013
Wörter/Satz	6,6	11,5
Manipulation	keine	1 Fehler/Satz
Produktion	Promis*D*-Nutzer 2000	? (70er Jahre)
Herkunft/ Meta-Information	ausländische Studierende der HU und FU Berlin	unbekannte Lerner am Goethe-Institut München
Stufe	Mittelstufe	ca. Mittelstufe
Besonderheiten	z.T. nur Phrasen	vollständige Sätze, keine weiteren Meta-Informationen
Sätze in der Evaluation	75	75

Tabelle 5.6: Übersicht über die Korpora

Das Heringer-Korpus

Das Heringer-Korpus (Heringer, 1995) wurde aufgebaut, um Lernern in einem Sprachlernprogramm mit Hilfe von fehlerhaften Sätzen die Vermeidung von häufigen Fehlern zu vermitteln. Um den Nutzern des Korpus den Überblick zu erleichtern, wurde das Korpus insofern manipuliert, als dass jeder Satz nur einen Fehler enthält. Gegebenenfalls wurde daher derselbe Satz mehrfach mit jeweils einem Fehler in das Korpus aufgenommen. Damit geht zwar ein Teil der Authentizität und die Komplexität möglicher Fehlerkombinationen verloren, aber das Spektrum der morphosyntaktischen Fehler bleibt erhalten. Das folgende Beispiel kann einen Eindruck vermitteln.

(5.23) Aber bis mein Praktikum beginnt $+_+$ habe ich 3 Monate Zeit.
Aber bis mein Praktikum beginnt, habe ich 3 Monate$+n+$ Zeit.
Aber bis mein $+p+$raktikum beginnt, habe ich 3 Monate Zeit.

Hier kann die Vermutung gelten, dass der Satz ursprünglich drei Fehler enthielt, und dann als drei separate Sätze mit jeweils einem Interpunktionsfehler, einem morphosyntaktischen Fehler und einem orthografischen Fehler

in die Datenbank aufgenommen wurden. Da in Promis*D* orthografische Fehler vor dem Parsing korrigiert werden müssen und Kommata in der Grammatik grundsätzlich als optional modelliert sind, wurde also nur der zweite Satz in die Evaluation mit aufgenommen.

Alle fehlerhaften Sätze sind mit vier Annotationen versehen:

- einer Markierung der Fehlerstelle (Heute $+arbeitet+$ mehr Frauen als früher.),
- einem Fehler-Code, der Aufschluss über die Kategorie und den genauen Typ des Fehlers gibt (˜KONGR ˜V ˜N ˜NUM ˜),
- einer korrigierten Version des Satzes (Heute arbeiten mehr Frauen als früher.) und
- einer Umschreibung des Fehlers (Das ist ein Fehler in der Kongruenz. Das finite Verb und das Subjekt müssen im Numerus übereinstimmen, auch bei nachgestelltem Subjekt.)

Die folgenden beiden Tabellen 5.7 und 5.8 können einen Eindruck davon vermitteln, wie die Verteilung der häufigen Fehler und Fehlertypen im Korpus ist. Folgende groben Fehlerklassen finden sich bezogen auf die Gesamtfehlerzahl am häufigsten im Korpus, wobei einige Beispiele zu einzelnen Klassen in Tabelle 5.8 aufgeführt sind.

Fehlertyp	Fehlercode	n	%
Ellipse	ELL	1170	16
Topikalisierung	TOP	970	14
Tempusabfolge	V TEMP	796	11
Morphologie	MORPH	702	10
Kongruenz	KONGR	685	10
Valenz	V VAL	581	8
Einfügung	PLUS	391	6
Orthografie	ORTH	292	4
Zeichensetzung	ZS	290	4
Wortbildung	WOBI	139	2
Sonstige			15

Tabelle 5.7: Fehlerklassen und Verteilung im Heringer-Korpus

- Auslassungsfehler (ELL) machen die weitaus häufigste Fehlerklasse aus. Bei weitem am häufigsten wurden Subjekte ausgelassen und mit Abstand folgen Präpositionen.

- Die Fehlerklasse der Topikalisierungsfehler (TOP) umfasst fast alle Arten von Verschiebungsfehlern, das heißt, es sind zwar alle notwendigen Elemente im Satz vorhanden, sie befinden sich aber nicht an der korrekten Position.

- Die folgenden zwei Fehlerklassen Tempusabfolge (V TEMP) und Morphologie (MORPH) werden von dem hier vorgestellten Verfahren nur zum Teil abgedeckt. Unter dem Begriff Tempusabfolge werden vor allem die Tempusfehler zwischen zwei koordinierten Sätzen oder einem Matrix- und einem subordinierten Nebensatz zusammengefasst. In dieser Klasse sind viele Fehler semantischer Art und können nur in Einzelfällen syntaktisch begründet werden. Morphologiefehler sind zum großen Teil regelhafte, aber falsch geformte Lexeme der Kategorien Adjektiv, Verb und Artikel. Wenn sich aus den Fehlformen ein existierendes Wort ergibt, kann der vorliegende Parser unter Umständen den Fehler erkennen (siehe auch Tabelle 5.8).

- Kongruenzfehler (KONGR) treten vor allem zwischen Subjekt und Verb sowie innerhalb von komplexen NPn auf, die zum Teil auch Adjektive enthalten, sodass im Prinzip alle Fehler dieses Typs in Promis$D$ erkannt werden sollten.

- Auch Fehler aus der Klasse Valenz (VAL) kann der Parser wie erläutert erkennen. Hier besteht offensichtlich die Voraussetzung, dass das Lexikon die entsprechenden Daten zur Verfügung stellen muss. Nur wenn zum Beispiel im Lexikon enthalten ist, dass ein Verb eine bestimmte Präposition fordert und es außerdem keinen Eintrag im Lexikon gibt, der eine identische Subkat-Liste ohne die Präposition enthält, kann ein Fehler „fehlende Präposition“ identifiziert werden. Unter die Klasse der Valenzfehler fallen zum Teil aber auch „einfache“ Fehler wie zum Beispiel der falsche Kasus eines Objekts, die wie die Kongruenzfehler mit dem Subjekt ohne Schwierigkeiten erkannt werden können.

- Bei der kleineren Klasse der Einfügungen (PLUS) handelt es sich zum Teil um zusammengezogene Präpositionen mit zusätzlichem Artikel, was auch in vielen Fällen vom Parser identifiziert werden kann.

- Die letzten drei Klassen Orthografie (ORTH), Zeichensetzung (ZS) und Wortbildung (WOBI), gehören ebenfalls zu den nicht korrigierbaren Kategorien. Allerdings liefert das System für Wörter, die nicht im Lexikon enthalten sind, Vorschläge, wie das Wort korrigiert werden könnte (siehe Abschnitt 6.2 zur Orthografie).

Tabelle 5.8 auf Seite 154 beinhaltet die 15 häufigsten Fehlercodes und Beispiele des Heringer-Korpus. Die morphosyntaktischen Kategorien, die unter anderem für die Evaluation berücksichtigt wurden beziehungsweise die vom Parser erkannt werden müssten, sind in der Tabelle auf der folgenden Seite mit einem + markiert.

In Bezug auf die Beispielsätze ist anzumerken, dass es sich bei einigen Kategorien um Fehler handelt, die nicht notwendigerweise morphosyntaktischer

Code	n	Erläuterung	Beispiel
˜KONGR˜V˜N˜NUM˜	265	+ Numerus-Inkongruenz	Heute *arbeitet* mehr Frauen als früher.
˜ZS˜KOM˜	264	Zeichensetzung	Sie hofften _ einen netten Mieter zu bekommen.
˜PRÄ˜KAS˜DAT˜	238	+ Präposition fordert Dativ	Die Chancen ergeben sich aus *die* Erkenntnis, daß ...
˜PRÄ˜	231	(+) falsche Präposition	Wir gingen schwimmen und lagen *auf* der Sonne.
˜TOP˜VST˜e˜	179	+ Verbstellung falsch	Er ging gleich ins Bett, weil er *war müde*.
˜V˜VAL˜PR˜KAS˜AKK˜	167	+ Valenz fordert Akkusativ	Er beschrieb mir *über* den Blick auf die Berge.
˜V˜VAL˜PRÄ˜	162	+ Valenz/Kasus der Präp falsch	Der See grenzt *von* dieses Land.
˜V˜TEMP˜PRÄT˜6a˜	153	Tempus (mit NS); sollte Prät sein	Es gab viele Leute, die schwarze Kleider *tragen*.
˜ELL˜V˜VAL˜PR˜KAS˜N˜	141	+ Subjekt fehlt	Plötzlich geht _ zum Verkäufer.
˜MORPH˜V˜TEMP˜PRÄT˜	131	Präteritumsform falsch geformt	Er *nehmt* ein Streichholz und sah nach.
˜V˜TEMP˜PERF˜HASEIN˜	117	+ hat/sein falsch oder fehlt	Ein sehr angenehmer Tag *habe* vergangen.
˜MORPH˜V˜TEMP˜PRÄS˜	114	Präsensform falsch geformt	Ihr wißt ja, daß ich eine gute Kamera kaufen *möchtete*.
˜V˜VAL˜PR˜KAS˜DAT˜	113	+ Valenz fordert Dativ	Sie folgte *den* Weg vom Vater zum Sohn.
˜TOP˜VST˜p˜	108	+ fin Verb muss von Pos. 3 nach 2	Meiner Meinung nach *es genügt* für die Kinder, ...
˜TOP˜VST˜z˜	107	+ fin Verb muss von 1 oder 3 nach 2	Außerdem *da ein Kurort liegt*.

Tabelle 5.8: Die häufigsten Fehlercodes im Heringer-Korpus

Natur sind, obwohl sie diese Markierung erhalten haben. Der Beispielsatz „Wir gingen schwimmen und lagen *auf* der Sonne.“ für die Fehlerkategorie ˜PRÄ˜ beispielsweise ist morphosyntaktisch nicht inkorrekt, da das Präpositionalobjekt vermutlich nicht obligatorisch ist. Ähnliches gilt auch für die Kategorie ˜V˜TEMP˜PRÄT˜6a˜, in der viele Sätze wie erwähnt durchaus akzeptabel erscheinen.

Eine letzte Eigenschaft des Heringer-Korpus, die hier vorgestellt werden soll, ist die Gesamtverteilung der 403 Fehlertypen über das Korpus. Erwähnenswert ist, dass bereits die achtzig häufigsten Fehlertypen (20%) ca. 79% der Sätze abdecken. Selbst die zehn häufigsten Fehlertypen aus der Tabelle 5.8 decken immerhin noch ca. 20% der Sätze ab. Im Gegensatz dazu treten ca. 50% der Fehlertypen nur ein-, zwei- oder dreimal auf. Es zeigt sich also, dass wenige Fehlertypen sehr häufig auftreten, während relativ viele Fehlertypen sehr selten auftreten.

Aus dem Heringer-Korpus wurden für die Evaluation 75 Sätze zufällig ausgewählt, wobei die einzige Einschränkung darin bestand, dass es sich bei dem Fehlertyp um einen morphosyntaktischen Fehler handeln musste (siehe die Tabelle 5.8 und die Markierung mit dem „+“ als Beispiel).[14] Für die Evaluation wurde die bestehende Annotation des Fehlertyps als Grundlage gewählt.

Das HU/FU-Korpus

Dieses Korpus wurde wie in Abschnitt 3.4.1 erläutert von ausländischen Studierenden in Mittelstufenkursen „Deutsch als Fremdsprache“ der HU und der FU Berlin bei der Benutzung von Promis*D* und in Simulationen des Programms produziert. Das Evaluationskorpus umfasst fast alle vollständigen und morphosyntaktisch fehlerhaften Sätze des HU/FU-Korpus. Einige Sätze wurden unter anderem aus folgenden Gründen weggelassen:

- Ein Satz enthält die Formulierung „ich anrufe aus Automatentelefon ...“, zu der der Parser keine Lösung finden kann, da kein Morphologiemodul enthalten ist, das das Kompositum *Automatentelefon* analysieren könnte.

- Einige weitere Sätze wurden weggelassen, da sie Relativsätze mit den Relativpronomen *wo* und *was* enthalten. Da die Verwendung dieser Relativpronomen auf sehr begrenzte Kontexte beschränkt ist, wurde dieser Typ in der Grammatik nicht modelliert.

- Die meisten Sätze enthielten verschiedene Namen mit ansonsten identischen Formulierungen, zum Beispiel „ich wohne in/auf ...“. Da dieselben Formulierungen schon mit einem anderen Namen oder in sehr

[14] Dies bedeutet auch, dass nicht alle morphosyntaktischen Fehlertypen in der Evaluation vertreten sind.

ähnlicher Form im Korpus enthalten sind, wurden diese Sätze weggelassen und nur eine Version in der Evaluation verwendet.

Damit ergibt sich aus diesem Bereich ein Korpus für die Evaluation von 75 Sätzen, die zum Teil mehrfach Fehler enthalten und die manuell für die Evaluation bestimmt wurden. Schließlich sei hier noch einmal darauf hingewiesen, dass sich in den Sätzen keine orthografischen Fehler befinden, da diese schon bei der Benutzung von Promis*D* beziehungsweise von mir korrigiert worden sind.

Evaluationsmethode

Bei der Frage einer geeigneten Evaluation orientiert sich der vorliegende Ansatz an Foster (2004). Zur Evaluation wurden die Grammatik und das Lexikon so erweitert, dass die korrigierten Versionen der Sätze im Evaluationskorpus abgedeckt wurden. Das heißt, dass der Parser mit abgeschalteter Fehlererkennung für jeden korrigierten Satz mindestens eine Analyse generieren konnte. Daraus folgt, dass in dieser Untersuchung keine so genannten False-Positives auftreten konnten. Im Anschluss wurden die fehlerhaften Sätze geparst.

Der Fokus liegt in dieser Evaluation insbesondere im Gegensatz zur Evaluation von „herkömmlichen“ Parsern nicht darauf, welchen Abdeckungsgrad die Grammatik und das Lexikon besitzen, sondern auf der Frage, welche belegten Fehler und Fehlertypen identifiziert werden können. Desweiteren besteht im Allgemeinen die Möglichkeit, zur Evaluation die Strukturen der Analysen miteinander zu vergleichen und daraus eine Form von Precision und Recall zu errechnen. Hier soll dagegen ein Vergleich der Parse-Resultate mit den von Hand ermittelten Fehlern durchgeführt werden, um auf diese Weise in Hinblick auf die Generierung von Fehlermeldungen ein Maß zu erhalten. Tabelle 5.9 enthält eine Übersicht über die manuell identifizierten Fehlertypen in den Evaluationssätzen analog zur Unterscheidung in Fehler in der K-Struktur (Konstituentenfehler) und Fehler in der F-Struktur (Kongruenz- und Rektionsfehler, POS-Fehler, Verbformfehler, „haben–sein–werden“-Fehler).

Bemerkenswert ist, dass im HU/FU-Korpus sehr häufig der Artikel insbesondere mit Ortsbezeichnungen sowie Straßennamen ausgelassen wurde, was sich vielleicht sowohl durch die Aufgabenstellung als auch die Muttersprache der Lerner erklären lässt, da ja beispielsweise im Russischen Definitheit nicht durch einen vorangestellten Artikel ausgedrückt wird. Im vollständigen Heringer-Korpus dagegen macht die Zahl der fehlenden Artikel (182) nur die Hälfte der fehlenden Subjekte und Objekte (310) aus. Außerdem lässt sich in der Tabelle erkennen, dass fast jeder Fehlertyp des HU/FU-Korpus auch in einem Satz des Heringer-Korpus auftaucht. Damit stellen die Sätze eine geeignete Ergänzung zum mit Promis*D* erzeugten Korpus dar. Die Gruppe

	HU/FU-Korpus	Heringer-Sätze	Gesamt
Konstituentenfehler			
Einfügung	3	5	8
Auslassung Artikel	19	–	19
Subj./Obj.	3	7	10
Sonst.	11	–	11
Verschiebung Subj.	3	5	8
Verb	5	9	14
Sonst.	1	2	3
Kongruenz-/ Rektionsfehler			
Subjekt	9	9	18
Objekt	18	9	27
Präp.-Objekt	13	12	25
POS-Fehler	3	–	3
Verbform	3	10	13
haben–sein–werden	2	8	10
	93	76	169

Tabelle 5.9: Manuell identifizierte Fehlertypen in den Evaluationssätzen

der Kongruenz- und Rektionsfehler umfasst nicht nur Fehler, die sich beispielsweise zwischen Subjekt und Verb ergeben, sondern auch Kongruenzfehler innerhalb einer NP, also zum Beispiel zwischen Artikel und Adjektiv. Verbformfehler bestehen beispielsweise aus einem falsch gewählten Infinitiv, obwohl eigentlich ein Partizip erforderlich wäre. Schließlich muss angemerkt werden, dass zwar eigentlich im Heringer-Korpus nur jeder Satz einen Fehler enthalten sollte, allerdings ein Satz ein falsches Auxiliar enthielt, das zudem nicht mit dem Subjekt kongruierte. Hier mussten also zwei Fehler angenommen werden, sodass sich eine Gesamtfehlerzahl von 76 ergibt, während 15 der HU/FU-Sätze zwei oder mehr Fehler enthielten.

Auswertung

Die folgende Tabelle 5.10 bietet einen Überblick über die mit Hilfe des Parsers erzielten Ergebnisse auf der Basis von Tabelle 5.9.

Ein identifizierter Fehler wurde jeweils einer von vier Kategorien (in den Korpus-Spalten von Tabelle 5.10) zugeordnet, was eine realtiv genaue Analyse ermöglicht: 1. der Fehler wurde korrekt und als präferierter Fehler erkannt, 2. der Fehler wurde unter anderem mit gleicher Präferenz zu anderen Fehlerhypothesen erkannt, 3. es gab zwar ein Analyseergebnis, aber der Fehler wurde nicht erkannt und 4. der Satz konnte nicht analysiert werden, das heißt, es wurde keine den ganzen Satz überspannende Kante generiert, was zumeist am festgelegten Maximalwert für die Anzahl der Kanten (20.000) lag. Von den Einfügungsfehlern im HU/FU-Korpus wurde also beispielsweise ein Fehler richtig erkannt und für zwei Fehler wurde keine/n Analyse/n

	HU/FU-Korpus	Heringer-Sätze	Gesamt
Konstituentenfehler			
Einfügung	1 / – / – / 2	2 / 1 / – / 2	8
Auslassung Artikel	13 / 2 / 2 / 2	–	19
Subj./Obj.	2 / – / – / 1	4 / – / 3 / –	10
Sonst.	3 / 1 / 4 / 3	–	11
Verschiebung Subj.	– / – / 1 / 2	2 / 2 / – / 1	8
Verb	– / 2 / 2 / 1	3 / – / 2 / 4	14
Sonst.	– / – / – / 1	– / – / 2 / –	3
Kongruenz-/ Rektionsfehler			
Subjekt	4 / 1 / 4 / –	7 / 2 / – / –	18
Objekt	10 / 6 / 2 / –	7 / – / 1 / 1	27
Präp.-Objekt	10 / – / 1 / 2	10 / 1 / – / 1	25
POS-Fehler	– / – / 2 / 1	–	3
Verbform	2 / – / 1 / –	9 / – / 1 / –	13
haben–sein–werden	2 / – / – / –	7 / – / – / 1	10
absolut	47 / 12 / 19 / 15	51 / 6 / 9 / 10	169
in %	51 / 13/ 20 / 16	67 / 8 / 12 / 13	100

Tabelle 5.10: Evaluation nach Fehlertypen in den Evaluationssätzen (korrekt / u.a. / falsch / nicht)

erreicht. Aus der Menge der Heringer-Sätze wurde beispielsweise ein Einfügungsfehler zweimal korrekt als präferierter Fehler identifiziert, einmal zusammen mit weiteren Hypothesen und zweimal schlug die Analyse fehl.

Für das HU/FU-Korpus ergibt sich also eine Zahl von 51% korrekt identifizierten Fehlern, während die Fehler im Heringer-Korpus sogar zu 67% erkannt worden sind. Die Fälle der u.a. korrekt identifizierten Fehler wurden in der Tabelle 5.10 unter „u.a." zusammengefasst und treffen nur für ca. 15% der Fehler des HU/FU-Korpus zu. Zum Beispiel wurde für den oben angeführten Satz „Heute arbeitet mehr Frauen als früher" sowohl ein Numerus- als auch ein Person-Fehler mit gleichem Fehlerwert ermittelt. An dieser Stelle sei betont, dass bei dieser Evaluation für den Bereich der F-Struktur lediglich die identifizierten Fehlertypen gewertet wurden. Der Versuch einer Korrektur für den Satz „Heute arbeitet mehr Frauen als früher" sollte beispielsweise zeigen, dass nur die Korrektur des Verbs mit einer Präferenz des Numerus-Fehlers für die Rückmeldung Sinn macht.

Wenn ein Fehler „falsch" identifiziert wurde, bedeutet das nicht, dass der erwartete Fehler nicht in der Gesamtergebnismenge lag, sondern nur, dass die präferierte Analyse mit dem geringsten Fehlerwert einen unplausiblen Fehler beinhaltete. Die Analyse einiger Sätze dieser Kategorie führte allerdings eindeutig zu nicht intendierten Interpretationen der Sätze. Als Beispiel sei hier der Satz „Unfall passierte Neukölln" angeführt. Die intuitive Korrektur umfasst die Ergänzung eines Artikels zum Subjekt *Unfall* und die Einfügung der Präposition *in* vor der Ortsangabe *Neukölln*. Der Par-

ser liefert aber als syntaktische Fehleridentifizierung lediglich die Ergänzung des Artikels, da morphosyntaktisch der Satz so interpretiert wird, dass der „Entität“ Neukölln ein Unfall passiert ist. Dieser Fall ist verknüpft mit dem Beispiel (5.10) auf Seite 130. Wie dort ausgeführt, können Sätze mit diesem Fehlertyp grundsätzlich nicht von einem Parser analysiert werden, dem ausschließlich morphosyntaktische Informationen zur Verfügung stehen.

Obwohl eigentlich die Ersetzung einer Wortart durch eine andere in Promis*D* nicht direkt implementiert worden ist, können doch zwei Sätze mit dieser Fehlerkategorie vermeintlich erkannt werden. In dem Satz „Ich esse liebe Gemüse und Tofu“ ist nicht klar, welche Wortart der Lerner verwenden wollte beziehungsweise verwendet hat. Möglich als gesuchtes Zielwort ist beispielsweise der Komparativ *lieber* oder aber das Adverb *gerne*. Das System erkennt allerdings hier einen reinen Formfehler und interpretiert *liebe* als eine falsche Form für das Adjektiv *liebes*, sodass hier ein Kombinationsfehler von *liebe* mit dem falschen beziehungsweise nicht vorhandenen Artikel präferiert wird.

An dieser Stelle kann angemerkt werden, dass in dem hier realisierten Übungstyp eine gewisse Notwendigkeit für das Erreichen eines Parse-Ergebnisses besteht, da nur auf diese Weise eine Interpretation der Eingabe und eine Fortführung des Dialogs erreicht werden kann (siehe Kapitel 6). Da für ca. 85% der Fehler (ca. 82% aller fehlerhaften Sätze) überhaupt eine Beschreibung generiert werden konnte, kann also in der überwiegenden Zahl der Fälle die Fortführung des Dialogs auf eine thematische Interpretation und nicht auf die Analyse von Schlüsselwörtern zurückgeführt werden.

Schließlich gab es einige Sätze, die in ihrer fehlerhaften Form nicht geparst werden konnten. Der Maximalwert wurde bei der Analyse von 13 Sätzen erreicht und für weitere 12 Sätze konnte keine S-Kante gefunden werden (16,6% der Sätze), wobei diese Sätz zum Teil auch mehrere Fehler enthielten.[15]

Diskussion

Zusammenfassend soll nun etwas allgemeiner auf die Fehler eingegangen werden, zu denen der Parser a) keine Analyse liefern kann und b) durchweg eine Identifizierung leistet. Hervorzuheben ist zunächst, dass die hier entwickelte Methode zur Verschiebung von Phrasen für die HU/FU-Sätze keinen Erfolg zeigt, auch wenn es nur wenige Fehler dieses Typs gibt. Die entsprechenden Fehler aus dem Heringer-Korpus dagegen konnten wenigstens zum Teil identifiziert werden, sodass zumindest für die wenigen Sätze ca. die Hälfte alle Verschiebungsfehler erkannt werden. Die folgenden Fehler wurden im Einzelnen nicht identifiziert, das heißt, der den Fehler enthaltende Satz

[15] Die Zahl von 25 nicht analysierbaren Sätzen stimmt zufällig mit der Anzahl der Fehler überein, die in den nicht analysierbaren Sätzen enthalten waren. Für zwei Sätze wurde zwar der Maximalwert erreicht und trotzdem wurden einige S-Kanten generiert.

konnte nicht analysiert werden, wobei der Fehler zumeist in Kombination mit weiteren Fehlern auftrat.

- Zumeist handelt es sich um Sätze, in denen das Modalverb oder das Auxiliar (zum Teil in subordinierten Sätzen) falsch platziert ist (7 Sätze).
- Weitere 4 Sätze ließen sich durch das Einfügen eines Kopula- beziehungsweise Modalverbs oder eines Auxiliars korrigieren.
- In 3 Sätzen müsste ein mehr oder weniger kleines „Partikel“ gelöscht werden, nähmlich ein abtrennbares Präfix, ein überflüssiger Artikel oder eine Präposition, da das Verb ein direktes Objekt verlangt.

In dieser Analyse zeigt sich, dass insbesondere das hier entwickelte Verfahren zur Erkennung von Verschiebungsfehlern nur einige wenige Fehler erkennen kann. Als Beispiel sei der folgende Satz aus dem HU/FU-Korpus angeführt, der zusätzlich die grundsätzlichen Schwierigkeiten bei der (manuellen) Bestimmung eines Fehler verdeutlichen kann.

(5.24) *Ich gerne ein fremde Sprache zu lernen.

Meines Erachtens bieten sich zwei Korrekturmöglichkeiten beziehungsweise eine Erklärung an:

(5.25) a. Ich will/möchte gerne eine fremde Sprache lernen.
Korrektur: Modalverb einfügen, *zu* löschen, *ein* wird zu *eine*.
Erklärung: finites Verb fehlt, Modalverb erfordert reinen Infinitv, *ein* muss Eigenschaft [GEN = f] haben.

b. Ich lerne gerne eine fremde Sprache.
Korrektur: *lernen* verschieben, *zu* löschen, *ein* wird zu *eine*.
Erklärung: finites Verb fehlt, *lernen* muss Eigenschaft [MOOD = fin] haben, *ein* muss Eigenschaft [GEN = f] haben.

Abgesehen vom Artikelfehler wird also im Wesentlichen entweder ein Modalverb nach dem *Ich* eingefügt und *zu* wird gelöscht oder *lernen* wird als finites Verb eingefügt, wobei insbesondere der zweite Fall nicht von dem hier entwickelten Verfahren erfasst wird. Anzumerken ist außerdem, dass die weiteren Sätze, in denen Konstituentenfehler nicht analysiert werden konnten, eine ähnliche „Fehlerstruktur“ aufweisen. Notwendig ist vermutlich, wie auch in den Anmerkungen zu nur semantisch erkennbaren Fehlern in Abschnitt 5.2.2 erwähnt, die Möglichkeit, funktionale Einheiten oder Phrasen des Satzes „umherzuschieben“, bis sich eine mögliche Interpretation ergibt. Auch im HU/FU-Korpus findet sich ein Satz, der nicht auf der Basis einer rein morphosyntaktischen Analyse korrigiert werden kann („Ein Auto in Ampel schlägt.“). Abgesehen vom fehlenden Artikel für das Substantiv *Ampel* besteht der Fehler intuitiv in der Fehlstellung der PP, die an das Ende

des Satzes gerückt werden sollte. Für eine noch umfassendere Fehleranalyse müssten insbesondere die Möglichkeiten der Verschiebung von Phrasen erweitert werden, was aber den Hypothesenraum zusätzlich erweitern würde. Neben dieser Erweiterung, die immer noch ausschließlich morphosyntaktische Informationen nutzt, besteht eine zweite Möglichkeit, die zum Beispiel auch in Menzel (2002) favorisiert wird, in der verstärkten Einbeziehung von semantischen Informationen zur Generierung einer Analyse, die aber wie erwähnt den linguistischen Intuitionen im Rahmen der LFG widerspricht.

Im Zusammenhang mit den angeführten Fehlertypen stehen idiomatische beziehungsweise kollokative Wendungen, bei denen unklar ist, ob eine vorliegende Beschränkung zum Bereich der Syntax oder bereits zur Semantik gezählt werden muss. Die Problematik zeigt sich bereits bei sehr einfachen Beispielen: Der Heringer-Korpus enthält den Satz „Ich gehe zum Fuß.“, bei dem morphosyntaktisch gesehen kein Fehler vorliegt.[16] Allerdings ist die Kombination von *gehen* und „zum Fuß“ so ungewöhnlich, dass intuitiv sofort auf einen Fehler geschlossen werden kann. Diese Schwierigkeit spiegelt sich auch in der ausführlichen Diskussion in der Linguistik wieder, bei der mit Mühen versucht wird, Regularitäten zum Auftreten von Kollokationen zu entwickeln. Die Diskussion ist ein Hinweis darauf, warum es einerseits so schwierig für Sprachenlerner ist, sich diesen Bereich anzueignen und andererseits die Frage der Modellierung in der Grammatik völlig ungeklärt ist. Im Rahmen von Promis$D$ kann in diesen Fällen keine Analyse erreicht werden.

Schließlich konnten 5 Sätze nicht analysiert werden, in denen ein „einfacher“ Kongruenzfehler vorlag. In diesen Fällen war die Analyse zu ambig, sodass der Maximalwert und damit kein Analyseergebnis erreicht wurde. Im Rahmen der vorhergehenden Diskussion der Effizienz wurde dieser Aspekt, der unter anderem im Synkretismus von Wortformen begründet liegt, bereits angesprochen.

Im Gegensatz zu den überwiegend auf der Wortstellung basierenden Fehlern, für die nur in beschränktem Umfang Korrekturen vorgenommen werden, können sich widersprechende Features mit Hilfe eines sehr generellen Verfahrens in die F-Struktur integriert werden. Damit ist es möglich, fast alle denkbaren Kongruenz- und Rektionsfehler zu identifizieren, soweit sie in der Grammatik modelliert sind. Ein weiterer Fehlertyp, der in der F-Struktur angesiedelt ist, kann unter dem Begriff Kategorienfehler zusammengefasst werden. Dazu gehören beispielsweise Finitheitsfehler, Auxiliarfehler und andere Fehler, die zwar auch in der F-Struktur kodiert sind, aber nichts über ein formales Verhältnis von zwei Satzelementen zueinander aussagen. Eine letzte allgemeine Fehlerkategorie, die hier aufgeführt werden soll, sind Auslassungsfehler in der Wortfolge. Auch die letzten beiden Typen werden in sehr vielen Fällen identifiziert und vom vorgestellten Mechanismus korrigiert.

[16] Der Satz war nicht Teil der Evaluation.

Zusammenfassend lässt sich festhalten, dass zwar für 18% der Evaluationssätze gar keine Analyse zustande kommt, aber für mehr als die Hälfte aller Sätze der auch in der manuellen Korrektur präferierte Fehler identifiziert werden kann. Hervorzuheben ist außerdem, dass hier im Gegensatz zu anderen Beispielen aus der Literatur a) authentische Sätze aus Lernerkorpora verwendet wurden und b) ein breites Spektrum von möglichen Fehlern identifiziert werden kann. Vor allem Fehler in der Konstituentenstruktur werden im Gegensatz zu Kongruenz- und Rektionsfehlern vom vorgestellten Parser bisher nur unzureichend erkannt. Problematisch sind Sätze, bei denen ein Element „nach vorne“ verschoben werden müsste, oder Sätze, in denen multiple Verschiebungen notwendig sind, um einen korrekten Satz zu erhalten.

Nachdem in diesem Teil eine technisch orientierte Perspektive auf das Problem der Fehlererkennung eingenommen wurde, soll nun zur Lernerperspektive zurückgekehrt werden. Geklärt werden soll im Folgenden, welche Schritte zur Generierung einer Rückmeldung notwendig sind und welche Adaptationen in Bezug auf den Inhalt einer Rückmeldung möglich sind, um den Lerner individuell beim Verständnis der Fehler und damit beim allgemeinen Sprachlernprozess zu unterstützen.

5.5 Rückmeldungen

Der Parse-Vorgang hat im Sinne der Fehleranalysestufen[17] von Seite 40 in der F-Struktur lediglich zu einer Identifizierung beziehungsweise Erklärung des Fehlers geführt und in der K-Struktur zur Identifizierung und zur Korrektur, wobei die Erklärung wie erwähnt ausgelassen wurde. Da die Information, die der Parser über fehlerhafte Eingaben liefern kann, für einen Lerner zumindest ungewohnt sind, sollte eine Übersetzung in eine verständlichere Rückmeldung stattfinden. Wie in Kapitel 2 dargestellt, besteht aus didaktischer Sicht in Bezug auf herkömmliche, tutorielle CALL-Programme die dringende Notwendigkeit für präziseres Feedback auf fehlerhafte Lernereingaben. Auf welche Weise die Meldung in Promis*D* dem Nutzer an der Oberfläche präsentiert, wurde in Kapitel 4 kurz illustriert, sodass hier nun die technischen Aspekte näher beleuchtet werden können.

Zur Rückmeldung von Konstituentenfehlern sei daran erinnert, dass der Parser nur eine bereits korrigierte Konsituentenstruktur der Wortstellung zurückliefert. Falls eine Korrektur notwendig war, um den Satz vollständig zu parsen, muss die Information über die Art der Korrektur als zusätzliches Element in der Kante untergebracht werden. Dieser Bereich wird zu Beginn

[17] Ein Fehler kann registriert (Gab es einen Fehler?), lokalisiert (Wo war der Fehler?), identifiziert (Was war falsch?), evaluiert (Wie schwerwiegend war der Fehler?), erklärt/erläutert (Welche Rückmeldung gibt es?) und möglicherweise auch korrigiert (Wie lautet die korrigierte Form?) werden.

dieses Abschnitts erläutert. Im Anschluss daran folgen einige Ausführungen zur Behandlung einer fehlermarkierten F-Struktur, die im Gegensatz zur K-Struktur Informationen über beim Parsen aufgetretene Fehlerursachen direkt enthält, wobei allerdings eine Korrektur nicht enthalten ist. Um eine Rückmeldung einschließlich einer Korrektur zu liefern, muss also die F-Struktur unter Einbeziehung der K-Struktur analysiert werden. Zum Abschluss des Unterkapitels werden die Möglichkeiten untersucht, wie die Rückmeldungen an die Erfordernisse und Wünsche des Nutzers adaptiert werden könnten.

Die Generierung der Fehlermeldungen erfolgt mit Hilfe einer Übersetzungstabelle, in der alle relevanten Attribute, Werte usw. mit natürlichsprachlichen Ausdrücken, die dem Lerner aus pädagogischen Grammatiken bekannt sein sollten, verknüpft sind, wobei wie in Kapitel 3 erläutert kaum eine „Übersetzung“ notwendig ist, da die in der LFG verwendeten Konzepte und Strukturen direkt mit denen in pädagogischen Grammatiken korrespondieren. Zur Ausgabe werden diese mit einigen vorgefertigten Textbausteinen verknüpft, um eine vollständige Fehlermeldung zu erhalten. Da sich die Fehleranalyse auf morphosyntaktische Fehler beschränkt, ist diese Methode der „Generierung“ meiner Meinung nach für alle auftretenden Fälle ausreichend. Es müssen lediglich das Ergebnis der Fehlererkennung, der Status der Adaptation (siehe Abschnitt 5.5.4) und möglicherweise eine Korrekturvariante an das Ausgabemodul gereicht werden.

5.5.1 K-Struktur und Rückmeldungen

Die Generierung einer Rückmeldung über einen Konstituentenfehler gestaltet sich zunächst relativ einfach. Die Kante in der Chart, die den gesamten Satz überspannt, enthält unter anderem einen Eintrag, der die Modifikation des Satzes dokumentiert (siehe Abschnitt 5.2.1, Seite 129). Mit Hilfe dieses Eintrages wird dann eine natürlichsprachliche Meldung generiert, wie das folgende Beispiel verdeutlicht.

(5.26) Satz: $_0$ *ich* $_1$ *habe* $_2$ *Unfall* $_3$ *gesehen* $_4$

Kante: $(< 0,4 > ms \rightarrow np\ vp\ \cdot,\ 2,$ [[ins,2,*det*]])

Fehlermeldung: Als 3. Wort muss vermutlich ein Artikel eingefügt werden.

Dem Satz in Beispiel (5.26) fehlt offensichtlich der Artikel zu *Unfall*, um grammatisch zu sein (siehe auch Abbildung 4.9, Seite 93). Eine passive Kante aus der Chart, die hier vereinfacht dargestellt ist, besteht aus der Angabe der Spannweite der Kante ($< 0,4 >$), dem Mutterknoten (ms), dem „Baum“ ($np\ vp$), einem Fehlerwert (2) und der Annotation, dass an der Position 2 ein Artikel eingefügt worden ist ([ins,2,*det*]). Falls es mehrere

Korrekturen gegeben hat, sind diese in derselben Liste dokumentiert. Die Positionsangabe muss für die Rückmeldung um 1 erhöht werden, da in der Chart die übliche Zählung mit dem Beginn bei 0 verwendet wird.

Eine Erweiterung der Rückmeldung um Informationen aus der Grammatik ist nicht vorgesehen, das heißt, dass Erklärungen zu der Ursache eines Konstituentenfehlers nicht gegeben werden können. Für den Lerner wäre zwar zum Beispiel die Information sinnvoll, dass es sich bei dem Substantiv *Unfall* um ein „normales“ Substantiv handelt, das im Singular grundsätzlich zusammen mit einem Artikel verwendet werden muss. Wie in Abschnitt 5.3.2 (Seite 145) erläutert, lässt sich die relevante Information aber nicht für einen Nutzer aus der Grammatik extrahieren. PS-Regeln, in denen die Unterscheidung zwischen Abstrakta etc. und normalen Substantiven sowie der Verwendung der Artikel kodiert ist, sind auch nicht ohne weiteres für einen Lerner verständlich, sodass sie für die Generierung einer Meldung nicht infrage kommen. Auch die alternative Meldung „Numerusfehler“, die sich auf der Basis der Grammatik generieren ließe, ist für den Lerner unsinnig, sodass aus diesem Grund die Fehlererkennung in der F-Struktur an der PS-Regel, die Plural-Substantive ohne Artikel lizensiert, blockiert wird.

Im Folgenden werden zwei Vorschläge gemacht, die dennoch die gewünschte Erklärung liefern könnten. Eine Möglichkeit besteht in der Konstruktion einer „Erläuterungsdatenbank“ zu bestimmten Fehlertypen, die beim Auftreten eines derartigen Fehlers zur Erweiterung der Rückmeldung konsultiert werden kann. Wenn der oben angeführte Fehler auftritt, kann einerseits die Korrektur aus der Analyse gewonnen und dem Lerner mitgeteilt und andererseits eine Erläuterung zu dem Fehler regelbasiert aus einer Datenbank gewonnen werden, die den morphosyntaktischen Hintergrund näher erläutert:

(5.27) WENN *det* fehlt und n hat Feature [NTYPE = norm]
DANN Erklärung „Substantiv braucht Artikel, weil normal“

In diesem Fall handelt es sich allerdings um eine sehr spezielle Behandlung eines jeden Fehlers, sodass eine uniforme Methode für eine Vielzahl von Fehlertypen nicht erreicht wird.

Eine zweite Möglichkeit besteht in der Anreicherung des verwendeten Alphabets in der PS-Grammatik und/oder der F-Struktur (siehe zum Beispiel Heift, 1998). Die PS-Regel, die eine NP mit einem Artikel und einem Substantiv beschreibt, enthielte dann die zusätzliche Information, dass es sich um eine solche Konstruktion handelt. Wenn dann ein Artikel eingefügt werden muss, könnte mit Hilfe dieser speziellen Annotation ein Feedback an den Lerner ausgegeben werden. Allerdings ergibt sich aus dieser Strategie entweder die nochmalige Erweiterung der Elemente einer Kante oder die Anreicherung der F-Struktur mit Daten, die nichts mit dem eigentlichen Parsevorgang zu tun haben. Insbesondere letztere Möglichkeit scheidet aber aus, wenn die Anforderung bestehen soll, eine Grammatik zu verwenden,

die allein für den Zweck der morphosyntaktischen Beschreibung grammatisch korrekter Sätze modelliert worden ist.

5.5.2 F-Struktur und Rückmeldungen

Auch die Generierung einer Rückmeldung aus der F-Struktur stellt sich zunächst als ein einfacher Prozess dar.

(5.28) Satz: $_0$ *ich* $_1$ *habe* $_2$ *ein* $_3$ *Unfall* $_4$ *gesehen* $_5$

F-Struktur (Ausschnitt):

$$\left[\ \text{OBJ} \ = \left[\begin{array}{l}\text{GEN} \ = \text{m} \\ \hline \textit{err} \ = \left\{\ \left[\ \text{GEN} \ = \text{n}\ \right]\ \right\}\end{array}\right]\ \right]$$

Fehlermeldung:

Im (direkten) Objekt gibt es einen Fehler: Das Genus stimmt nicht: maskulin vs. neutrum

In diesem Beispiel besagt die extrahierte F-Struktur, dass ein Genus-Mismatch[18] innerhalb des Objekts vorliegt (siehe auch Beispiel (4.1) auf Seite 97). Da auf der Basis eines Widerspruchs zwischen den Werten eines Attributs keine Korrektur der Lexeme vorgenommen wird, muss diese im Nachhinein erfolgen, wenn die Rückmeldung nicht nur aus Information über die Attribute und ungefähren Kongruenz- oder Rektionsangaben bestehen soll. Allgemeiner gesehen kann also zunächst die Rückmeldung nur darin bestehen, dass entweder ein funktionales Element einen Fehler enthält, der sich auf eine bestimmte morphosyntaktische Eigenschaft bezieht, oder dass ein Fehler in Bezug auf ein subkategorisierendes beziehungsweise regiertes Element vorliegt.

Zur Rückmeldung einer erweiterten Erläuterung und einer Korrektur ist im ersten Schritt eine komplexere Analyse des Ableitungsbaumes zur Zuordnung der entscheidenden Attribute und ihrer Werte zu einzelnen Wörtern erforderlich. In einem zweiten Schritt müssen dann Lexeme im Lexikon identifiziert werden, deren Attribute den Widerspruch in der F-Struktur auflösen können. An obigen Beispiel soll das Verfahren erläutert werden. Die extrahierte Struktur, die nur die Informationen über den Fehler enthält, hat in etwas anderer Darstellung folgende Form.

(5.29)
$$\left[\ \text{OBJ} \ = \left[\begin{array}{l}\text{GEN} \ = \text{m} \\ \hline \textit{err} \ = \left\{\ \left[\ \text{GEN} \ = \text{n}\ \right]\ \right\}\end{array}\right]\ \right]$$

[18] Die Entscheidung, ob es sich bei diesem Fehler um einen Genus- oder um einen Kasus-Fehler handelt, ist wie erwähnt nicht trivial.

Der Ursprung des Widerspruchs lässt sich so auf den Teilbaum mit der Funktion [OBJ] einschränken, in dem alle Wörter auf die ursprünglichen Belegungen des Genus-Attributs untersucht werden müssen. Für das Beispiel gilt, dass für *Unfall* im Lexikon nur ein Eintrag vorhanden ist, der das Feature [GEN = m] besitzt, während für den Artikel zwar mehrere Einträge sowohl mit den Features [GEN = m] als auch [GEN = n] vorhanden sind, aber nur einer dieser Einträge mit den übrigen Features wie Numerus [NUM = sg] und Kasus [CASE = acc] korrespondiert. Aus dieser Korrekturhypothese lässt sich die Fehlermeldung generieren, dass der Artikel für eine Korrektur vermutlich von *ein* zu *einen* korrigiert werden muss.

Wenn ein Kongruenzfehler zwischen einem Verb und einem funktionalen Argument besteht, werden im Gegensatz zum vorhergehenden Beispiel zunächst Vorschläge für die Korrektur von Subjekt und Verb gemacht. Das folgende Beispiel zeigt die F-Struktur für den einfachen Satz „Ich lacht.“, für den der Parser eine Person-Inkongruenz ermittelt.

(5.30)
$$\left[\begin{array}{ll} \text{PRED} & = \text{'LACHEN <(SUBJ)>'} \\ \text{TENSE} & = \text{pred} \\ \text{SUBJ} & = \left[\begin{array}{ll} \text{NUM} & = \text{sg} \\ \text{PERS} & = 1 \\ \text{CASE} & = \text{nom} \\ \text{PRED} & = \text{'PRO'} \\ \hline \textit{err} & = \left\{ \left[\text{PERS} \quad = 3 \right] \right\} \end{array}\right] \end{array}\right]$$

*Ich lacht.

(5.31) Korrektur nach: [PERS = 1] → *Ich lache.*
[PERS = 3] → *Er lacht.*

Mit Hilfe des Lexikons kann sowohl ein Verb mit der Bedeutung „LACHEN“, mit dem der Satz korrigiert werden könnte („Ich lache“), als auch ein Pronomen bestimmt werden, das den Widerspruch auflösen kann („Er lacht“). Wenn das funktionale Element, beispielsweise das Subjekt komplex ist, müssen offensichtlich alle Wörter der NP überprüft und ein entsprechender Korrekturvorschlag gemacht werden. Als weitere plausible Korrekturmöglichkeit kommt intuitiv auch eine Korrektur des Verbs infrage, in dem das Tempus geändert wird und die Form *lacht* zu *lachte* korrigiert wird. Im Folgenden soll gezeigt werden, dass es sich anbietet, bestimmte Attribute als unveränderbar zu markieren, um möglichst wenig Korrekturhypothesen zu generieren.

Bestimmung der korrekten Form

Dem Lerner können bei einem Feature-Mismatch wie in Beispiel 5.31 gezeigt in bestimmten Fällen zwei Varianten der Korrektur angeboten werden: einmal die Version mit dem Feature direkt in der F-Struktur und eine

zweite Version mit dem Feature aus der Fehlerliste. Für den praktischen Fall aber unterliegt die Ermittlung der korrekten Wortformen verschiedenen Beschränkungen, das heißt, bei der Ermittlung eines korrekten Wortes können bestimmte Eigenschaften benannt werden, die als unveränderlich gelten müssen. Beispielsweise erfolgt das Nachschlagen trivialerweise nicht auf dem gesamten Lexikon, sondern ist auf die jeweiligen Wortarten beschränkt, da davon ausgegangen werden kann, dass diese invariant sind. Zusätzlich existieren weitere Einschränkungen zur Korrektur von einzelnen Wortarten, die berücksichtigt werden müssen. Die unveränderbaren Features sollten, wenn sie nicht die Fehlerursache selber darstellen, bei der Suche im Lexikon gewahrt bleiben, damit sichergestellt werden kann, dass eine sinnvolle Variante als Korrektur geliefert wird. Die folgende Tabelle bietet einen Überblick.

Wortart	unveränderbare Eigenschaft
det	definit/indefinit
prn	prontype
n	PRED-Wert, (Numerus)
a	PRED-Wert
v	PRED-Wert, Subkat-Liste, Tempus und Diathese
aux	Diathese und haben/sein-Feature

Tabelle 5.11: Unveränderbare Features bezogen auf die Wortart

Wenn also ein Artikel korrigiert werden muss, sollte mindestens die Eigenschaft definit/indefinit erhalten bleiben, was für unbestimmte Artikel unter Umständen ein Problem darstellt, da im Deutschen im Plural keine unbestimmten Artikel existieren. Ohne weitere Beschränkungen wird als Korrektur zu einen indefiniten Artikel im Singular mit Numerus-Fehler ein Artikel im Plural wie beispielsweise *einige* gefunden, der im implementierten Lexikon als Artikel modelliert ist (siehe Eisenberg, 1999, S. 144). Bei der Suche nach Pronomen muss lediglich der Typ des Pronomen konstant bleiben, damit eine korrekte Version gefunden werden kann. Substantive und Adjektive schließlich müssen die gleiche Bedeutung und Substantive zusätzlich einen identischen Numerus-Wert behalten. Verben sollten die Bedeutung, den Subkategorisierungsrahmen, das Tempus sowie die Diathese beibehalten, damit eine korrekte Form identifiziert werden kann. Weitere Wortarten scheinen sinnvollerweise nicht korrigierbar, offensichtlich insbesondere dann, wenn es sich um nicht flektierbare Wörter handelt. Zum Beispiel sollten Präpositionen (wie auch Adverbien und Ähnliches) vermutlich nicht verändert werden, da sie „nur“ den Kasus des Präpositionalobjekts regieren, es sei denn, ein Verb selegiert eine bestimmte Präposition.

Vollständigkeits- und Kohärenzfehler

Der letzte Typ von Fehler, der Fehlermeldungen produzieren kann, ist ein Subkategorisierungsfehler. Wie in Abschnitt 5.2.1 erläutert, wird mit Hilfe der Vollständigkeits- und Kohärenzprinzipien der LFG eine F-Struktur dahingehend überprüft, ob alle Subkategorisierungen erfüllt sind. Neben fehlendem Subjekt, Objekt oder satzartigem Komplement kann hier auch das Fehlen einer subkategorisierten PP erkannt werden.

Eine Rückmeldung mit diesem Fehlertyp enthält allerdings wie bei einem Feature-Mismatch nur die Erklärung, dass ein funktionales Argument fehlt beziehungsweise nicht an der erwarteten Position identifiziert werden konnte. Beispielsweise kann für den Satz „Plötzlich geht zum Verkäufer.“ aus Beispiel (5.8) auf Seite 129 einzig die Fehlermeldung ausgegeben werden, dass das Subjekt fehlt. Um zusätzlich einen Korrekturvorschlag zu generieren, müsste meines Erachtens ein vollständig neuer Parsevorgang gestartet werden, der aber gerade aus Effizienzgründen vermieden werden soll, sodass also im vorliegenden Rahmen nicht ohne weiteres ermittelt werden kann, dass das Subjekt nach dem finiten Verb eingefügt werden müsste.

Nach diesen Anmerkungen zur Generierung von Erklärungen und Korrekturen in Promis*D* soll nun auf einen weiteren Bereich eingegangen werden, der für die präzise Rückmeldung des erwarteten Fehlers entscheidend sein kann, der aber nicht implementiert worden ist. Es geht dabei um die Präferierung von Erklärungen für den Fall, dass sich aus der Analyse mehrere konkurrierende Möglichkeiten der Erklärung ergeben (Menzel, 1992, S. 106ff).

5.5.3 Erweiterte Fehlerpräferenzmöglichkeiten

Neben der Auswahl einer Fehlerhypothese aufgrund einer abstrakten, numerischen Bewertung kann es Fälle geben, in denen ein eher konkretes, linguistisches Kriterium für die Auswahl genutzt werden kann. Dieses ist besonders dann relevant, wenn es nicht „nur“ um die Korrektur, sondern um die Erklärung eines Fehlers gegenüber dem Lerner geht. Menzel (1992) unterscheidet für dieses Problem fünf Möglichkeiten der Selektion, die zum Teil nicht übertragbar sind, da sie auf seinem Konzept der Fehleranalyse beruhen. Die folgenden Kriterien zur Selektion stehen meines Erachtens hier zur Verfügung.

1. Merkmalspräferenz

Wenn zwei konkurrierende Fehlererklärungen existieren, die sich nur im Typ des Merkmals unterscheiden, kann aus der Präferenz eines bestimmten Merkmals die Wahl der intuitiven Erklärung folgen. Für diesen Fall lassen sich allerdings Beispiele konstruieren, bei denen der Fehlerwert der präferierten Lösung über dem der dispräferierten Analyse liegt, was die Festlegung auf eine Erklärung weiter verkompliziert. Im HU/FU-Korpus befindet sich der

folgende Satz, bei dem die (möglicherweise) präferierte Rückmeldung „Kasusfehler“ aus der Analyse mit dem höheren Fehlerwert konstruiert werden sollte.

(5.32) *Ich habe direkt ein Unfall gesehen.[19]

Fehlerwert 1:

$$\left[\begin{array}{l}\text{DIROBJ} = \left[\begin{array}{l}\text{GEN} = \text{m} \\ \hline err = \left\{ \left[\text{GEN} = \text{n} \right] \right\}\end{array}\right]\end{array}\right]$$

Fehlerwert 2:

$$\left[\begin{array}{l}\text{DIROBJ} = \left[\begin{array}{l}\text{CASE} = \text{acc} \\ \hline err = \left\{ \left[\text{CASE} = \text{nom} \right] \right\}\end{array}\right]\end{array}\right]$$

Im Vergleich mit dem schon einmal angeführten Satz „Der Götter zürnen.“ (Seite 141) stellt die generelle Präferenz für den Kasusfehler allerdings ein Problem dar, da für diesen Satz gerade nicht die Kasusfehler-Analyse präferiert ist. Im Widerspruch zum Beispiel (5.32) sollte man außerdem für das Deutsche generell annehmen, dass das Genus der Substantive als Fehlerquelle bevorzugt wird (Menzel, 1992, S. 110). Es zeigt sich also, dass sich zwar Beispiele konstruieren lassen, für die bestimmte Merkmalspräferenzen die geeignete Korrektur und Erklärung generieren, dass aber allgemein gültige Regeln nur schwierig zu konstruieren sind und von weiteren Faktoren beeinflusst werden.

2. Funktionspräferenz

Insbesondere bei der Analyse des Deutschen kann ein Fall auftreten, in dem der Parser nicht eindeutig entscheiden kann, welche Phrasen eines Satzes Subjekt und Objekt sind. Für den Beispielsatz „Die Mann sieht die Frau.“ besteht in jedem Fall ein Numerus- oder ein Genusfehler innerhalb der ersten NP, ohne das geklärt wäre, ob die NP das Subjekt oder das Objekt darstellt. In diesem Fall kann es sich anbieten, das Kriterium der Funktionspräferenz anzuwenden, um entweder die Korrektur des Artikels von *die* zu *der* (Subjekt) oder zu *den* (Objekt) vorzunehmen. Da die Grundwortstellung das Subjekt an erster Stelle vor dem finiten Verb und das Objekt nach dem Verb vorsieht, sollte im Fall einer Auswahl die Analyse gewählt werden, die das Subjekt vor dem Objekt festlegt.

3. Grundformpräferenz

Das Lernen von Vokabeln basiert üblicherweise besonders bei flektionsreichen Lexemen auf einigen wenigen Grundformen. Wie Menzel (1992) kann man annehmen, dass die Eigenschaften einer Grundform wie zum Beispiel Kasus Nominativ bevorzugt werden sollten, wenn es eine Auswahl an Alternativen gibt. Für das von ihm angeführte Beispiel „Der Götter zürnen.“

[19] Wie in Abschnitt 5.3.1 erläutert ergeben sich die unterschiedlichen Bewertungen aus den verschiedenen Zeitpunkten, zu dem der Feature-Clash aufgetreten ist. Der Fehlerwert 1 ergibt sich bei der Unifikation der F-Strukturen von Artikel und Substantiv, während sich der Fehlerwert 2 bei der Unifikation von Objekt und Verb ergibt.

macht die Annahme der Grundform Sinn, da auf diese Weise der intendierte Numerusfehler mit einer allgemeinen Maßgabe präferiert werden kann.

Zwei Aspekte sind in diesem Zusammenhang wichtig. Erstens gilt die Grundformpräferenz vermutlich nur für bestimmte Features, wie zum Beispiel Kasus, da eine Präferenz für das Numerus kaum Sinn macht. Schließlich sollte der Satz „Der Götter zürnen.“ nicht zu Subjekt im Singular korrigiert werden, auch wenn man unter Umständen von einer „Singularpräferenz“ ausgehen kann, da es sich hierbei um die Grundform handelt. Zweitens scheint es hier eine starke Interaktion mit der morphologischen Komplexität zu geben. Gemeint ist, dass wenn eine morphologisch oder orthografisch komplexe Form gewählt worden ist, diese bewusst gewählt worden ist und daher eher nicht korrigiert werden sollte. Für einen Satz wie „Der Mann lachen.“ bedeutet das, dass nicht etwa das Subjekt von Singular zu Plural geändert werden sollte, obwohl damit das Verb in seiner „Grundform“ erhalten bliebe, sondern das Verb muss in die dritte Person, Singular geändert werden, damit eine für den Lerner plausible Korrektur erreicht wird.

Auch wenn diese Möglichkeiten zur Fehlerpräferenz hier nicht weiter ausgewertet und evaluiert worden sind, zeigt sich allerdings an den Ergebnissen der Evaluation in Abschnitt 5.4.2, dass für den überwiegenden Teil der Fehler, die in der F-Struktur kodiert werden, die erwartete Erklärung ohne zusätzliche Regeln präferiert wird (71% der Fehler in der F-Struktur). Schließlich erscheint mir die Interaktion der unterschiedlichen Präferenzmöglichkeiten sehr komplex, sodass damit meines Erachtens kaum ein sinnvolles, das heißt möglichst uniformes Regelwerk mit wenigen Ausnahmen konstruiert werden könnte.

Die Auswahl einer möglichen Präferenzstrategie gestaltet sich schließlich durch die in Beispiel 5.21 (Seite 144) angeführte Nutzung von zusätzlichen Relationen schwieriger. Hier muss unter anderem eine Abwägung zwischen der Möglichkeit zur präzisen Rückmeldung und der Verwendung von bestimmten Constrainttypen beispielsweise der Negation stattfinden. Als Beispiel sei hier sowohl der Eintrag zum Lexem *Mann* als auch zu *Mensch* angeführt. Der erste Eintrag enthält zum Kasus nur die Information, dass er nicht Genitiv sein darf, während der zweite Eintrag das Feature [CASE = nom] beinhaltet.

(5.33) Mann, N,
$$\begin{bmatrix} \text{NTYPE} & = \text{norm} \\ \text{NUM} & = \text{sg} \\ \text{CASE} & \neq \text{gen} \\ \text{PRED} & = \text{‘MANN’} \end{bmatrix}$$

(5.34) Mensch, N,
$$\begin{bmatrix} \text{NTYPE} & = \text{norm} \\ \text{NUM} & = \text{sg} \\ \text{CASE} & = \text{nom} \\ \text{PRED} & = \text{‘MENSCH’} \end{bmatrix}$$

Für den Satz „Den Mann lacht.“ ermittelt der Parser einen Kasusfehler des Subjekts, da für das Subjekt Akkusativ ermittelt wurde. Als Korrektur wird sinnvollerweise nur der Artikel korrigiert, da *Mann* auch Nominativ sein kann. Dagegen wird für den Satz „Den Mensch lacht.“ der Kasusfehler innerhalb des Subjekts diagnostiziert. Für das Lexem *Mensch* existiert nur der eine angegebene Eintrag (5.34), da die anderen Formen *Menschen* lauten. Auch hier läuft die Korrektur offensichtlich auf die Ersetzung von *den* hinaus.

Die Schwierigkeit dieser Analyse besteht in den beiden unterschiedlichen Rückmeldungen, obwohl es sich aus der Lernerperspektive um identische Fehler handelt, für die idealerweise also auch identische Fehlermeldungen produziert werden sollten. Im vorliegenden Fall ist das aber nur möglich, wenn die unterschiedlichen Kasus für das Lexem *Mann* explizit im Lexikon berücksichtigt werden, was zur Konsequenz hätte, dass sich die Einträge vervielfachen. Eine weitere Alternative wäre die Integration von Wertemengen beziehungsweise Disjunktionen von atomaren Werten, die aber auch zur Folge hätte, dass die Analyse einer Eingabe komplexer würde, da mehr unterschiedliche Lexikoninformationen berücksichtigt werden müssten. Einerseits wird hier also mit Hilfe der Komprimierung der lexikalischen Einträge zum Beispiel durch die Negation ein kompaktes Lexikon ermöglicht, während andererseits in Bezug auf die Fehlerbestimmung eine gewisse Beliebigkeit eingeführt wird (siehe auch die Diskussion in Menzel, 1992, S. 30ff). In Promis*D* wurde dieser Nachteil zugunsten der Effizienz in Kauf genommen.

Damit ist der Abschnitt zu möglichen Erweiterungen der Generierung von Rückmeldungen abgeschlossen, in dem einerseits gezeigt wurde, welche zusätzlichen Möglichkeiten neben der numerischen Bewertung von Analysen bestehen und andererseits, dass bestimmte Implementationsentscheidungen zwar die Rückmeldungen „unsauberer“ erscheinen lassen, aber für eine kompaktere Lexikonrepräsentation sorgen. Im Folgenden soll zum Abschluss des Kapitels schließlich auf die von den Analysemöglichkeiten unabhängige, lernerorientierte Adaptation der Rückmeldungen eingegangen werden.

5.5.4 Adaptation der Fehlermeldungen

Eine erweiterte Möglichkeit zur Rückmeldung, die sich aus der tiefen Analyse der Lernereingaben ergibt, besteht in der Adaptation des Feedbacks an die Kenntnisse des Lerners (siehe zum Beispiel Heift, 2001; Menzel, 1992). „Kenntnisse“ ist hier ein sehr allgemeiner Begriff, der sich nicht nur auf den jeweiligen Sprachstand bezieht, sondern beispielsweise auch die Kenntnisse eines Lerners in linguistischer Terminologie umfasst. Beachtenswert ist außerdem, dass es in den folgenden Erläuterungen um die Erklärung und nicht um die Korrektur einer fehlerhaften Lernereingabe geht, da ja die Korrektur lediglich aus der Präsentation korrigierter Wortformen besteht.

Verschiedene Aspekte der Adaptation sollten den Lernprozess in sinn-

voller Weise unterstützen.

- Erstens kann mit Hilfe einer angepassten Fehlermeldung für ein schnelles Verständnis gesorgt werden, indem der Lerner nicht mit dem Verstehen des Meldungstextes aufgehalten wird. Gerade in den hier vorgestellten kommunikativen Übungsaufgaben bietet es sich an, als zusätzliche Abstufung zum optionalen Aufruf der Meldung den Text so zu gestalten, dass er das Verständnis des Fehlers optimal unterstützt.

- Ein zweiter Aspekt ist die Vermeidung von Frustration bei der Verwendung des Systems und insbesondere bei der Nutzung der ausführlichen Fehlermeldungen. Gerade die in kommerziellen Systemen erzeugten Fehlermeldungen geben nach Ansicht von Didaktikern, wie auf Seite 35f. erwähnt, Anlass zur Frustration. Mit Hilfe von an den Lerner, an die Aufgabenstellung und an das Lernziel angepassten Meldungen kann diesem Effekt entgegen gewirkt werden.

- Schließlich kann die Formulierung einer Fehlermeldung auch als Herausforderung und zusätzliche Lernaufgabe gesehen werden. Mit Hilfe des im Meldungstext präsentierten Sachverhalts kann neues Wissen über Sprache und die zur Beschreibung eingesetzte Terminologie vermittelt werden. Der Text sollte eine dem Lernprozess angepasste Unterstützung darstellen, indem der Lerner mit seinem individuellen Wissen über die Sprache und über Sprachbeschreibung berücksichtigt, aber auch gefordert wird.

Wenn Fehlermeldungen an die Kenntnisse eines Lerners adaptiert werden sollen, können meines Erachtens mehrere Dimensionen unterschieden werden. Diese Dimensionen, die im Folgenden dargestellt werden, schließen sich nicht notwendigerweise aus, sondern verlaufen zum Teil auch orthogonal zueinander. Allerdings kann eine direkte Übertragung der separaten Dimensionen in das System vermutlich nicht realisiert werden, das heißt, einem Lerner kann nicht eine Auswahl in jeder Dimension geboten werden, sondern es müssen bestimmte Zusammenfassungen einzelner Dimensionsebenen vorgenommen werden. Schließlich muss darauf hingewiesen werden, dass auch ein Modul zur Lernermodellierung die folgenden Dimensionen nutzen könnte, um automatisch individuelles Feedback zu liefern.

Linguistische Terminologie

Eine erste Dimension ist die Verwendung linguistischer Terminologie in der Erläuterung an den Lerner. Für den Lerner einer Sprache, der gewisse Kenntnisse in sprachwissenschaftlicher Terminologie hat, lassen sich damit die Fehler knapp und sehr präzise beschreiben. Für Lerner, die in dieser Hinsicht keine Vorerfahrung haben, müssen entweder Beschreibungen mit eher „klassischer“ Terminologie zur Verfügung gestellt werden oder es kann in einer

weiteren Dimension stärker Bezug auf die tatsächlich verwendeten Wörter genommen werden, um damit einen leichter verständlichen Meldungstext zu generieren. Ein Beispiel aus Heift (2001, S. 101f) kann dies verdeutlichen, in dem allerdings mehrere Dimensionen miteinander vermischt sind, die im Anschluss getrennt dargestellt werden.

(5.35) *Lernereingabe*: Der Zeit läuft.

Feedback Advanced:	Da ist ein Genusfehler bei dem Subjekt.
Feedback Beginner:	DER von DER ZEIT ist falsch.
	ZEIT ist nicht männlich.

Im ersten Feedback „Advanced“ in diesem Beispiel werden die Begriffe *Genus* und *Subjekt* verwendet, deren Verständnis ein gewisses Maß an linguistischem Wissen voraussetzt. Das Feedback für den „Beginner“ referiert in erster Linie auf die vom Lerner verwendeten Wörter und verwendet nur den Begriff *männlich*. Das folgende Beispiel für dieselbe Eingabe kann die Dimension „Linguistische Terminologie“ noch einmal stärker verdeutlichen, wobei die Unterpunkte a. und b. jeweils beispielhafte Realisierungen der Extreme sind.

(5.36) 1. Dimension: Linguistische Terminologie

a. Das Substantiv im Subjekt regiert den Artikel hinsichtlich des Genus, der aber unterschiedlich ist.
b. DER ZEIT hat die falsche Form.
   DER passt dem Geschlecht nach nicht zu ZEIT.

Explizitheit

Zusätzlich enthalten die Hinweise aus Beispiel (5.35) eine Staffelung in der Dimension der Explizitheit der Angaben. Eine Verbalisierung kann präzise den Fehler erklären und zusätzlich Bezug auf eine mögliche Korrektur nehmen. Während im ersten Feedback (5.35 a.) nur ein Anhaltspunkt geliefert wird, bei dem der Lerner selber die Analyse zur Erreichung einer Korrektur fortsetzen muss, wird dagegen im zweiten Feedback (5.35 b.) zwar eine Korrekturmöglichkeit selber nicht erwähnt, aber es werden sehr deutliche Hinweise gegeben. Im folgenden Beispiel (5.37) wird dieser Aspekt noch einmal stärker hervorgehoben.

(5.37) 2. Dimension: Explizitheit

a. Im Subjekt gibt es einen Rektionsfehler.
b. Im Subjekt gibt es einen Genusfehler.
   Das Substantiv regiert den Artikel hinsichtlich des Genus.
   Der Artikel ist aber maskulin und das Substantiv ist feminin.

Konkretisierung

Als eine dritte Dimension kann meines Erachtens das Einbeziehen der Lernereingabe angesehen werden. Das Advanced Feedback (5.35 a.) kann als ein Extrem angesehen werden, da hier kein direkter Bezug zu Wörtern oder Phrasen der Eingabe hergestellt wird, sondern nur auf das Subjekt referiert wird, das vom Lerner zusätzlich identifiziert werden muss. Dagegen wird im Feedback für „Beginner“ die Lernereingabe mit in die Erklärung übernommen und anhand der konkreten Wörter wird eine Erläuterung gegeben. Wiederum soll das folgende Beispiel diese Dimension verdeutlichen.

(5.38) 3. Dimension: Konkretisierung

a. Im Subjekt gibt es einen Genusfehler.
   Die Rektion des Artikels durch das Substantiv verlangt ein anderes Genus.

b. Im Subjekt DER ZEIT gibt es einen Genusfehler.
   Die Rektion des Artikels DER durch das Substantiv ZEIT verlangt ein anderes Genus.

Ein zusätzlicher Aspekt, der sich mit einer tiefen Analyse realisieren lässt, ist die Sequenzialisierung der Fehlermeldungen (siehe auch Menzel, 1992, S. 215ff). Damit wird einerseits die Ausgabe von multiplen Fehlermeldungen in einzelne, leichter verständliche Abschnitte erreicht. Andererseits kann damit die Ausgabe von komplexen Fehlermeldungen strukturiert werden. Nach Aufforderung durch den Lerner werden Meldungen mit zunehmender Informativität beziehungsweise Präzision dargestellt. Wenn der Lerner glaubt, die Fehlermeldung verstanden zu haben, kann die Sequenz abgebrochen werden.[20]

Tatsächlich implementiert worden ist in experimenteller Form die Dimension der Explizitheit und die Dimension der linguistischen Terminologie, das heißt also einerseits eine Unterscheidung in der Ausführlichkeit und andererseits von „pädagogischer“ und „wissenschaftlicher“ Terminologie, die allerdings nicht weiter evaluiert worden ist. Allgemeiner gesehen kann die computerlinguistische Analyse dazu genutzt werden, nach unterschiedlichen Dimensionen strukturierte Rückmeldungen zu generieren, um auf diese Weise ein leichteres Verständnis der Meldungen zu unterstützen, und außerdem den Lernprozess nicht nur über korrekte Sprache, sondern auch über sprachliches Wissen zu fördern.

Zusammenfassend lässt sich für den Abschnit „Rückmeldung“ festhalten, dass für den Bereich der K-Struktur Rückmeldungen aus Annotationen zu

[20] Da im hier vorgestellten Übungstyp eine Korrektur der Eingabe durch den Lerner nicht sinnvoll ist, kann allerdings aus einem Abbruch nicht gefolgert werden, dass der Lerner die Fehlermeldung tatsächlich verstanden hat.

den Kanten generiert werden, falls Korrekturen für das erfolgreiche Parsing der Eingabe notwendig waren. Erklärungen zu den Fehlern lassen sich meines Erachtens kaum aus der „reinen“ Grammatik extrahieren; Erklärungen können allerdings entweder als Zusätze in die Grammatik integriert werden oder aus einer Datenbank gezogen werden, die Erläuterungen zu unterschiedlichen Fehlertypen, beispielsweise grundlegende Satzmuster, enthält.

Für Fehler, die in der F-Struktur aufgetreten sind, ergibt sich genau die umgekehrte Situation. Die Fehler-Attribute liefern zwar Informationen zu den betroffenen Eigenschaften beispielsweise in der Kongruenz oder Rektion, das heißt eine Erklärung für einen Fehler, aber eine Korrektur kann erst nach Abschluss des Parsings mit Hilfe des gesamten Ableitungsbaumes vorgenommen werden. In vielen Fällen sind auf diese Weise präzise Rückmeldungen sowohl mit Erklärungen als auch Korrekturen auch ohne linguistisch motivierte Präferenzregeln möglich, wie die Evaluation gezeigt hat. Wie oben erläutert bieten sich verschiedene Präferenzstrategien an, die aber nicht implementiert und evaluiert worden sind, unter anderem, da sich kein uniformes Regelwerk zu ergeben scheint. In Promis*D* sind die Möglichkeiten der adaptiven Gestaltung des Feedbacks nur rudimentär realisiert.

5.6 Fazit

In diesem Kapitel wurde ein Konzept für die Analyse von morphosyntaktischen Fehlern ausschließlich auf der Grundlage des Verfahrens enwickelt, das zweigeteilt die Grundstrukturen der LFG, K- und F-Strukturen nutzt. Konstituentenfehler werden mit Hilfe eines veränderten Earley-basierten Parsingverfahrens erkannt, in dem wie üblich aktive Kanten in die Chart eingetragen werden, die alle Hypothesen für eine mögliche, korrekte Fortsetzung des Satzes enthalten. Falls diese Hypothesen aber nicht erfüllt werden können, werden mit Hilfe eines Triggers, der durch bestimmte Chartkonfigurationen ausgelöst wird, lexikalische Dummy-Elemente in die Chart eingefügt und Wörter beziehungsweise Phrasen „aus der Chart entfernt“, um sie möglicherweise an einer späteren Position wieder einzufügen. Insbesondere für letzteren Fall der Fehlplatzierung von Phrasen sind meines Wissens nur sehr wenige Verfahren entwickelt worden. Dass Phrasen dabei nur von links nach rechts verschoben werden können, scheint nach der Evaluation nur für bestimmte Konstruktionen von Bedeutung. Falls sich mit dieser Korrektur der normale Parsevorgang bis zum Ende fortsetzen lässt, ist eine mögliche Fehlerquelle gefunden. Der Fehler wird als Annotation an die Kante angefügt, die zur Beschreibung des Satzes dient, und zur S-Kante propagiert. Damit ist die Markierung dann auch in der passiven Kante enthalten, die den gesamten Satz überspannt. Da die Analyse eines Satzes auf der korrigierten Version der K-Struktur basiert, muss die Korrektur nicht aus einer Struktur extrahiert werden, das heißt, eine Fehlermeldung zu Konstituentenfehlern

kann also zwar eine korrigierte Version und die dafür notwendige Korrektur präsentieren, eine präzise Fehlermeldung mit einer Erklärung ist aber mit zusätzlichen Ressourcen möglich.

Kongruenz- und Rektionsfehler sowie weitere Kategorienfehler werden mit Hilfe der F-Struktur identifiziert. Falls bei der Unifikation zwei identische Attribute mit unterschiedlichen atomaren Werte unifiziert werden sollen, wird ein Mechanismus aktiviert, der die sich widersprechenden Werte in einer Struktur integriert. Damit ist die Ursache für den Fehler in der F-Struktur markiert und kann nach dem Parsing zur Erklärung und zur Korrektur verwendet werden. Die Herkunft und Korrektur eines Fehlers wird nicht in der resultierenden F-Struktur gespeichert, sondern wird erst am Ende des Parsevorgangs mit Hilfe des Ableitungsbaumes erschlossen. Für die F-Struktur gilt also der umgekehrte Fall wie für die K-Struktur: Die Erklärung eines Fehlers kann direkt aus der F-Struktur an den Lerner ausgegeben, eine Korrektur muss aber mit Hilfe des Ableitungsbaumes generiert werden.

Zwei verschiedene Mechanismen werden in Promis*D* eingesetzt, um in kontrollierter Weise die Generierung von Fehlerhypothesen zu steuern. Für die Generierung von zusätzlichen Fehlerkanten in der Chart wurde ein Trigger entworfen, der nur bei einer bestimmten Chart-Konfiguration die Erweiterung der Chart um Fehlerhypothesen an genau definierten Positionen erlaubt. Zusätzlich wurden sowohl die Features, bei denen ein Fehler auftreten darf wie auch die POS, die in die Chart eingefügt werden, auf die in Korpora relevanten Fehlertypen eingeschränkt, sodass auch hier der Hypothesenraum für die Analyse von fehlerhaften Eingaben kontrolliert geöffnet werden kann.

Die Analyse der Lernerkorpora und die Evaluation demonstrieren, dass sich mit Hilfe der hier entwickelten Verfahren große Bereiche von möglichen morphosyntaktischen Fehlern identifizieren lassen. Obwohl es wünschenswert wäre, können semantische Fehler nicht identifiziert werden, was im Rahmen der Nutzung (des morphosyntaktischen Apparats) der LFG außerdem nicht der linguistischen Intuition entspricht. Somit ist ein System entstanden, dass in großen Teilen morphosyntaktische Lernerfehler erkennen, korrigieren und erklären kann. Entlang verschiedener Dimensionen lassen sich die Fehlermeldungen an die Erfordernisse des Lerners anpassen. Der Lernprozess wird gefördert, indem einerseits in den Dialogaufgaben ein adaptives Feedback geliefert und andererseits der identifizierte Fehler erklärt und/oder korrigiert wird.

Kapitel 6

Sonstige Module

Sowohl die Dialogsteuerung als auch einige weitere kleine Module wie beispielsweise die Orthografie- und die Semantikkontrolle sind Bestandteile, die für ein System wie Promis*D* unerlässlich sind. Allerdings sind sie nicht der Schwerpunkt dieser Arbeit und werden daher hier nur kurz vorgestellt.

Die Dialogsteuerung, die zunächst betrachtet wird, sollte einerseits so gestaltet sein, dass dem Lerner ein größtmögliches Maß an Flexibilität gewährt wird, und andererseits so eng gefasst sein, dass das System in der Lage ist, fehlerhaftes Verhalten auch identifizieren zu können. Zwischen diesen beiden Polen muss meines Erachtens ein Abwägen stattfinden, da mit der Einschränkung der Flexibilität auch der Hypotheseraum eingeschränkt wird und damit umgekehrt die Chancen der Fehlererkennung steigen. Wenn bei der Implementierung außerdem ein modulares Konzept verfolgt wird, besteht die Möglichkeit, auf einfache Art und Weise das System zu modifizieren beziehungsweise zu erweitern. Um diese Aspekte zu realisieren, verwendet die Dialogsteuerung in Promis*D* zwei getrennte Wissensbasen, wobei die eine Informationen zur Form von Frage-Antwort-Dialogen und die andere Informationen über den Inhalt eines Gespräches enthält. Aus diesem Konzept folgt, dass in allen bisher implementierten Dialogsituationen der Rechner als Dialogpartner die Fragen stellt und der Lerner dazu aufgefordert ist, die Fragen mit einem Satz zu beantworten. Die Wissensbasis zum Dialoginhalt ist hierarchisch nach der Ausführlichkeit der Informationen aufgebaut, sodass durch den Einsatz von konjunktiven und disjunktiven Verknüpfungen zwischen verschiedenen Informationsknoten sowohl Informationen bis zu einer bestimmten Ausführlichkeit als auch unterschiedliche Informationen wahlweise abgefragt werden können. Für den Lerner bedeutet das, dass ein relativ flexibler Dialog geführt werden kann, das heißt, aufgrund der tiefen Analyse und der damit verbundenen Abstrahierung von den eingegebenen Wörtern beziehungsweise Sätzen besteht die Möglichkeit, einen Dialog in vielen Varianten zu bearbeiten.

Im zweiten Teil des Kapitels werden die Module zur Orthografie- und

zur Semantikkontrolle vorgestellt und motiviert. Wie schon in Abschnitt 4.2 angedeutet, findet keine Verzahnung der Orthografiekontrolle mit der Erkennung von morphosyntaktischen Fehlern statt, sodass alle Wörter der Eingabe bis auf großgeschriebene Wörter im Lexikon enthalten sein müssen, um eine morphosyntaktische Analyse zu ermöglichen. Trotzdem wird dem Lerner nicht nur mitgeteilt, dass ein Wort nicht im Lexikon des Systems enthalten ist, sondern er wird durch das System insofern unterstützt, als dass eine Anzahl von „ähnlichen“ Wörtern präsentiert wird. Semantische Fehler, das heißt Verletzungen von Selektionsrestriktionen werden zwar auf der Basis einer Wissensbasis ermittelt, aber deren Auswirkungen werden nicht als separate Fehlermeldungen präsentiert, sondern als Reaktion des Dialogpartners, wie in Abschnitt 4.2 erläutert.

6.1 Dialogsteuerung

Zunächst wird das Konzept der Dialoggrammatik oder auch Diskursgrammatik vorgestellt, das dazu dient, einen Dialog zu modellieren, im Fall von Promis*D* einen Frage-Antwort-Dialog. Aufbauend auf einer Dialoggrammatik wird inhaltliches Wissen benötigt, damit das Programm sinnvoll auf die Eingaben des Lerners reagieren kann. Der Aufbau dieser Wissensbasis wird anschließend beschrieben.

Zunächst sollen hier einige Aspekte aus Kapitel 2 kurz wiederholt werden, um als Basis für die technischen Anforderungen die Lernerperspektive noch einmal darzustellen. Dort wurde gezeigt, dass es aus didaktischer Perspektive wünschenswert ist, wenn Aufgaben, die mit Hilfe des Computers bearbeitet werden, den Erwerb einer Reihe von Wissenstypen unterstützen, um einen umfassenden Lernprozess zu gewährleisten. Um den Erwerb der kommunikativen Kompetenz zu unterstützen, müssen Aufgaben in erster Linie zum Gebrauch der Sprache in kommunikativen Situationen auffordern, wozu im Unterricht vor allem kleine „Dialogsituationen“ geschaffen werden, in denen der Lerner seine kommunikativen Fähigkeiten durch den aktiven Gebrauch der Sprache trainieren kann. Insbesondere „traditionelle Lern- und Übungsprogramme“ als auch „tutoriell orientierte, geschlossene Multimedia-Anwendungen“ (siehe Seite 17) unterstützen, wie ausgeführt, den Lernprozess nicht. Vor allem der aktive Gebrauch der Sprache wird in computerbasierten Übungen durch die Konzentration auf einzelne sprachliche Phänomene üblicherweise stark eingeschränkt. Das heißt, dass vor allem von einem größeren Kontext unabhängige Aspekte der Sprache im Mittelpunkt stehen. Selbst wenn in den Übungen ein Kontext geschaffen wird, stehen doch meistens „formale“ Aspekte, wie zum Beispiel die Formen und der Gebrauch von Auxiliaren oder unregelmäßige Verbformen im Vordergrund. Bei anderen Übungen, wie dem Bearbeiten von Lückentexten, muss zwar vom Lerner der Zusammenhang zwischen den Sätzen konstruiert wer-

den, um eine Lücke füllen zu können, aber die „Leistung“ des Lerners kann üblicherweise nur in der Produktion von einzelnen Wörtern bestehen, die durch die Aufgaben stark determiniert werden.[1] Ein Programm, das einen Frage-Antwort-Dialog simuliert, kann, wie schon in den vorhergehenden Kapiteln ausgeführt, einige dieser Bereiche deutlich verbessern, das heißt, dass insbesondere die Übungsform des Dialogs dazu genutzt werden kann, den Lerner zum aktiven Gebrauch seiner Sprachkenntnisse anzuregen.

Eine Möglichkeit, einen Frage-Antwort-Dialog unabhängig vom Inhalt zu modellieren, besteht in der Verwendung einer so genannten Dialog- oder Diskursgrammatik. Dieses Konzept ist schon verschiedentlich in der Literatur vorgeschlagen worden, zum Beispiel in Fawcett und Taylor (1989) oder Carberry (1990). Dieser spezielle Grammatiktyp hat in Dialogsystemen vor allem die Aufgabe, die Interpretation einer Eingabe an einer beliebigen Stelle in der Abfolge der Dialogbewegungen zu unterstützen. Wenn zum Beispiel vom System eine Frage an den Nutzer gestellt wurde, sollte die darauf folgende Reaktion sinnvollerweise entweder als Antwort auf die Frage oder als Nachfrage interpretiert werden. Abbildung 6.1 zeigt den vereinfachten Kern einer solchen Grammatik, die die Abfolge der in Promis$D$ präsentierten Fragen und erwarteten Antworten modelliert.

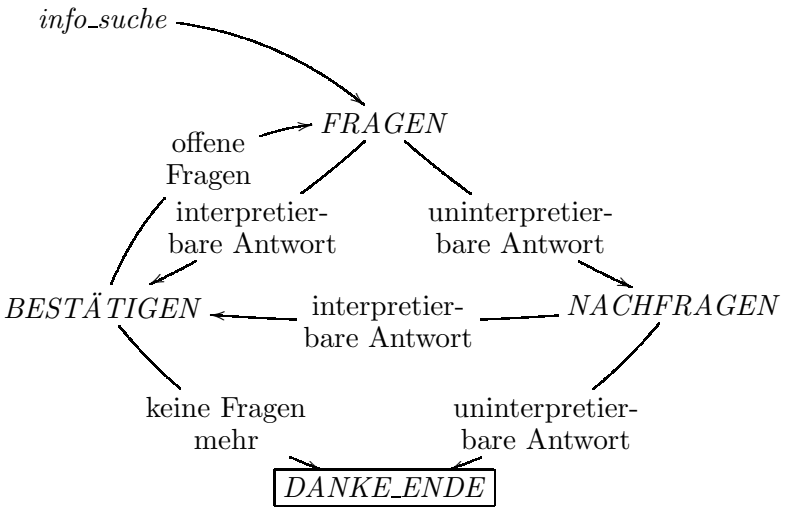


Abbildung 6.1: Vereinfachte Diskursgrammatik

Nach dem Beginn des Dialogs (*info_suche*) wird dem Lerner eine Frage präsentiert (*FRAGEN*), die sich aus der Wissensbasis zum Dialoginhalt (Abbildung 6.2) ergibt. Falls die Eingabe des Lerners als Antwort auf die Frage interpretiert werden konnte (interpretierbare Antwort), wird eine Bestäti-

[1] Vielfach werden die zu ergänzenden Wörter sogar in Listen vorgegeben, was dazu führen kann, dass der Lerner lediglich ausprobiert, welche Wörter in welcher Lücke vom Programm akzeptiert werden.

gung und, wenn es noch offene Frage gibt (offene Fragen), eine neue Frage ausgegeben. Für den Fall, dass die Eingabe nicht interpretierbar war, wird zunächst vom System nachgefragt (*NACHFRAGEN*), um dann entweder die Antwort zu bestätigen oder den Dialog abzubrechen (*DANKE_ENDE*). In der Implementierung wird tatsächlich zweimal nachgefragt, bevor der Dialog abgebrochen wird.

Anzumerken ist, dass die Dialoggrammatik im Rahmen des Systems gleichzeitig auch eine wesentliche Einschränkung der Analysefähigkeit zur Dialogform darstellt, da auf dieser Basis nur Frage-Antwort-Dialoge geführt werden können. Ähnliche Strukturen werden auch zur Analyse und Auswertung von natürlich-sprachlichen, das heißt Mensch-zu-Mensch-Dialogen verwendet, wobei der Umfang der Grammatik offensichtlich wesentlich größer sein muss. Als Beispiele seien Carletta et al. (1997, Strukturierung der so genannten Maptask-Dialoge) oder Alexandersson et al. (2000, Modellierung von Terminabsprachen) angeführt.

Eine einfacher zu implementierende Alternative wäre die Verknüpfung einer solchen Dialoggrammatik mit dem Inhalt eines Dialogs, wie sie zum Beispiel in dem System PROMISE (Bauer et al., 1994; John, 1994) realisiert wurde. Ein einziges Übergangsnetzwerk muss dazu allerdings nicht nur Informationen zu jeder Frage und der sich daran anschließenden möglichen Antwort haben, sondern jede Dialogbewegung muss außerdem vorausgesehen und im Netzwerk berücksichtigt werden. Wenn aber diese Strukturtypen wie in Promis*D* getrennt werden, muss die Abfolge von Fragen, Nachfragen und Antworten nur jeweils einmal in der Dialoggrammatik modelliert werden. Zusätzlich besteht durch die Modularisierung die Möglichkeit, gezielt Erweiterungen vorzunehmen: Beispielsweise muss „nur“ die Wissensbasis mit den Dialoginhalten ergänzt werden, wenn ein neuer Frage-Antwort-Dialog hinzugefügt werden soll. Auch der umgekehrte Fall ist denkbar. Wenn sich in Experimenten herausstellen sollte, dass zum Beispiel zweimaliges Nachfragen durch das System bei nicht interpretierbarer Eingabe für den Lerner nicht ausreichend ist, reicht eine Änderung der Dialoggrammatik aus, wohingegen in einem kombinierten Netzwerk an sehr vielen Punkten weitere Knoten hinzugefügt werden müssten, um diese Erweiterung zu realisieren.

Ein weiterer Aspekt der Verwendung einer solchen Dialoggrammatik ist die Nutzung als Richtlinie für das Führen von Dialogen. Falls der Lerner sich unerwartet anders als in der Dialoggrammatik beschrieben verhalten sollte, kann das Anlass zu einer plausiblen Fehlermeldung an den Lerner sein. Das heißt, eine Eingabe kann syntaktisch und thematisch dem Wissen des Systems entsprechend korrekt und doch keine in der Diskursgrammatik vorgesehene Reaktion sein. Ein Meldung an den Lerner beinhaltet dann die Nachricht, dass die Eingabe vermutlich keine geeignete Weiterführung des Dialogs darstellt.[2]

[2]Eine erste Ausarbeitung findet sich in Reuer (1999).

Technisch gesehen handelt es sich um ein einfaches Übergangsnetzwerk, in dem bestimmte Eingaben den Wechsel des Zustandes auslösen. Ob eine Eingabe einen Wechsel bewirkt, hängt von der Interpretation der Eingabe mit Hilfe der zweiten Wissensbasis ab. Wie erwähnt, wurden in dieser Wissensbasis drei Dialoge strukturiert: 1. eine allgemeine Vorstellung des Lerners, 2. eine Wegbeschreibung und 3. eine Unfallmeldung. Wie die folgende vereinfachte Abbildung 6.2 für den Unfallmeldungsdialog andeutet, ist die Wissensbasis hierarisch organisiert, das heißt allgemeine Informationen stehen oben an der Wurzel und speziellere weiter unten.

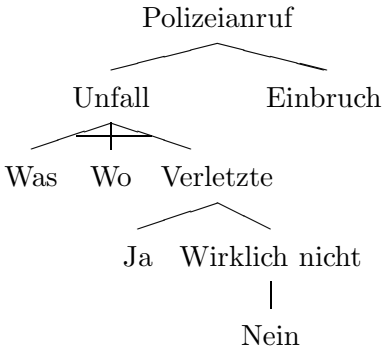


Abbildung 6.2: Vereinfachtes Dialog-Wissen (Unfallmeldung)

Die Schwesterknoten in einer Hierarchie können mit konjunktiven oder disjunktiven Verknüpfungen miteinander verbunden werden. Damit lassen sich zusätzliche Informationen über den möglichen Ablauf eines Dialogs in der Hierarchie speichern. In Abbildung 6.2 ist die konjunktive Verknüpfung durch die Verbindung der Äste zu *Was*, *Wo* und *Verletzte* dargestellt, was bedeutet, dass der Dialog zum Unfall nur erfolgreich beendet werden kann, wenn der Lerner Angaben zum Ablauf des Unfalls (Was), zum Ort (Wo) und zur Zahl der Verletzten gemacht hat. Dagegen ist die Angabe der Verletzten variabel gestaltet: Der Lerner kann angeben, dass es keine Verletzten gegeben hat; er kann aber auch angeben, dass es mehrere Verletzte zu beklagen gibt.

Wichtig bei der Interpretation einer solchen Hierarchie ist, dass die „Forderungen“ der Blätter eines Baumes im Laufe des Dialogs erfüllt werden müssen. Wenn also zu Beginn nur die Information gegeben wird, dass ein Unfall stattgefunden hat, wird vom System die Frage nach den genauen Umständen des Unfalls ausgegeben. Der Baum wird mit Hilfe der Dialoggrammatik solange abgeschritten, bis alle Blätter des Baumes besucht sind oder der Lerner mit nicht interpretierbaren Eingaben einen Abbruch des Dialogs erreicht.

Ein Beispielknoten dieses Baumes hat die folgende Form.

(6.1)
```
(1) dat(notzeuge,
(2)     [[‘Hat wirklich niemand den Unfall gesehen?’],np,
          hatwrklch_nmndunfllgshn],
(3)     [ [‘SEM’,[illokution,negativ]],
          [‘SEM’,[aktor,‘MENSCH’,[NEG,plus]],
            [gegenstand,‘KOLLISION’],[PRED,‘VISUELL’]],
          [‘SEM’,[gegenstand,‘ZEUGE’,[NEG,plus]],
            [PRED,‘EXIST’]],
          [‘KEYW,[‘niemand’]]],
(4)     [[‘Schade...’],schade],
(5)     [[‘Wer hat denn den Unfall gesehen?’],np,
          werhatunfllgshn],
(6)     [zeuge1,zeuge2,zeuge3,zeugewitz,notzeuge1],
(7)     ‘Niemand hat den Unfall gesehen.’).
```

Ein Knoten besteht aus 7 Elementen:

1. Eindeutiger Name des Knotens
2. Frage nach Informationen, mögliche POS der Lernereingabe außer S, Name der Sounddatei
3. Repräsentationen der zulässigen Antworten
4. Reaktion des Systems nach zulässiger Antwort, Name der Sounddatei
5. Nachfrage, mögliche POS der Lernereingabe außer S, Name der Sounddatei
6. Verweise auf Tochterknoten
7. Beispielsatz zur Beantwortung der Frage

Der Knoten (6.1) wird verwendet, wenn der Lerner die Frage nach Zeugen des Unfalls verneint. Da es wichtig ist, dass jemand den Unfall beschreiben kann, wird noch einmal mit der Frage aus Zeile (2) nachgehakt. Die Antwort darauf kann außer aus einem Satz auch aus einer NP bestehen (zum Beispiel „Niemand.“), was offensichtlich abhängig von der Form der Frage ist. Das letzte Element in dieser Zeile verweist auf eine Sounddatei, mit deren Hilfe die Frage auch gehört werden kann.

Zeile (3) unterstützt die Interpretation der Lernereingabe. Der erste Teil (`[illokution, negativ]`) kommt zur Anwendung, wenn der Lerner als Antwort auf die vorhergehende Frage einfach nur „Nein“ eingibt. Mit Hilfe des zweiten Teils werden Sätze wie zum Beispiel „Kein Mensch hat den Unfall gesehen“, „Niemand hat den Unfall gesehen“ oder „... sah den Unfall“ (`[aktor, ‘MENSCH’, [neg, ...]...]`) verarbeitet. Der dritte Teil bezieht sich auf eine Eingabe wie zum Beispiel „es gibt keinen Zeugen“ (`[gegenstand, ‘ZEUGE’, [neg, plus]], [pred, ‘EXIST’]`) beziehungsweise „keine Zeugen“. Es können also eine Reihe von Eingaben in verschiedenen Variationen erkannt und für die weitere Verarbeitung interpretiert werden. Die verschiedenen Einträge sind mit einer Markierung `‘SEM’` versehen, die offensichtlich nur für geparste Sätze Anwendung finden können. Die Angaben in einem solchen SEM-Element

entsprechen einem Auszug der Angaben aus einer F-Struktur. Zusätzlich lassen sich Stichworte eingeben (`'KEYW'`), mit denen die Eingabe verglichen wird, falls der Vergleich mit den semantischen Repräsentationen gescheitert ist, das heißt, falls der Parser keine Beschreibung generieren konnte.

Zeile (4) dient dazu, eine direkte Reaktion des Systems auszugeben. Dieses scheint mir wichtig, obwohl dadurch in manchen Fällen ein gewisser „künstlicher" Charakter entsteht. Mit der Ausgabe einer Reaktion bekommt der Lerner einen Hinweis darauf, ob das System die Eingabe in sinnvoller Weise verarbeiten konnte. Umgekehrt wird der Inhalt dieser Zeile nicht ausgegeben, falls die Eingabe nicht interpretiert werden konnte. Auch in dieser Zeile ist das letzte Element wieder ein Verweis auf eine Sounddatei.

Zeile (5) stellt eine Nachfrage dar, falls die Eingabe nicht interpretiert werden konnte. Als weitere Elemente sind hier wieder die Eingabe eines Satzes oder einer NP durch den Lerner und das Aufrufen einer Sounddatei möglich.

Zeile (6) verweist auf Knoten, die zur Interpretation der Folgefrage aus Zeile (2) herangezogen werden sollen. Falls diese Liste leer ist, ist das Blatt des Baumes erreicht und eine Frage aus einem weiteren Themenbereich kann folgen.

Die letzte Zeile (7) stellt die Hilfe für den Lerner dar. Dieser Satz wird ausgegeben, falls der Lerner wie in Kapitel 4.1 beschrieben den Hilfeknopf klickt.

Das Modul erlaubt es also, einen Aufgabentyp zu gestalten, der den didaktischen Anforderungen an Sprachlernprogramme zum großen Teil genügen kann. Obwohl die Reihenfolge der Präsentation der Fragen durch die Reihenfolge des Abschreitens der Hierarchie festgelegt ist, ergibt sich eine gewisse Flexibilität, die zum einen im hierarchischen Aufbau der Wissensbasis und zum anderen in den semantischen Repräsentationen, die für die Interpretation genutzt werden, begründet ist. Einerseits kann das Programm mit Informationen umgehen, die nach und nach präsentiert werden, sodass der Lerner entweder nur einen Unfall melden oder sofort den Ort des Unfall mitteilen kann. Zum anderen ergibt sich die Möglichkeit, mit Hilfe der aus der F-Struktur extrahierten semantischen Repräsentationen diverse Eingabesätze abzudecken, bei der die Annahme gerechtfertigt sein kann, dass sie im Wesentlichen dasselbe bedeuten.

Bei der Evaluation mit den Studierenden hat es gelegentlich auch Versuche gegeben, das System auszureizen, um dann beobachten zu können, wie das Programm reagiert. Bei einem solchen Dialogmodul, bei dem es sich offensichtlich, wie in der Einleitung erwähnt, nicht um die Grundlage einer „Konversations-Übungsmaschine" handelt, kann immer eine aus Lernersicht akzeptable Eingabe konstruiert werden, die vom Programm nicht verarbeitet werden kann. Aus diesem Grund muss sich das System konsequent zeigen und den Dialog abbrechen, wenn eine Eingabe nicht interpretiert werden kann. Allerdings muss der Lerner in solch einem Fall außerdem darüber in-

formiert werden, dass seine Eingabe möglicherweise sinnvoll war, aber vom System wegen der Beschränktheit der Modelle nicht im aktuellen Kontext für eine Fortsetzung des Dialogs nutzbar war. Für die Erweiterung der Dialogfähigkeiten von Promis*D* wären neben dem Ausbau der Wissensbasen vor allem auch Module notwendig, die die Eingaben des Lerners in Rahmen eines so genannten „mixed-initiative"-Dialog verarbeiten könnten. Neben den erweiterten Möglichkeiten, Wissen aus den Wissensbasen zu inferieren, wäre vermutlich zusätzlich eine Sprachgenerierungs-Komponente erforderlich, die flexiblere Reaktionen des simulierten Dialogpartners erlaubt.

6.2 Orthografie- und Semantikkontrolle

Im zweiten Teil dieses Kapitels sollen zwei weitere Module kurz erläutert werden, da auch von diesen Modulen Fehlermeldungen an den Lerner ausgegeben werden können. Angemerkt sei noch einmal, dass sowohl die Orthografie als auch die Semantik Bereiche sind, die nicht im Mittelpunkt dieser Arbeit standen.

6.2.1 Orthografie

Obwohl die Orthografie als etwas Nebensächliches erscheinen mag und im allgemeinen Kontext von computerlinguistischen Anwendungen häufig davon ausgegangen wird, dass die Wörter einer Eingabe korrekt sind, muss offensichtlich in einem ICALL-System mit orthografisch fehlerhaften Eingaben gerechnet werden.[3] Allerdings ist anzumerken, dass insbesondere im Bereich der Erkennung von gesprochener Sprache die Identifikation einzelner Wörter ein bekanntes Problem ist. Häufig wird zur Lösung dieses Problems mit so genannten Worthypothesengrafen gearbeitet, mit deren Hilfe für bestimmte Unterabschnitte der Eingabe Worthypothesen in strukturierter Form aufgebaut werden können. Wie zu Beginn des Kapitels 5 angemerkt, geht es im Rahmen eines Sprachlernprogramms aber nicht um robustes Parsing, sondern um die möglichst präzise Identifikation von Fehlern, die dem Lerner auch mitgeteilt werden sollten.

Da also einerseits das Modul zur Behandlung von morphosyntaktischen Fehlern in Promis*D* eine orthografisch fehlerfreie Eingabe benötigt und andererseits dem Lerner auch morphosyntaktische Fehler mitgeteilt werden sollten, also sensitives Parsing erforderlich ist, muss ein Modul im Programm integriert sein, das den Lerner bei der Produktion orthografisch korrekter Eingaben unterstützt. Ein Konzept, das dem hier vorgestellten sehr ähnlich ist, aber den Schwerpunkt auf die Behandlung der stark agglutinierenden Sprache Türkisch legt, wird in Oflazer (1996) vorgestellt.

[3] Zur Präsentation der Orthografiekontrolle und der Korrekturvorschläge auf der Programmoberfläche siehe Abschnitt 4.2.

Um die Orthografiekontrolle in Promis$D$ zu implementieren, wurden zunächst Vollform-Wortlisten aus der CELEX-Datenbank zusammengestellt.[4] Aus diesen ca. 130.000 Einträgen wurden dann zwei Buchstabenbäume generiert, um eine kompakte und effizient zu nutzende Darstellung der Informationen zu erreichen. In einem Baum sind die Wörter in korrekter Schreibweise enthalten und in dem anderen sind die Wörter rückwärts geschrieben kodiert. Die folgende Abbildung deutet einen Ausschnitt aus einem solchen Baum an.

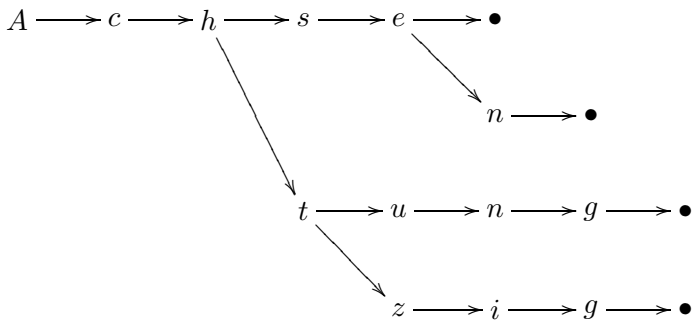


Abbildung 6.3: Buchstabenbaum

In dieser Struktur befinden sich die Wörter *Achse*, *Achsen*, *Achtung* und *Achtzig*. Wenn nun beispielsweise die Eingabe *Achen* analysiert werden soll, können mit Hilfe dieses Baumes und der Berechnung der so genannten Levenshtein-Distanz die Wörter ermittelt werden, die eine bestimmte Ähnlichkeit mit dem Wort aus der Lernereingabe haben. Dazu werden die Wörter ermittelt, die sich „kaum“ vom Wort aus der Lernereingabe unterscheiden. Als mögliche Fehler beziehungsweise als notwendige Editieroperationen zur Korrektur werden

- ein Buchstabe zuviel,
- ein Buchstabe zu wenig sowie
- ein Buchstabe verändert

angenommen. Wörter, die einen größeren Abstand aufweisen, werden nicht berücksichtigt, das heißt, für diese Wörter wird kein Korrekturvorschlag ausgegeben. Implementiert ist die Analyse derart, dass zunächst die Wörter aus dem Baum extrahiert werden, die den gleichen Anfang haben. Damit kann auf ein relativ aufwendiges Protokollieren während der Suche verzichtet werden. Die gleiche Analyse wird nun mit Hilfe eines Buchstabenbaumes durchgeführt, der die Wörter rückwärts geschrieben enthält. Den so ermittelten Wörtern kann ein spezifischer Abstandswert entsprechend der Anzahl der Editieroperationen zugewiesen werden. Die ähnlichsten Wörter werden

[4] Diese Datenbank des Max-Planck-Instituts für Psycholinguistik in Nijmegen (NL) ist zur Zeit nur über das Linguistic Data Consortium zu beziehen.

dann an den Lerner als Korrekturvorschläge ausgegeben. Obwohl die Annahme einer Editierdistanz von 1 als sehr gering für Lernereingaben in einem ICALL-System erscheinen mag, hat die praktische Erprobung gezeigt, dass dieser Wert in sehr vielen Fällen ausreichend ist.

Um die Ergebnismenge noch weiter einzuschränken, müsste notwendigerweise auf den Satzkontext zurückgegriffen werden. Es bietet sich beispielsweise an, mit Hilfe eines Taggers auf der Basis eines relativ kleinen Kontextes die Wortart eines spezifischen Wortes weiter einzuschränken, was aber in Promis*D* nicht implementiert worden ist. Meines Erachtens sollte es dann möglich sein, den Korrekturvorschlag auf einige wenige Elemente einzuschränken.

6.2.2 Semantik

Der letzte hier vorzustellende Bereich, in dem bei der Bearbeitung einer Dialogübung Fehler erkannt werden können, ist der Bereich der Selektionsrestriktionen. Zunächst soll kurz auf die Funktionalität dieses Moduls eingegangen werden, um zu zeigen, auf welche Weise ein Verstoß gegen (grobe) Selektionsrestriktionen erkannt werden kann, und im Anschluss werden dann einige prinzipielle Schwierigkeiten betrachtet.[5]

Der wesentliche Teil des Moduls besteht aus einer Wissensbasis, in der die semantischen Repräsentationen eines Teils des Lexikons hierarchisch strukturiert sind, um Weltwissen zu modellieren. Um eine Verbindung vom Lexikon zu dieser Wissensbasis zu schaffen, finden sich die Werte der PRED-Attribute der einzelnen Lexikoneinträge (siehe Beispiel (3.13), Seite 68) hier als Knoten in der Hierarchie wieder. Das so genannte Toplevel dieser Wissensbasis bildet dazu eine Ausnahme und besteht zum Teil aus Konzepten, die nicht als Lexeme auftreten. Diese oberste Ebene der Hierarchie unterscheidet einerseits Konzeptuelles in Abgrenzung zu wahrnehmbaren Dingen und andererseits auch sicht- und hörbare Aktionen voneinander.

Die Einträge zu Aktionen, das heißt im Normalfall Verben, beinhalten insbesondere die thematischen Rollen aus dem Lexikon und ihre möglichen Belegungen. Nach einem Parse werden diese Belegungen aus der F-Struktur extrahiert und mit den in der semantischen Wissensbasis festgelegten Möglichkeiten verglichen. Das folgende Beispiel 6.2 soll das verdeutlichen.

(6.2) Satz: Der Mann lacht.

[5] Die hier realisierte Form der Semantikkontrolle basiert in Teilen auf den Konzepten des Studienprojekts PROMISE (Bauer et al., 1994; John, 1994) (siehe Seite 5).

F-Struktur:
$$\begin{bmatrix} \text{PRED} & = \text{`HEITERKEIT} < \uparrow \text{SUBJ}> \text{'} \\ \text{TENSE} & = \text{pres} \\ \text{SUBJ} & = \begin{bmatrix} \text{PRED} & = \text{`MANN'} \\ \text{FORM} & = \text{agent} \\ \text{CASE} & = \text{nom} \\ \text{PERS} & = 3 \\ \text{NUM} & = \text{sg} \end{bmatrix} \end{bmatrix}$$

Sem-Struktur: [[PRED, 'HEITERKEIT'], [agent, 'MANN']]

WB-Eintrag:
```
klasse(HEITERKEIT,[GEMUETSZUSTAND],
       [agent,MENSCH]).
klasse(MANN,[MENSCH],[]).
```

Aus der Analyse des Satzes durch den Parser ergibt sich die semantische Struktur [[PRED, 'HEITERKEIT'], [agent, 'MANN']] (siehe auch Beispiel (6.1)). Diese Struktur wird nun mit dem Eintrag aus der Wissensbasis (WB-Eintrag) verglichen. In diesem Fall enthält der WB-Eintrag die Informationen, dass das Konzept `HEITERKEIT` ein Subkonzept von `GEMUETSZUSTAND` ist und dass die thematische Rolle `agent` mit dem Konzept `MENSCH` gefüllt sein muss. Dieses ist der Fall, da `MANN` an anderer Stelle in der Wissensbasis als Subkonzept von `MENSCH` verzeichnet ist (im Beispiel als zweiter WB-Eintrag).

Üblicherweise wird in computerlinguistischen Anwendungen eine semantische Wissensbasis in dieser Form dazu genutzt, die Eingaben des Nutzers semantisch zu interpretieren beziehungsweise Inferenzen daraus zu ziehen. In einem Sprachlernsystem muss eine solche Hierarchie allerdings wie erwähnt eine zusätzliche Aufgabe erfüllen, nämlich eine Plausibilitätskontrolle unterstützen. Vor allem aus der Nutzung zur Analyse von semantischen Fehlern ergeben sich dabei entscheidende Schwierigkeiten. Aus verschiedenen Gründen kann eine solche Wissensbasis nur unvollständig bleiben. Ein wesentlicher Grund ist, dass nicht alle Möglichkeiten der semantisch akzeptablen Formen kodiert werden können, da gerade beim Gebrauch der Sprache ein kreatives Moment eingesetzt werden kann, was dieses verhindert. Hinzu kommt das klassische Problem der so genannten „closed world assumption“. In einem System wie Promis*D* muss wie schon erwähnt davon ausgegangen werden, dass der Lerner Eingaben machen wird, die nicht analysiert werden können beziehungsweise die nicht in einer Wissensbasis kodiert worden sind.

Als Konsequenz aus diesen Problemen wurde hier die Entscheidung getroffen, „schwache“ Selektionsrestriktionen nicht zu modellieren. Verben, die in dieser Hinsicht nicht oder kaum restringiert sind, sind daher nicht in der Wissensbasis enthalten. Falls also vom System ein semantischer Fehler des Lerners diagnostiziert worden ist, kann mit großer Wahrscheinlichkeit davon ausgegangen werden, dass die Rückmeldung über den Fehler korrekt ist. In welcher Form sich eine Fehlermeldung dem Lerner präsentiert, wird in Abschnitt 4.2.3 dargestellt.

In diesem Kapitel wurde gezeigt, dass mit Hilfe einer einfachen, aber flexiblen Architektur die Strukturierung von Frage-Antwort-Dialogen ermöglicht wird, die im Gegensatz zu klassischen Sprachlernprogrammen einigen didaktischen Anforderungen genügen kann. Einerseits wird eine Flexibiltät erreicht, die es dem Lerner erlaubt, in einen größeren Kontext aktiv seine Sprachkenntnisse einzusetzen und zu trainieren. Andererseits ist dieser Übungstyp wiederum so beschränkt, dass sinnvolle Rückmeldungen über das Verhalten des Lerners im Dialog gegeben werden können. Schließlich wurden zwei weitere Module in Promis*D* integriert, wobei die Orthografiekontrolle in erster Linie eine notwendige Anforderung für die morphosyntaktische Analyse darstellt und die Semantikkontrolle den Versuch zeigt, wie dem Lerner auch für diesen Bereich Feedback auf fehlerhafte Eingaben gegeben werden kann.

Kapitel 7

Schlusswort

In diesem Kapitel soll zunächst ein Fazit gezogen werden, das zusammenfassend zeigt, inwieweit die zu Beginn dieser Arbeit erläuterten Ziele erreicht worden sind. Zum Abschluss folgt dann ein Ausblick, in dem sowohl ein möglicher praktischer Einsatz von Promis$D$ und dessen Konsequenzen als auch eine mögliche Ausdehnung der CL-basierten Fehlererkennung beziehungsweise der linguistischen Daten betrachtet werden.

7.1 Fazit

Die Entwicklung eines Verfahrens zur antizipationsfreien Erkennung und Erklärung von grammatischen Fehlern war das primäre Ziel dieser Arbeit. Drei wesentliche Aspekte wurden dabei berücksichtigt: 1. Grammatiktheorie: Die Theorie der LFG eignet sich aus verschiedenen Gründen für einen Einsatz in einem ICALL-System. 2. Übungstyp und Fehlermeldungen: Der simulierte Dialog als Übungstyp erweitert die Möglichkeiten des Computereinsatzes für das Sprachenlernen nach didaktischen Grundsätzen, der insbesondere auch präzise Fehlermeldungen umfasst. 3. Antizipation und Effizienz: Das antizipationsfreie Parsingverfahren erlaubt die Identifizierung einer Vielzahl von Fehlern im Rahmen einer präzise definierbaren und skalierbaren Erweiterung der angewandten Methoden Unifikation und Earley-basiertes PS-Parsing.

1. Grammatiktheorie

Es wurde eine Grammatiktheorie integriert, die insbesondere aus den folgenden drei Gründen für den Einsatz in einem ICALL-System geeignet ist.

- Die Theorie ist formal fundiert und damit vom Grundsatz her in Kombination mit einem Parser einsetzbar. Dieser Aspekt drückt sich auch in zahlreichen bereits existierenden Implementationen aus, die zum Teil umfassende Grammatiken beinhalten.

- Die Theorie hat sich zur Beschreibung von zahlreichen, typologisch unterschiedlichen Sprachen als geeignet erwiesen, was sie aus einem bestimmten Blickwinkel zu einer linguistisch adäquaten Theorie macht. Angemerkt werden kann, dass die Grammatikmodellierung bereits einen wesentlichen Teil der erwartbaren Äußerungen von Lernern im Rahmen der Aufgabenstellung umfasst und mit Hilfe eines größeren Lexikons kaum mehr uninterpretierbare Eingaben auftreten sollten.
- Im Vergleich mit anderen gängigen Grammatiktheorien hat sich die LFG als relativ ähnlich zu pädagogischen Grammatiken gezeigt, die im Sprachunterricht eingesetzt werden, sodass sich Rückmeldungen, die für einen Lerner verständlich sein sollten, für alle morphosyntaktisch relevanten Bereiche ohne große Transformationen aus der formalen Beschreibung, vor allem der F-Struktur, generieren lassen. Darüberhinaus können möglicherweise auch vollständige LFG-artige Beschreibungen dem Lerner direkt präsentiert werden, um wie in einer Lernergrammatik morphosyntaktische Strukturen zu verdeutlichen.

2. Übungstyp und Fehlermeldungen

Wie in Kapitel 2 ausgeführt, sehen Didaktiker neben den vor allem methodisch-didaktisch begrenzten Übungstypen die völlig unzureichenden Rückmeldungen kommerzieller Programme auf Lernereingaben als einen der wesentlichen verbesserungswürdigen Bereiche an. Obwohl manche Programme mit Hilfe der Antizipation von „bis zu zehn richtigen Antworten“ (Rüschoff und Wolff, 1999, S. 69) schon auf eine gewisse Bandbreite an erwartbaren Eingaben reagieren können, bleibt diese direkte Antizipation ein prinzipielles Problem.

Im Gegensatz dazu kann das hier vorgestellte System in begrenztem Umfang auf frei formulierte Sätze reagieren und stellt mit dem Übungstyp des simulierten Dialogs eine entscheidende Erweiterung für den Bereich der tutoriell orientierten Anwendungen dar. Außerdem ist der in Promis*D* integrierte Parser in der Lage, auf der Basis der LFG insbesondere für Kongruenz- und Rektionsfehler präzise und umfangreiche Rückmeldungen zu liefern, wobei die Art und Weise der Rückmeldung an den kommunikativen Übungstyp angepasst ist. Die Meldungen bestehen für Fehlermarkierungen in der F-Struktur aus einer Angabe der beteiligten funktionalen Argumente, der sich widersprechenden Werte und deren Feature sowie der möglichen Korrekturen der betroffenen Wörter. Damit stehen dem Lerner sämtliche morphosyntaktischen Informationen der Grammatik und des Lexikons zur Verfügung. Bei Konstituentenfehlern sind diese Informationen eingeschränkt, da in dem hier realisierten Verfahren nur eine Korrektur, aber eine Erklärung wie erwähnt nicht von der Beschreibung geliefert werden kann. Die Erklärung eines Fehlers lässt sich meines Erachtens nur mit Hilfe einer umfassenden Datenbank realisieren, die komplexe Regeln enthält, wie beispielsweise die folgende:

(7.1) WENN Aussagesatz UND Non-Subjekt am Satzanfang,
DANN Subjekt direkt nach dem finiten Verb

Aus diesen Regeln, die zum Teil in der K-Struktur und zum Teil in der F-Struktur überprüft werden müssen, können dann, allerdings mit erheblichem Aufwand, sinnvolle Erklärungen für den Lerner generiert werden. Die Generierung einer Erklärung lediglich auf der Grundlage der im Ableitungsbaum auftretenden Kategorien scheint mir dagegen ungleich schwieriger realisierbar zu sein.

Die Implementation von erweiterten Präferenzstrategien für bestimmte Analysen, die über einfache numerische Fehlerwerte hinausgehen, scheitert nach den hier gezeigten Möglichkeiten an dem Konzept für eine kompaktere Lexikonrepräsentation und an sich vermutlich widersprechenden Präferenzregeln. Desweiteren lassen sich verschiedene Dimensionen identifizieren, in denen die Rückmeldungen an den Wissensstand des Lerners adaptiert werden können. Je nach Vorkenntnissen und Aufgabenstellung lassen sich die zu präsentierenden Angaben zumindest für sich widersprechende Merkmalsbelegungen individuell steuern.

3. Antizipation und Effizienz

Zur Identifikation von Fehlern wurden den Strukturelementen der LFG entsprechend zwei Verfahren entwickelt, die im Prinzip ohne die Antizipation von Fehlertypen und -positionen in den linguistischen Wissensbasen die Beschreibung für fehlerhafte Eingaben liefern können. Damit ist es möglich, eine Grammatik und ein Lexikon ohne Bezug zur Sprachlernsituation zu entwickeln und zur fehlersensitiven Analyse von nahezu freien Lernereingaben einzusetzen. Die Erkennung von Konstituentenfehlern wurde zunächst auf die Erkennung von Auslassungs-, Einfügungs- und Verschiebungsfehlern beschränkt, für die im Gegensatz zum ähnlichen Ansatz von Lee et al. (1995) nur eine Hypothese generiert wird, wenn es dazu Hinweise in der Chart gibt. Für Kongruenz- und Rektionsfehler, die in der F-Struktur einer LFG-artigen Analyse auftreten, wurde ein Verfahren entwickelt, das die sich widersprechenden Werte mit Hilfe einer Erweiterung der Subsumption beziehungsweise der Unifikation in der F-Struktur speichert.

Um eine Möglichkeit zur korpusbasierten Optimierung der Verfahren zu erhalten, wurde allerdings sowohl für Fehler in der F-Struktur als auch in der K-Struktur eine Beschränkung der „zulässigen“ Fehlertypen integriert.[1] Menzel (1992, S. 225) weist darauf hin, dass für die Realisierung eines praktischen Systems „entscheidend [ist], wie man eine Kopplung antizipationsfreier Techniken mit antizipationsbasierten Ansätzen so organisiert, daß eine Symbiose der spezifischen Vorteile beider Ansätze wirksam werden kann.“

[1] Damit kann als Nebeneffekt ein gezieltes Debugging bei der Entwicklung der Grammatik zum Beispiel durch Relaxierung einzelner Attribute vorgenommen werden.

Wesentlich ist in Promis*D*, dass Features, die zum Beispiel lediglich der Optimierung der Analysen dienen und daher für den Lerner irrelevant sein sollten, selektiv als Fehlerposition ausgeschlossen werden können, sodass auf diese Weise unverständliche und unsinnige Fehlermeldungen verhindert werden können. Das Gleiche gilt beispielsweise auch für die Einfügung von Lexemen zur Korrektur eines Auslassungsfehlers: Mit einer Beschränkung bei der Einfügung auf die POS, die sich bei der Korpusanalyse häufig als Korrekturkandidaten für einen Auslassungsfehler gezeigt haben, wird die Generierung von unsinnigen Fehlerhypothesen vermieden.

Obwohl der Rechenaufwand zur Bestimmung einer minimalen Lösung nicht unerheblich ist, wurde gezeigt, wie eine generelle antizipationsfreie Methode zur Analyse von fehlerhaften Eingaben in Kombination mit eindeutigen Einschränkungen auf häufige Fehlertypen in ein intelligentes Sprachlernsystem integriert werden kann. In der Evaluation wurde insbesondere dargelegt, dass einerseits Verschiebungsfehler vermutlich aufgrund der Links-Rechts-Abarbeitung mit Hilfe von PS-Regeln schlecht identifiziert werden konnten, wohingegen andererseits sowohl Einfügungs- und Auslassungsfehler als auch Fehler, die in der F-Struktur kodiert werden müssen, mit sehr guter Präzision erkannt werden.

7.2 Ausblick

Obwohl schon einige Konzepte zur Diagnose von morphosyntaktischen Fehlern entwickelt worden sind, wurde der Bereich der „intelligenten“ Fehlererkennung bisher mit sehr wenigen Ausnahmen nicht in kommerzielle Systeme integriert. Ein wesentliches Hindernis stellt sicher die aufwendige Entwicklung eines solchen Systems mit schmalen Gewinnmargen pro verkauftem Programm dar. Eine zweites Problem sind vermutlich auch die zum Teil völlig überhöhten Erwartungen von Sprachlehrern und -lernern an Programme, die computerlinguistische Verfahren zur Analyse einsetzen. Atwell (1999) wie auch Nerbonne (2002) deuten darauf hin, dass hier wohl auch ein Akzeptanzproblem liegt. Atwell zufolge ergaben Nutzerstudien im ISLE-Projekt (Menzel et al., 2001), dass Sprachlehrer sich eine „Konversations-Übungs-Maschine“ wünschten, mit der die Lerner Konversation betreiben könnten, um so die notwendigen Techniken zu üben (Atwell, 1999, S. 34f.). Ein weiteres Merkmal der Maschine sollte die Fähigkeit sein, die Fehler der Lerner zu erkennen, zu korrigieren und zu erläutern. Obwohl die Realisierung dieser Vorstellung vermutlich noch in einiger Ferne liegt, zeigen einige Projekte, wie die in Heift (2001) oder deSmedt (1995) vorgestellten, sowie mit Einschränkungen auch das hier präsentierte System, dass die Anwendung von computerlinguistischen Methoden zur Fehlererkennung zumindest für ausgewählte sprachliche Phänomene in der Praxis erfolgreich sein kann.

Als Ausblick werden im Folgenden einige Aspekte des hier entwickel-

ten Ansatzes diskutiert, die für einen praktischen Einsatz des Programms berücksichtigt werden müssten. Erörtert werden sollen dabei vor allem die Konsequenzen aus der offenen Gestaltung des Übungstyps in Promis*D* im Gegensatz zu eher engen Übungsformaten wie in Heift (2001), die zwar auch von einem Parser zur Fehleranalyse unterstützt werden, aber ansonsten herkömmlichen Programmen entsprechen. Zum Abschluss der Arbeit wird dann beleuchtet, welche Aspekte bei der Erweiterung einerseits der Fehlererkennung und andererseits der Grammatik beziehungsweise des Lexikons berücksichtigt werden müssen.

7.2.1 Aspekte eines praktischen Einsatzes

Im Vordergrund soll zunächst die Frage stehen, welche Auswirkungen die mit Hilfe der Dialogsimulation gewonnene Flexibilität auf das Lernerverhalten haben kann. Da die Bearbeitung eines Dialogs unterschiedliche Abfolgen und wechselnde Antworten zulässt, können unterschiedliche Dialogstrukturen verfolgt werden, die sich aus den Eingaben eines Lerners ergeben. Diese Variabilität kann einen Lerner reizen, denselben Dialog noch einmal in anderer Form zu bearbeiten, während bei herkömmlichen Systemen mit Multiple-Choice- oder Einsetzaufgaben die Lösungen üblicherweiser auf eine einzige mögliche Antwort beschränkt sind. Eine Wiederholung ist in diesem Fall nur von sehr begrenztem Wert, sodass dieser Missstand in klassischen Programmen üblicherweise mit einer sehr großen Anzahl von Übungen kompensiert wird. Die gleichen Anmerkungen treffen auch auf die angeführten parserbasierten Systeme zu, die zwar das Feedback mit Hilfe der tiefen Analyse verbessern, ansonsten aber die gleichen Übungstypen liefern wie herkömmliche Programme.

Im Gegensatz dazu hat sich bei der Sammlung der Daten für das erwähnte Evaluationskorpus gezeigt, dass die unterschiedlichen Lerner diverse Strategien zur Bearbeitung der Dialoge verfolgt haben, die auch vielfach zum erfolgreichen Abschluss geführt haben. Aus dieser Beobachtung kann geschlossen werden, dass es durchaus für einen Lerner interessant sein kann, einen Dialog, wie er in Promis*D* konzipiert wurde, mehrfach mit unterschiedlichen Eingaben beziehungsweise Aussagen, die offensichtlich im Rahmen der Aufgabestellung bleiben müssen, zu bearbeiten. Die durch die Konzeption des Systems gewonnene Flexibilität kann also für den Lerner eine interessante Erweiterung der Aufgabenstellung darstellen. Ohne ausführliche Untersuchungen zum Lernerverhalten bei der Nutzung des Programms können diese Annahmen allerdings nur vorläufig gelten. Welche erstrebenswerten Konsequenzen sich aus der Aufgabenstellung in didaktischer Hinsicht ergeben, habe ich bereits in Kapitel 4 erläutert.

Ein weiterer Aspekt, der angesprochen werden soll, sind die Anforderungen an das System aufgrund der Aufgabenstellung, die relativ offen gehalten ist, um dem Lerner eine selbstformulierte Eingabe zu ermöglichen.

Hierbei muss eine Abwägung stattfinden zwischen einer offenen Aufgabenstellung, die beliebige Eingaben zulässt und damit Gefahr läuft, auch einige „False-Positives“ zu erzeugen sowie einer strikteren Aufgabenstellung, die eine starke Begrenzung der syntaktischen Formen nahelegt und daher genauere Analysen liefern können sollte. Gegenüber einigen anderen Systemen wurde hier der erste Ansatz verfolgt, um den didaktischen Empfehlungen zur sinnvollen Unterstützung des Lernprozesses zu folgen. Einerseits wird also dem Lerner die Möglichkeit geboten, sein aktives Wissen über die Sprache einzusetzen und andererseits ist das System in der Lage, zu den wesentlichen Fehlertypen im morphosyntaktischen Bereich präzise Rückmeldungen zu liefern.

Eine Einschränkung der Aufgabenstellung würde zwar die Chance erhöhen, dass sowohl alle zu erwartenden syntaktischen Strukturen als auch Fehler präziser identifiziert werden können, da die Grammatik und das Lexikon eine wesentlich geringere Komplexität aufweisen können. Allerdings wäre damit das Programm wieder auf den Status eines „drill“-Programms herabgesetzt, das wie in Kapitel 2 geschildert, didaktisch-methodische Nachteile mit sich bringt.[2]

Eine vorläufige Analyse des HU/FU-Korpus hat außerdem ergeben, dass Lerner in dem hier entwickelten Übungstyp des simulierten Dialogs am Rechner nur begrenzte syntaktische Strukturen aktiv verwenden, die sicherlich auch von den vom System präsentierten Fragen abhängen. Aus der Tatsache, dass die wesentlichen Anpassungen für die Evaluation im Bereich des Lexikons und nicht im Bereich der Grammatik vorgenommen werden mussten, kann meiner Ansicht nach geschlossen werden, dass die Grammatik nach einer beschränkten Entwicklungszeit in der Lage sein sollte, für nahezu alle erwartbaren Eingaben eine Beschreibung zu liefern. Die Entwicklung einer offenen Aufgabenstellung erscheint aus diesen Gründen sinnvoll und möglich.

Im Anschluss an diese Ausführungen zu einem möglichen praktischen Einsatz des Systems und einigen daraus folgenden Konsequenzen, sollen nun zwei Aspekte in Bezug auf die Fehlererkennung diskutiert werden, die meiner Ansicht nach für eine Erweiterung sowohl der Grammatik als auch des Diagnoseumfangs relevant sind.

[2] Ein zusätzlicher Bereich, der bei einem praktischen Einsatz des Systems zum Sprachenlernen offensichtlich bedacht werden muss, ist die Komplexität der Aufgabengenerierung. Wie in Abschnitt 4.4 gezeigt, sind mindestens sechs verschiedene Wissensbasen für die Gestaltung der Dialoge und der Eingabeanalyse notwendig. Im gravierendsten Fall sind bei der Gestaltung eines neuen Dialogs Erweiterungen in allen Wissensbasen notwendig, wobei vermutlich insbesondere der zu erwartende Wortschatz einen wesentlichen Teil ausmacht. Das wichtigste Element besteht aber in der Wissensbasis mit den Dialoginhalten selber, die mit erheblichen Aufwand vollständig neu aufgebaut werden muss. Im Gegensatz dazu besteht ein wesentlicher Vorteil der restriktiven Programmkonzepte mit einfachen Aufgabentypen in der relativ einfachen Generierung der Aufgaben und dem Einsatz weniger Ressourcen.

7.2.2 Diagnosefähigkeit vs. Diagnoseaufwand

Aus der Tatsache, dass die Grammatik in Promis*D* speziell für die Evaluation entwickelt worden ist und dass das Verfahren zur Erkennung von Konstituentenfehlern auf Annahmen über mögliche Chartzustände basiert, stellt sich möglicherweise im Anschluss an die Überlegungen zum praktischen Einsatz die Frage, ob das Fehlererkennungsverfahren auch bei einer größeren Erweiterung oder bei der Verwendung einer anderen Grundstruktur der PS-Regeln erfolgreich funktioniert. Nach einigen Anmerkungen zu diesem Punkt wird am Schluss dieses Abschnitts noch einmal der Aspekt der korpusbasierten Entwicklung des Analyseverfahrens und den daraus resultierenden Überlegungen zur Anpassung an die belegten Fehler betrachtet.

Zunächst geht es also um die Frage, ob sich die Leistung des hier präsentierten Verfahrens wesentlich ändert, wenn beispielsweise eine umfassendere und komplexere Grammatik integriert wird, da sich das Verfahren zur Erkennung von Konstituentenfehlern ganz entscheidend auf die von der Grammatik gelieferten Chart-Konfigurationen stützt. Als ein problematisches Beispiel seien hier Postpositionen genannt, die ich in der in Promis*D* eingesetzten Grammatik versuchsweise modelliert habe. Wenn die Grammatik eine Regel zur Beschreibung von Postpositionen (siehe Abschnitt 3.4.3) enthält, werden grundsätzlich alle NPn auch als mögliches Argument einer Postposition in die Chart eingetragen. Wenn aber eine Argument-NP zum Verb wie im folgenden, konstruierten Beispiel falsch positioniert ist (siehe auch die Beispiele in (5.3) auf Seite 118), wird damit die Erkennung des Fehlers wesentlich erschwert, da erst nach der NP *dich* erkannt werden kann, dass keine Postposition vorhanden ist, sondern dass es sich um ein falsch positioniertes Verb-Argument handelt.

(7.2) *Der Mann dich sieht.

Neben einer Erweiterung der Trigger-Bedingungen, die den Korrekturprozess auslösen, ist für derartige Fälle eine Erweiterung und möglicherweise auch ein Änderung der Korrekturstrategien notwendig, die aber nicht untersucht worden ist.

Versuchsweise habe ich in einem experimentellen Rahmen eine kleine Grammatik auf der Basis des topologischen Modells für die deutsche Wortstellung in Anlehnung an Clément et al. (2002) entwickelt und auch einfache Sätze mit exemplarischen Konstituentenfehlern konstruiert, um den Korrekturmechanismus an einem anderen Grammatiktyp untersuchen zu können. Für einfache, fehlerhafte Sätze wie beispielsweise „Mann lacht.“ oder „Ich habe gesehen dich.“ ergaben sich in fast allen Fällen die erwarteten Analysen, obwohl eben eine völlig andere Grammatikstruktur gewählt wurde. Diese Tatsache deutet daraufhin, dass die hier entwickelten Korrekturmechanismen für Konstituentenfehler auch mit strukturell anderen Grammatikregeln sinnvolle Fehleranalysen liefern können, sodass hier von einem universel-

lerem Ansatz ausgegangen werden kann, als es vielleicht ursprünglich den Anschein hatte.

Neben der Öffnung der „nicht-linguistischen“ Beschränkungen für die Erkennung von Konstituentenfehlern besteht natürlich auch die Möglichkeit, die linguistischen Beschränkungen zu öffnen, um möglicherweise weitere Fehlertypen identifizieren zu können. Wie in Kapitel 5 ausgeführt, lassen sich für die Erkennung von Fehlern sowohl in der F-Strutur als auch in der K-Struktur leicht Restriktionen über die möglichen Fehlerhypothesen festlegen, die, wenn sie als „harte Constraints“ realisiert sind, den Suchraum in ganz erheblichen Maß beschränken können. Allerdings wird damit auch die Erkennung von Fehlern in den somit ausgeschlossenen Bereichen verhindert, sodass ein Abwägen zwischen der Diagnosefähigkeit und dem Laufzeitverhalten des Systems notwendig ist. Als Beispiel sei hier noch einmal die Zulassung eines Fehlers an einem bestimmten Feature und dessen Einfluss auf den Analyseaufwand erwähnt (Abschnitt 5.4.1, Seite 149). Im Gegensatz dazu gibt es offensichtlich auch Features, die lediglich Informationen für eine spätere Interpretation liefern und daher gar nicht erst für einen möglichen Fehlerfall berücksichtigt werden. Es muss also bei der Entscheidung, bestimmte Features als nicht relaxierbar zu kategorisieren, in jedem Einzelfall geprüft werden, ob das jeweilige Feature zur Erklärung eines häufigen Lernerfehlers beitragen kann.

Auch Konstituentenfehler sind konstruierbar, die im vorliegenden Korpus gar nicht auftreten und daher von der Fehlererkennung zunächst ausgeschlossen werden können. Ein besonderer Aspekt in diesem Bereich ist meiner Meinung nach, dass durch Konstituentenfehler zum Teil deutliche Interpretationsschwierigkeiten entstehen, die sich bei Kongruenz- oder Rektionsfehlern in dieser Form nicht ergeben können. Vermutlich aus diesem Grund kommt ein Satz mit einer sehr stark veränderten Wortstellung, bei dem die Bedeutung nicht oder nur kaum zu erschließen ist, also eher selten vor. Diese Einsicht wird in Promis$D$ ausgenutzt, um zum Beispiel die Liste der Wortarten aufzustellen, die für Auslassungsfehler in Frage kommen.

Allgemeiner gesehen lässt sich festhalten, dass im vorliegenden Ansatz eine Reihe von Möglichkeiten bestehen, die Verfahren zur Fehleranalyse sowohl im Bereich der „nicht-linguistischen“ als auch der linguistischen Beschränkungen so zu modifizieren beziehungsweise zu adaptieren, dass die Rate der Fehlererkennung zwar steigt, aber dennoch nicht an jeder beliebigen Position beliebige Fehlerhypothesen angenommen werden. Wie gezeigt wurde, konnte allerdings auch in der vorliegenden Form eine durchschnittliche Quote von 69% erkannten Fehlern erreicht werden.[3]

Zusammenfassend lässt sich festhalten, dass in Promis$D$ ein Übungs-

[3]Schließlich könnte die Integration von unterschiedlich gewichteten Attributen (Menzel und Schröder, 1998a) die Erklärungsfähigkeit des Systems erhöhen, da auch in den Fällen, in denen eine einzige Erklärung nicht erreicht werden kann, die Berücksichtigung einer feiner gestaffelten Gewichtung Erfolg bringen würde.

typ realisiert worden ist, der gegenüber herkömmlichen Formen nicht nur didaktischen Anforderungen genügt, sondern auch eine deutlich flexiblere Bearbeitung der Aufgaben zulässt. Obwohl aufgrund der Offenheit der Aufgabenstellung mit einer größeren Unsicherheit bezüglich der Diagnosequalität gerechnet werden muss, hat sich bei der Betrachtung des HU/FU-Korpus herausgestellt, dass mit der hier vorgestellten Grammatik und den Verfahren zur Fehlererkennung schon ein wesentlicher Teil der aufgetretenen Phänomene analysiert werden konnte. In Bezug auf die Interaktion von Effizienz und Umfang der identifizierbaren Fehler hat sich gezeigt, dass einerseits der Suchraum mit Hilfe der „harten“ Beschränkung von Fehlerpositionen verkleinert werden kann und andererseits die belegbaren Fehler berücksichtigt werden müssen, da eine Beschränkung die Erkennung eines Fehlers an einer bestimmten, möglicherweise plausiblen Position verhindert. Die Schlussfolgerung aus den Ergebnissen muss für diesen Aspekt daher lauten, dass auf der einen Seite zwar durch die Aufweichung der Kriterien für die Fehlerannahme die Effizienz leiden wird, auf der anderen Seite aber nur mit Hilfe eben dieser Aufweichung eine fortgesetzte Fehlererkennung möglich ist, wenn eine größere Grammatik für den praktischen Einsatz integriert wird.

Literaturverzeichnis

Alexandersson, Jan, Engel, Ralf, Kipp, Michael, Koch, Stephan, Küssner, Uwe, Reithinger, Norbert und Stede, Manfred: Modeling negotiation dialogs. In: Wahlster, Wolfgang (Hg.) *Verbmobil: Foundations of Speech-to-Speech Translation*, Springer, Berlin, S. 441–451. 2000.

Alexandersson, Jan, Maier, Elisabeth und Reithinger, Norbert: A robust and efficient three-layered dialog component for a speech-to-speech translation system. Verbmobil-Report 50, DFKI, Saarbrücken, 1994.

Appelo, Lisette und de Jong, Franciska (Hg.): *Twente Workshop on Language Technology - Computer-Assisted Language Learning*. Universiteit Twente, Enschede, 1994.

Atwell, Eric: *The Language Machine*. British Council, London, 1999. URL `http://www.comp.leeds.ac.uk/eric/atwell99.pdf`.

Austin, Peter K.: Lexical functional grammar. In: Smelser, Neil J. und Baltes, Paul (Hg.) *International Encyclopedia of Social and Behavioral Sciences*, Elsevier, Amsterdam, S. 8748–8754. 2001.

Baayen, R. Harald: *Word Frequency Distributions*. Kluwer Academic Publishers, Dordrecht, 2001.

Bauer, Pascal, John, Roul S., Kronenberg, Friedrich, Krüger, Anja, Menzel, André, Reuer, Veit und Unsöld, Robert: *Abschlussbericht PROMISE Studienprojekt*. Osnabrück, 1994.

Bod, Rens und Kaplan, Ronald: A probabilistic corpus-driven model for lexical functional analysis. In: *Proceedings of ACL/COLING 98*. Montreal, 1998. URL `http://www.aclweb.org/anthology/P98-1022.pdf`.

Bod, Rens, Scha, Remko und Sima'an, Khalil (Hg.): *Data-Oriented Parsing*. CSLI, Stanford, 2003.

Bouma, Gosse: Defaults in unification grammar. In: *Proceedings of the 28th Conference of the Association for Computational Linguistics (ACL)*. Pittsburgh, 1990, S. 165–172. URL `http://www.aclweb.org/anthology/P90-1021.pdf`.

Bresnan, Joan (Hg.): *The mental representation of grammatical relations*. MIT Press, Cambridge, MA, 1982.

Bresnan, Joan: *Lexical-Functional Syntax*. Blackwell, Oxford, 2001.

Bröker, Norbert: *Eine Dependenzgrammatik zur Kopplung heterogener Wissensquellen*. Niemeyer, Tübingen, 1999.

Bußmann, Hadumod: *Lexikon der Sprachwissenschaft*. Alfred Kröner, Stuttgart, zweite Auflage, 1990.

Butt, Miriam, King, Tracy Holloway, Niño, María-Eugenia und Segond, Frédérique: *A Grammar Writer's Cookbook*. CSLI Publications, Stanford, 1999.

Butzkamm, Wolfgang: *Psycholinguistik des Fremdsprachenunterrichts: natürliche Künstlichkeit: von der Muttersprache zur Fremdsprache*. A. Francke, Tübingen, zweite Auflage, 1993.

Campone, I. und Wehrli, Eric: Le traitement des expressions figées dans l'analyseur FIPS. In: *Actes de la conférence TALN-96*. 1996, S. 210–219.

Carberry, Sandra: *Plan Recognition in Natural Language Dialogue*. MIT Press, Cambridge, MA, 1990.

Carletta, J., Isard, A., Isard, S., Kowtko, J., Doherty-Sneddon, G. und Anderson, A.: The reliability of a dialogue structure coding scheme. In: *Computational Linguistics*, Band 23(1):S. 13–31, 1997. URL `http://www.aclweb.org/anthology/J97-1002.pdf`.

Carpenter, Bob: Skeptical and credulous default unification with applications to templates and inheritance. In: Briscoe, Ted, Copestake, Ann und Paiva, Valeria de (Hg.) *Inheritance, Defaults and the Lexicon*, Cambridge University Press, Cambridge, S. 13–37. 1993.

Chapelle, Carol A.: *Computer applications in second language acquisition : foundations for teaching, testing and research*. Cambridge University Press, Cambridge, 2001.

Chomsky, Noam: *Lectures on Government and Binding*. Foris, Dordrecht, 1981.

Clément, Lionel, Gerdes, K. und Kahane, S.: An lfg-type grammar for german based on the topological model. In: *Proceedings of LFG02*. Athen, 2002, S. 116–129. URL `http://cslipublications.stanford.edu/LFG/7/lfg02cgk.pdf`.

Clément, Lionel und Kinyon, Alexandra: XLFG - an LFG parsing scheme for french. In: *Proceedings of LFG01.* , 2001, S. 47–65. URL `http://cslipublications.stanford.edu/LFG/6/lfg01clementkinyon.pdf`.

Dalrymple, Mary (Hg.): *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach.* MIT Press, Cambridge, MA, 1999.

Dalrymple, Mary, Kaplan, Ronald M., Maxwell, John T. und Zaenen, Annie (Hg.): *Formal Issues in Lexical-Functional Grammar.* CSLI, Stanford, 1995.

deSmedt, William H.: Herr kommissar: An icall conversation simulator for intermediate german. In: Holland et al. (1995), S. 183–199.

Deutsch aktiv (1986): *Deutsch aktiv, Neu Lehrbuch 1 A.* Berlin, 1986.

Diehl, Erika, Christen, Helen, Leuenberger, Sandra, Pelvat, Isabelle und Studer, Thérèse: *Grammatikunterricht: Alles für die Katz?* Niemeyer, Tübingen, 2000.

Dokter, Duco und Nerbonne, John: A session with Glosser-RuG. In: Jager et al. (1998), S. 88–94.

Dokter, Duco, Nerbonne, John, Schürcks-Grozeva, Lily und Smit, Petra: Glosser-RuG: a user study. In: Jager et al. (1998), S. 167–176.

Dörre, J.: Weiterentwicklung des Earley-Algorithmus für kontextfreie und ID/LP-Grammatiken. LILOG-Report 28, IBM Deutschland, Stuttgart, 1987.

Dowty, D. R.: Thematic proto-roles and argument selection. In: *Language*, Band 67:S. 547–619, 1991.

Dreyer, Hilke und Schmitt, Richard: *Lehr- und Übungsbuch der deutschen Grammatik - Neubearbeitung.* Hueber, Ismaning, 2000.

Dulay, Heidi, Burt, Marina und Krashen, Stephen: *Language Two.* Oxford University Press, New York, 1982.

Earley, Jay: An efficient context-free parsing algorithm. In: *Communications of the ACM*, Band 13:S. 94–102, 1970.

Eisele, Andreas und Dörre, Jochen: A lexical-functional grammar system in prolog. In: *Proceedings of the 11th COLING 86.* Bonn, 1986, S. 551–553.

Eisenberg, Peter: *Grundriß der deutschen Grammatik - Der Satz*, Band 2. Metzler, Stuttgart, 1999.

Ellis, R.: *The Study of Second Language Acquisition.* Oxford University Press, New York, zweite Auflage, 1994.

Falk, Yehuda N.: *Lexical-Functional Grammar : an Introduction to Parallel Constraint-based Syntax*. CSLI Publications, Stanford, 2001.

Fawcett, R.P. und Taylor, M.M.: A generative grammar for local discourse structure. In: Taylor, M.M., Neel, F. und Bouwhuis, D.G. (Hg.) *The structure of Multimodal Dialog*, Elsevier, North-Holland. 1989.

Foster, Jennifer: A unification strategy for parsing agreement errors. In: Pilière, Catherine (Hg.) *Proceedings of the 5th ESSLLI 2000 Student Session*. Birmingham, 2000.

Foster, Jennifer: Parsing ungrammatical input: An evaluation procedure. In: *Proceedings of LREC2004*. Lisbon, 2004, S. 2039–2042.

Foster, Jennifer und Vogel, Carl: Parsing ill-formed text using an error grammar. In: *Artificial Intelligence Review*, Band 21:S. 269–291, 2004.

Fouvry, Frederik: *Robust processing for constraint-based grammar formalisms*. Dissertation, Gradudate School, University of Essex, [Colchester, UK], 2003.

Frank, Anette, King, Tracy Holloway, Kuhn, Jonas und Maxwell, John: Optimality theory style constraint ranking in large-scale lfg grammars. In: Sells, Peter (Hg.) *Formal and Empirical Issues in Optimality-theoretic Syntax*, CSLI Publications, Stanford, S. 367–397. 2001.

Gamper, Johann und Knapp, Judith: Adaption in a language learning system. In: *Online-Proceedings des 9. GI-Workshops: ABIS-Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen.* , 2001. URL `http://www.kbs.uni-hannover.de/~{}henze/ABIS_Workshop{2001}/final/Gamper_final.pdf`.

Garrett, Nina: ICALL and second language acquisition. In: Holland et al. (1995), S. 345–358.

Gnutzmann, C. und Kiffe, M.: Mündliche fehler und fehlerkorrekturen im hochschulbereich. zur einstellung von studierenden der anglistik. In: *Fremdsprachen Lehren und Lernen*, Band 22:S. 91–128, 1993.

Götze, Lutz: Quousque tandem? von der verführbarkeit des menschen durch die maschine. In: *Deutsch als Fremdsprache*, Band 42(1):S. 11–14, 2004.

Granger, Sylviane, Vandeventer, Anne und Hamel, Marie-Josée: Analyse de corpus d'apprenants pour l'ELAO basé sur le TAL. In: *T.A.L.*, Band 42(2):S. 609–621, 2001.

Grüner, Margit und Hassert, Timm: *Computer im Deutschunterricht*. Langenscheidt, München, 2000.

Hamburger, Henry: Viewpoint abstraction: a key to conversational learning. In: Appelo und de Jong (1994), S. 23–32.

Hamburger, Henry: Tutorials tools for language learning by two-medium dialogue. In: Holland et al. (1995), S. 183–199.

Handke, Jürgen: Wizdom: A multi-purpose language tutoring system based on ai techniques. In: Swartz und Yazdani (1992), S. 293–306.

Heift, Trude: *Designed Intelligence: A Language Teacher Model*. Dissertation, Simon Fraser University, Burnaby, B.C., 1998.

Heift, Trude: Error-specific and individualized feedback in a web-based language tutoring system: Do they read it? In: *ReCALL*, Band 13(2):S. 129–142, 2001.

Heift, Trude: Multiple learner errors and meaningful feedback: A challenge for ICALL systems. In: *CALICO*, Band 20(3):S. 533–548, 2003.

Helbig, Gerhard und Buscha, Joachim: *Deutsche Grammatik - Ein Handbuch für den Ausländerunterricht*. Langenscheidt Verlag Enzyklopädie, Leipzig, 17. Auflage, 1996.

Hendrickson, James M.: Error correction in foreign language teaching: Recent theory, research and practice. In: Croft, Kenneth (Hg.) *Readings on English as a Second Language*, Winthrop Publisher, Cambridge, MA, S. 153–174. Zweite Auflage, 1980.

Hennig, Mathilde: Die darstellung des tempussystems in deutschen grammatiken. In: *Deutsch als Fremdsprache*, Band 4, 1997.

Henrici, Gert und Herlemann, Brigitte: *Mündliche Korrekturen im Fremdsprachenunterricht*. Goethe-Institut, München, 1986.

Heringer, Hans Jürgen: *Aus Fehlern lernen*. Augsburg, 1995. CD-ROM für Win9x/NT.

Higgins, John: *Computers and English language learning*. Intellect, Oxford, 1995.

Holland, V. Melissa, Kaplan, Jonathan D. und Sams, Michelle R. (Hg.): *Intelligent Language Tutors*. Erlbaum, Mahwah, NJ, 1995.

Hu, Quian, Hopkins, Jeff und Phinney, Marianne: Native English writing assistant - a CALL product for english reading and writing. In: Jager et al. (1998), S. 95–100.

Hubbard, Philip L.: Non-transformational theories of grammar: Implications for language teaching. In: Odlin, Terence (Hg.) *Perspectives on Pedagogical Grammar*, Cambridge University Press, Cambridge, S. 49–71. 1994.

Jager, Sake, Nerbonne, John und Van Essen, Arthur (Hg.): *Language teaching and language technology*. Swets and Zeitlinger, Lisse, 1998.

Jensen, K., Heidorn, G. E., Miller, L. A. und Ravin, Y.: Parse fitting and prose fixing: Getting hold on ill-formdness. In: *Computational Linguistics*, Band 9(3-4):S. 147–160, 1983. URL `http://www.aclweb.org/anthology/J83-3002.pdf`.

John, Roul Sebastian: PROMISE: Steps towards communicative english language teaching in an interactive CALL system. In: Appelo und de Jong (1994), S. 117–118.

Kaplan, Ronald M. und Bresnan, Joan: Lexical-Functional Grammar: A formal system for grammatical representation. In: Bresnan (1982), S. 173–281.

Kaplan, Ronald M. und Maxwell, John T.: LFG grammar writer's workbench. Technischer Bericht, Xerox PARC, 1996. URL `ftp://ftp.parc.xerox.com/pub/lfg/lfgmanual.ps`.

Kaplan, Ronald M., Netter, Klaus, Wedekind, Jürgen und Zaenan, Annie: Translation by structural correspondences. In: *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics*. Manchester, 1989, S. 272–281. URL `http://www.aclweb.org/anthology/E89-1037.pdf`.

Kars, Jürgen und Häussermann, Ulrich: *Grundgrammatik Deutsch*. Diesterweg, Sauerländer, Frankfurt a.M., Aarau, 1997.

Karttunen, Lauri: Features and values. In: *Proc. 10th Int. Conference on Computational Linguistics (COLING)*. Stanford, 1984, S. 28–33. URL `http://www.aclweb.org/anthology/P84-1008.pdf`.

Kato, Tsuneaki: Yet another chart-based technique for parsing ill-formed input. In: *Proc. 4th Conference on Applied Natural Language Processing (ANLP)*. Stuttgart, 1994, S. 107–112. URL `http://www.aclweb.org/anthology/A94-1018.pdf`.

Kleppin, Karin: *Fehler und Fehlerkorrektur*. Langenscheidt, Berlin, 1998.

Kleppin, Karin und Königs, Frank G.: *Der Korrektur auf der Spur - Untersuchungen zum mündlichen Korrekturverhalten von Fremdsprachenlehrern*. Brockmeyer, Bochum, 1991.

Königs, Frank G.: Fehlerkorrektur. In: Bausch, Karl-Richard, Christ, Herbert und Krumm, Hans-Jürgen (Hg.) *Handbuch Fremdsprachunterricht*, A. Francke, Tübingen, S. 268–272. Dritte Auflage, 1995.

Krashen, Stephen: *The Input Hypothesis.* Pergamon Press, Oxford, 1985.

Kroeger, Paul R.: *Analyzing Syntax – A Lexical-functional Approach.* Cambridge University Press, Cambrige, 2004.

Krüger-Thielmann, Karin: *Wissensbasierte Sprachlernsysteme.* Narr, Tübingen, 1992.

Kuhn, Jonas und Rohrer, Christian: Approaching ambiguity in real-life sentences - the application of an Optimality Theory-inspired constraint ranking in a large-scale LFG grammar. In: *Proc. der Jahrestagung der Sektion Computerlinguistik der DGfS.* Heidelberg, 1997.

Langer, Hagen: *Parsing-Experimente: praxisorientierte Untersuchungen zur automatischen Analyse des Deutschen.* Peter Lang, Frankfurt am Main u.a., 2001.

Lee, Kong Joo, Kweon, Cheol Jung, Seo, Jungyun und Kim, Gil Chang: A robust parser based on syntactic information. In: *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics (EACL).* Dublin, 1995, S. 223–228. URL `http://www.aclweb.org/anthology/E95-1031.pdf`.

Levelt, Willem J. M.: *Speaking: From intention to articulation.* MIT Press, Cambridge, MA, 1989.

Levin, Lori S. und Evans, David A.: ALICE-chan: A case study in ICALL theory and practice. In: Holland et al. (1995), S. 77–98.

Levy, Michael: *Computer-Assisted Language Learning - Context and Conceptualization.* Clarendon Press, Oxford, 1997.

Ludewig, Petra: *Korpusbasiertes Kollokationslernen – Computer-Assisted Language Learning als prototypisches Anwendungsszenario der Computerlinguistik.* Habilitationsschrift an der Universität Osnabrück, Fachbereich Sprach- und Literaturwissenschaften, 2003.

Ludewig, Petra und Hötker, Wilfried (Hg.): *Lexikonimport, Lexikonexport: Studien zur Wiederverwertung lexikalischer Informationen.* Niemeyer, Tübingen, 1996.

Lyon, Gordon: Syntax-directed least-error analysis for context-free languages: A practical approach. In: *Communications of the ACM*, Band 17(1):S. 3–14, 1974.

Lyster, Roy und Ranta, Leila: Corrective feedback and learner uptake. In: *Studies in Second Language Acquisition*, Band 20:S. 37–66, 1997.

Macht, Konrad: Vom umgang mit fehlern. In: Timm, Johannes-Peter (Hg.) *Englisch lernen und lehren - Didaktik des Englischunterrichts*, Cornelsen, Berlin, S. 353–365. 1998.

Maxwell, John T., III und Kaplan, Ronald M.: An efficient parser for LFG. In: Butt, Miriam und King, Tracy Holloway (Hg.) *On-line Proceedings of the First LFG Conference (LFG96).* , Grenoble, 1996. URL `http://www-csli.stanford.edu/publications/LFG/1/maxwell.ps`.

Mellish, Chris S.: Some chart-based techniques for parsing ill-formed input. In: *Proc. 27th Conference of the Association for Computational Linguistics (ACL).* , 1989, S. 102–109. URL `http://www.aclweb.org/anthology/P89-1013.pdf`.

Menzel, Wolfgang: *Modellbasierte Fehlerdiagnose in Sprachlehrsystemen.* Niemeyer, Tübingen, 1992.

Menzel, Wolfgang: Parsing mit inkonsistenten grammatiken. In: *Kognitionswissenschaft*, Band 9:S. 175–184, 2002.

Menzel, Wolfgang: *Error diagnosis and feedback generation for language tutoring systems.* Lille, 2003. Slides for the ELSNET Summer School, URL `http://nats-www.informatik.uni-hamburg.de/pub/ESS2003/WebHome/ess-allk.ps.gz`.

Menzel, Wolfgang, Herron, Daniel, Morton, Rachel, Pezzotta, Dario, Bonaventura, Patrizia und Howarth, Peter: Interactive pronunciation training. In: *ReCall*, Band 13(1):S. 67–78, 2001.

Menzel, Wolfgang und Schröder, Ingo: Constraint-based diagnosis for intelligent language tutoring systems. In: *Proceedings IT & KNOWS, XV. IFIP World Computer Congress.* Vienna/Budapest, 1998a, S. 484–497.

Menzel, Wolfgang und Schröder, Ingo: Constraint-based diagnosis for intelligent language tutoring systems. Report nr. FBI-HH-B-208-98, Fachbereich Informatik, Universitä Hamburg, 1998b.

Menzel, Wolfgang und Schröder, Ingo: Model-based diagnosis under structual uncertainty. In: *Proc. 13th European Conference on Artificial Intelligence (ECAI).* 1998c, S. 284–288.

Michaud, Lisa N. und McCoy, Kathleen F.: Evaluating a model to disambiguate natural language parses on the basis of user language proficiency. In: *Proc. 9th International Conference on User Modeling.* Springer, Heidelberg, 2003, S. 96–105.

Mohanan, K.P.: Grammatical relations and clause structure in malayalam. In: Bresnan (1982), S. 504–589.

Müller, Stefan: *Deutsche Syntax deklarativ : head-driven phrase structure grammar für das Deutsche*. Niemeyer, Tübingen, 1999.

Nagata, Noriko: Intelligent computer feedback for second language instruction. In: *The Modern Language Journal*, Band 77(3):S. 330–339, 1993.

Naumann, Sven und Langer, Hagen: *Parsing*. Teubner, Stuttgart, 1994.

Nerbonne, John: Computer-assisted language learning and natural language processing. In: Mitkov, Ruslan (Hg.) *Handbook of Computational Linguistics*, Oxford University Press, Oxford, S. 670–698. 2002.

Nerbonne, John, Jager, Sake und van Essen, Arthur: Language teaching and language technology: An introduction. In: Jager et al. (1998), S. 1–10.

Nordlinger, Rachel: *Constructive Case: Evidence from Australian Languages*. CSLI, Stanford, 1998.

Oflazer, Kemal: Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction. In: *Computational Linguistics*, Band 22(1):S. 73–89, 1996. URL `http://www.aclweb.org/anthology/J96-1003.pdf`.

Ouanes, Hédi: Zur problematik des umgangs mit sprachlichen fehlern im unterricht daf. In: *Info DaF*, Band 19:S. 732–739, 1992.

Pienemann, Manfred: *Language Processing and Second Language Development - Processability Theory*. Benjamins, Amsterdam, 1998.

Pollard, Carl Jesse und Sag, Ivan A.: *Information-based Syntax and Semantics, Volume 1: Fundamentals*. CSLI, Menlo Park, 1987.

Pollard, Carl Jesse und Sag, Ivan A.: *Head-driven phrase structure grammar*. CSLI, Stanford, 1994.

Reimann, Monika: *Grundstufen-Grammatik für Deutsch als Fremdsprache*. Hueber, Ismaning, 1996.

Reuer, Veit: Dialogue processing in a CALL-system. In: *Proc. 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Bergen, 1999, S. 253–256. URL `http://www.aclweb.org/anthology/E99-1037.pdf`.

Reuer, Veit: Error-recognition and parsing of syntactically mildly ill-formed natural language. In: *Proc. LFG00 Conference*. CSLI Publications, Berkeley, 2000, S. 215–225. URL `http://cslipublications.stanford.edu/LFG/5/lfg00reuer.pdf`.

Reuer, Veit: Error recognition and feedback with Lexical Functional Grammar. In: *CALICO Journal*, Band 20(3):S. 497–512, 2003.

Reuer, Veit und Kühnberger, Kai-Uwe: Feature constraint logic and error detection in ICALL systems. In: Blache, P. und Stabler, E. (Hg.) *LACL 2005, LNAI 3492*, Springer, Berlin Heidelberg, S. 271–286. 2005.

Ritter, Markus: *Computer und handlungsorientierter Unterricht*. Ludwig Auer, Donauwörth, 1995.

Rohrer, Christian und Schwarze, Christoph: Eine grammatiktheorie für die prozedurale linguistik: Die lexikalisch-funktionale grammatik (lfg). In: Schnelle, Helmut und Rickheit, Gert (Hg.) *Sprache in Mensch und Computer - Kognitive und neuronale Sprachverarbeitung*, Westdeutscher Verlag, Opladen, S. 9–62. 1988.

Rösler, Dietmar und Tschirner, Erwin: Neue medien und deutsch als fremdsprache - viele fragen und ein aufruf zur diskussion. In: *Deutsch als Fremdsprache*, Band 3:S. 144–155, 2002.

Rüschoff, Bernd und Wolff, Dieter: *Fremdsprachenlernen in der Wissensgesellschaft: zum Einsatz der Neuen Technologien in Schule und Unterricht*. Hueber, Ismaning, 1999.

Rug, Thomas und Tomaszewski, Andreas: *Grammatik mit Sinn und Verstand*. Klett Edition Deutsch, München, 1993.

Rypa, Marikka und Feuerman, Ken: CALLE: An exploratory environment for foreign language learning. In: Holland et al. (1995), S. 55–76.

Sågvall Hein, Anna: A grammar checking module for swedish. SCARRIE Deliverable 6.6.3, Uppsala University, Uppsala, 1998.

Salaberry, M. Rafael: A theoretical foundation for the development of pedagogical tasks in computer-mediated communication. In: *CALICO Journal*, Band 14(1):S. 5–34, 1996.

Salaberry, M. Rafael: Review of sake jager, john a. nerbonne, a. j. van essen (eds.): Language teaching and language technology. In: *Language Learning & Technology*, Band 4(1):S. 22–25, 2000. URL `http://llt.msu.edu`.

Schneider, David und McCoy, Kathleen F.: Recognizing syntactic errors in the writing of second language learners. In: *Proc. 17th Int. Conference on Computational Linguistics (COLING)*. Montreal, 1998, S. 1198–1204. URL `http://www.aclweb.org/anthology/P98-2196.pdf`.

Schröder, Ingo, Menzel, Wolfgang, Foth, Kilian und Schulz, Michael: Modeling dependency grammar with restricted constraints. In: *T.A.L.*, Band 41(1):S. 113–142, 2000.

Schulze, Mathias, Hamel, Marie-Josée und Thompson, June: Language processing in CALL. In: *ReCALL*, 1999. Special Issue.

Schwarze, Christoph: *Grammatik der italienischen Sprache*. Niemeyer, Tübingen, 1995.

Schwind, Camilla: Sensitive parsing: Error analysis and explanation in an intelligent language tutoring system. In: *Proc. 12th Int. Conference on Computational Linguistics (COLING)*. , 1988, S. 608–613. URL `http://www.aclweb.org/anthology/P88-2127.pdf`.

Schwind, Camilla B.: Error analysis and explanation in knowledge based language tutoring. In: Appelo und de Jong (1994), S. 77–92.

Smolka, G.: Feature constraint logics for unification grammars. In: *Journal of Logic Programming*, Band 12:S. 51–87, 1992.

Storch, Günther: *Deutsch als Fremdsprache: eine Didaktik*. Fink (UTB), München, 1999.

Swartz, M.L. und Yazdani, M. (Hg.): *Intelligent Tutoring Systems for Foreign Language Learning*. Springer, Berlin, 1992.

Thelen, Tobias: Wie passt das wort betten in das haus? grundlagen und ergebnisse des computerprogramms mops zur vermittlung der schärfungsschreibung. In: Röber-Siekmeyer, Christa und Tophinke, Doris (Hg.) *Schärfungsschreibung im Fokus. Zur schriftlichen Repräsentation sprachlicher Strukturen im Spannungsfeld von Sprachwissenschaft und Didaktik*, Schneider, Baltmannsweiler. 2002.

Themen (1997): *Themen neu 1: Lehrwerk für Deutsch als Fremdsprache*. Ismaning, zweite Auflage, 1997.

Thomann, Johannes: Lfg as a pedagogical grammar. In: *Proceedings of LFG02*. Athen, 2002, S. 366–372. URL `http://cslipublications.stanford.edu/LFG/7/lfg02thomann.pdf`.

Uesseler, Manfred: Fehlerkorrektur - einige überlegungen und praktische hinweise. In: *Fremdsprachlicher Unterricht*, Band 45(2):S. 70–72, 1992.

Vandeventer, Anne: Creating a grammar checker for CALL by constraint relaxation: a feasibility study. In: *ReCALL*, Band 13(1):S. 110–120, 2001.

Vogel, Carl und Cooper, Robin: Robust chart parsing with mildly inconsistent feature structures. In: Schöter, Andreas und Vogel, Carl (Hg.) *Nonclassical Feature Systems*, Edinburgh University, Band 10 von *Edinburgh Working Papers in Cognitive Science*, S. 197–216. 1995.

Way, Andy: Machine translation using lfg-dop. In: Bod, R., Scha, R. und Sima'an, K. (Hg.) *Data-Oriented Parsing*, CSLI, Stanford, S. 359–384. 2003.

Weischedel, Ralph M. und Sondheimer, Norman K.: Meta-rules as a basis for processing ill-formed input. In: *Computational Linguistics*, Band 9:S. 161–177, 1983. URL `http://www.aclweb.org/anthology/J83-3003.pdf`.

Wolff, Dieter: Neue technologien und fremdsprachliches lernen - versuch einer bestandsaufnahme. In: *Deutsch als Fremdsprache*, Band 35:S. 136–140 + 205–211, 1998.

Zock, Michael: SWIM or SINK: the problem of communicating thought. In: Swartz und Yazdani (1992), S. 235–247.

Danksagung

Bedanken möchte ich mich bei Jürgen Kunze und bei Wolfgang Menzel für die Betreuung der Arbeit sowie bei Claus Rollinger für seine großzügige Haltung während unseres Projekts. Petra Ludewig saß mir in der Endphase gegenüber und war quasi gezwungen, auf meine Fragen und Probleme mit fachlich kompetenter und moralischer Hilfe zu reagieren. Auch ihr gebührt besonderer Dank. Für die Hinweise und Hilfen bei kniffligeren theoretischen Fragen möchte ich mich bei Helmar Gust und speziell bei Kai-Uwe Kühnberger bedanken. Ein Dank auch an Heike, Horst und Nicole vom Institut für deutsche Sprache und Linguistik, mit denen ich für einige Jahre zusammenarbeiten durfte. „Of course, non of these people is to blame for any remaining problems. My computer accepts full responsibility; it put the mistakes in when I wasn't looking.“ (Falk, 2001, S. xiii)

Und schließlich möchte ich mich ganz besonders bei Anneli, Linus und Okke bedanken, die mich in anstrengenden Zeiten ertragen und zugleich unterstützt haben.

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Arbeit selbständig ohne fremde Hilfe verfaßt und nur die angegebene Literatur und Hilfsmittel verwendet zu haben.
Veit Reuer

10. Februar 2003