**Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH**

**Research
Report**
RR-90-17

Generalisierte Phrasenstruktur-Grammatiken und ihre Verwendung zur maschinellen Sprachverarbeitung

Stephan Busemann

Dezember 1990

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
D-6750 Kaiserslautern, FRG
Tel.: (+49 631)205-3211/13
Fax: (+49 631)205-3210

Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, FRG
Tel.: (+49 681) 302-5252
Fax: (+49 681) 302-5341

Generalisierte Phrasenstruktur-Grammatiken und ihre Verwendung zur maschinellen Sprachverarbeitung

Stephan Busemann

DFKI-RR-90-17

Der vorliegende Artikel ist eine geringfügig überarbeitete Version des dritten und des vierten Kapitels der Dissertation des Autors [Busemann 1990]. Die zugrundeliegende Forschung wurde im Rahmen des vom Bundesminister für Forschung und Technologie unter dem Kennzeichen 1013211 geförderten Projekts KIT-FAST an der Technischen Universität Berlin durchgeführt. Die Arbeit wurde am DFKI im Projekt DISCO, das vom BMFT unter dem Kennzeichen ITW 9002 gefördert wird, fertiggestellt.

Generalisierte Phrasenstruktur-Grammatiken und ihre Verwendung zur maschinellen Sprachverarbeitung

Stephan Busemann
DFKI GmbH
Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11
Tel.: (0681) 302 5286

e-mail: busemann@dfki.uni-sb.de

Zusammenfassung

Der vorliegende Artikel setzt sich mit der Syntaxtheorie der Generalisierten Phrasenstruktur-Grammatiken (GPSG) auseinander, gibt eine neue formale Definition des aktuellen Formalismus aus [Gazdar *et al.* 1985] an und zeigt die mit diesem Formalismus verbundenen Probleme auf. Darüber hinaus wird begründet, warum der Formalismus nicht effizient implementierbar ist. Es wird eine konstruktive Version von GPSG vorgeschlagen, die für die maschinelle Sprachverarbeitung (Parsing und Generierung) geeignet ist. Der Artikel kann gleichzeitig als eine Grundlage für Lehrveranstaltungen über GPSG dienen.

This article describes the syntax theory of Generalized Phrase Structure Grammar (GPSG), introduces a new formal definition for the formalism described in [Gazdar *ei al.* 1985], and reveals the problems connected with this formalism. Moreover it is shown why the formalism cannot efficiently be implemented. A constructive version of GPSG is suggested that is suitable for parsing and generation. This report may also serve as a basis for lectures about GPSG.

Inhaltsverzeichnis

Danksagung

Die dieser Arbeit zugrundeliegende Forschung wurde im Rahmen der EUROTRA-D-Begleitforschung an der TU Berlin durchgeführt, wo in den Projekten KIT-NASEV und KIT-FAST[1] der Formalismus der Generalisierten Phrasenstruktur-Grammatiken für die maschinelle Übersetzung nutzbar gemacht wurde.

Ich danke Christa Hauenschild, Bill Keller, James Kilbury, Susanne Preuß, Carla Umbach und Wilhelm Weisweber für ungezählte fruchtbare Diskussionen im Verlauf der vergangenen fünf Jahre, durch die sie wesentlich zum Zustandekommen dieser Arbeit beigetragen haben. Für darin enthaltene Fehler bin selbstverständlich allein ich verantwortlich.

l Einführung

Viele Syntaxformalismen für natürliche Sprachen kodieren linguistisch-theoretische Aussagen indirekt. Einfache Phrasenstrukturregeln beschreiben z.B. sowohl Dominanzbeziehungen als auch Präzendenzbeziehungen zwischen Knoten. Damit gehen Aussagen über Wortstellung (etwa (1)) nicht direkt aus dem Formalismus hervor, sondern müssen durch Prüfung aller Regeln verifiziert werden.

(1) *Wenn NPs und PPs als Schwestern auftreten, so stehen NPs vor PPs*

Die Theorie der Generalisierten Phrasenstruktur-Grammatiken (GPSG) wurde mit dem Ziel entwickelt, linguistische Aussagen über Eigenschaften natürlicher Sprachen *auf direkte Weise* zu kodieren. Die Gesamtheit dieser Aussagen[2] bestimmt, welche Ausdrücke einer natürlichen Sprache wohlgeformt sind und welche nicht. Z.B. wird die genannte in Phrasenstrukturregeln enthaltene Information in GPSG von Regeln verschiedenen Typs repräsentiert, nämlich von sogenannten ID-Regeln, die Aussagen über unmittelbare Dominanz darstellen und von sogenannten LP-Aussagen, die die Abfolge von Schwesterknoten beschränken. Mit einer LP-Aussage kann ein sprachliches Fakt wie (1) direkt ausgedrückt werden (2).

(2) NP X PP

In GPSG gibt es noch weitere Regeltypen. Mithilfe von sogenannten Feature Cooccurence Restrictions (FCRs) läßt sich z.B. direkt ausdrücken, daß nur Verben ein Tempus haben. Feature Specification Defaults (FSDs) ermöglichen die Kodierung von Markiertheitskonventionen (etwa, daß eine NP im Englischen den Kasus Akkusativ erhält, sofern nicht andere Festlegungen entgegenstehen).

In wohl allen Ersetzungsregelsystemen für natürliche Sprachen lassen sich systematische Beziehungen zwischen Regeln auffinden. Ein bekanntes Beispiel ist die (verbunabhängige) Relation zwischen dem Objekt eines Aktiv-Satzes und dem Subjekt eines entsprechenden Passiv-Satzes. In GPSG werden diese und andere Beziehungen durch Metaregeln explizit ausgedrückt.

[1]KIT steht für *Künstliche Intelligenz und Textverstehen,* NASEV für *neue Analyse- und Syntheseverfahren m der maschinellen Übersetzung* und FAST für *Funktor-Argument-Strukturen im Transfer.*

[2]Sie umfaßt sowohl „universelle" als auch einzelsprachspezifische Aussagen.

Bestimmte als universell angesehene sprachliche Regularitäten werden mit allgemeinen Prinzipien erfaßt, den sogenannten Merkmalinstantiierungsprinzipien (MIPs). Sie sehen eine generelle Behandlung u.a. von Kongruenz, Koordination und Long-Distance-Phänomenen vor.

Diese linguistische Beschreibungsebene von GPSG stellt eine *Metasprache* mit eigener Syntax und Semantik dar. Ihre Mächtigkeit ist auf die Beschreibung kontextfreier Sprachen begrenzt. Eine GPS-Grammatik kann stets auf eine kontextfreie Grammatik zurückgeführt werden.

1.1 Aufbau der Arbeit

Im ersten Teil dieser Arbeit wird GPSG aus linguistischer Sicht als Theorie der Syntax natürlicher Sprachen beschrieben. Die wichtigste Grundlage der Diskussion bildet das Hauptwerk von GKPS. Zuerst jedoch wird die Entwicklung und der aktuelle Stand der linguistischen Forschung im Bereich der GPSG dargestellt. Abschnitt 2 zieht einige Hauptzüge der linguistischen Entwicklung von GPSG nach und führt die wesentlichen Elemente des Formalismus aus linguistischer Sicht ein. Dabei wird die Theorie von 1982 [Gazdar/Pullum 1982] zugrunde gelegt. Anschließend wird die Weiterentwicklung in GKPS skizziert und die Unterschiede hervorgehoben. Beide Versionen von GPSG spielen eine maßgebliche Rolle für Implementationen (vgl. Teil II).

Danach gibt Abschnitt 3 eine kurze, formale Definition einer GPS-Grammatik gemäß GKPS, indem die Syntax der Regeltypen angegeben und die Verwendung der Regeln und Prinzipien durch einen Ableitungsbegriff spezifiziert werden, wie er aus der Theorie formaler Sprachen bekannt ist. Dieser Abschnitt vermittelt noch keine linguistische Motivation für die Definitionen, gibt aber einen komprimierten Überblick über die Komponenten von GPSG und einige ihrer formalen Eigenschaften. Die Definitionen werden bei den nachfolgenden Betrachtungen ergänzt.

Abschnitt 4 enthält eine ausführliche Beschreibung der meisten Komponenten des Formalismus von GKPS, diskutiert linguistisch fragwürdige Vorgehensweisen und zeigt auf, an welchen Stellen nach Alternativen gesucht werden muß. Dabei werden hauptsächlich die Komponenten der Metasprache behandelt, die im zweiten Teil im Rahmen des Berliner GPSG-Formalismus (vgl. Abschnitt 6) rekonstruiert wurden; nämlich komplexe Kategorien, das ID/LP-Format, Metaregeln, FCRs und MIPs. Es wird wenig zu den semantischen Ansätzen im Rahmen der GPSG zu sagen sein, durchaus aber einiges zu problematischen Aspekten der Schnittstelle zwischen Syntax und Semantik.

Die im ersten Teil formulierte Kritik rüttelt zwar an den Grundpfeilern des GPSG-Ansatzes, doch dies bedeutet nicht, daß die Idee der Metasprache als solche untauglich wäre. Es bedeutet vielmehr, daß die Ansprüche an die Erklärungsfähigkeit der Metasprache realistisch formuliert werden müssen, und das meint: bescheidener. Dementsprechend wird im zweiten Teil dieser Arbeit eine Version von GPSG vorgestellt, die eine geringere Anzahl linguistischer Generalisierungen ausdrückt, dafür aber für die automatische Sprachverarbeitung geeignet ist.

In Abschnitt 5 werden Eigenschaften der späten GPSG diskutiert, derentwegen eine vollständige und originalgetreue Implementation praktisch nicht realisierbar

ist. Die wesentliche Ursache liegt in der ausschließlich deskriptiven Sicht verborgen, die die wohlgeformten syntaktischen Strukturen negativ charakterisiert, nämlich als diejenigen, die durch die verschiedenen Komponenten des Formalismus *nicht ausgeschlossen* werden. Stattdessen wird für eine konstruktive Sicht plädiert, die die wohlgeformten syntaktischen Strukturen grundsätzlich positiv charakterisiert als diejenigen, die durch die verschiedenen Komponenten des Formalismus *erzeugt* werden.

Auf der Grundlage dieser Sichtweise wird in Abschnitt 6 die Implementation der wesentlichen Komponenten des Formalismus beschrieben. Der Formalismus bildet das linguistische Kernstück des Berliner GPSG-Systems (Abschnitt 7), das als Bestandteil eines MÜ-Systems konzipiert wurde. Es erlaubt Parsing und Generierung mit unterschiedlichen Grammatiken (hier Fragmente für Deutsch und Englisch) sowie die bidirektionale Verwendung von Grammatiken; d.h. grammatisches Wissen ist unabhängig von Verarbeitungsstrategien repräsentiert.

Abschnitt 8 stellt die verwendeten Grammatiken vor. Inwieweit die postulierte Unabhängigkeit der Grammatiken vom Verarbeitungswissen im Berliner GPSG-System verwirklicht ist, untersucht Abschnitt 9. Er diskutiert darüber hinaus, welche grundsätzlichen Probleme sich bei einer bidirektionalen Verwendung von deklarativ repräsentiertem sprachlichen Wissen ergeben.

Vor dem Hintergrund der Trennung von grammatischem Wissen und Prozessen beschreibt Abschnitt 10 die Interaktion der Lexika mit den Grammatiken. Die Lexika wurden als Stammformenlexika implementiert und mit einer separaten Flexions- bzw. Lemmatisierungskomponente versehen. Um Übergenerierung zu vermeiden, erfordert dieser Ansatz eine sorgfältige Abstimmung mit der Syntax.

Der zweite Teil wird durch einen Blick auf andere GPSG-basierte computerlinguistische Ansätze in Abschnitt 11 abgerundet. Es werden eigenständige Weiterentwicklungen bestimmter Aspekte des Formalismus gezeigt, womit jeweils eine Vereinfachung unter Inkaufnahme eines gewissen Verlustes an linguistischer Aussagekraft verbunden ist. Dabei wird deutlich werden, daß den Implementationen sehr unterschiedliche Zielsetzungen und linguistische Ansprüche zugrundeliegen.

Schließlich diskutiert Abschnitt 12 einen speziellen Aspekt dieses Vergleichs, der einen Hauptunterschied zwischen der mittleren und der späten GPSG reflektiert; nämlich die Folgen der Kompilation von GPS-Grammatiken in kontextfreie Grammatiken und die der direkten Interpretation der Metasprache während der Verarbeitung.

1.2 Theorie und Formalismus: Eine Begriffsklärung

Vorab erscheint mir eine Klärung der Begriffe „Theorie" und „Formalismus" wichtig (eine generelle Diskussion und Abgrenzung findet sich in [Shieber 1988]). Unter der Theorie verstehe ich die linguistisch begründete Vorgehensweise zur Erklärung (und Vorhersage) sprachlicher Phänomene (z.B. die Trennung von Dominanz- und Präzedenzbeziehungen, um Fakten über die Wortstellung in natürlichen Sprachen explizit zu machen). Unter dem Formalismus verstehe ich eine formale Sprache, die die Formulierung von theorie-konformen Grammatiken erlaubt (z.B. die Definition des ID/LP-Formats und der Wirkungsweise von ID-Regeln und LP-Aussagen).

Mit einem GPSG-Formalismus sollen die von der GPSG-Theorie behaupteten

linguistischen Sachverhalte ausgedrückt werden können. Im Idealfall würde der Formalismus *genau* die theoretischen Aussagen reflektieren und könnte somit als *Interpretation* der Theorie betrachtet werden. Eine vollständige Übereinstimmung hat man in nichttrivialen Systemen bisher nicht erreicht. Formalismen können i.a. mehr ausdrücken, als die Theorie besagt. In GPSG kann man den Formalismus mißbrauchen (im Hinblick auf die *GPSG-Theorie),* indem man z.B. Informationen, die durch FCRs ausgedrückt werden, in ID-Regeln kodiert. Der Formalismus verbietet nicht, neue Merkmale zu definieren oder die Wertebereiche von Merkmalen erweitern zu dem Zweck, bestimmte Strukturbeschreibungen zu unterbinden.

Die vollständige Übereinstimmung zwischen Formalismus und Theorie ist kein vorrangiges Ziel mehr, wie sich an den unterschiedlichen Ansätzen zu einer Formalisierung von GPSG zeigt (ein Überblick findet sich in [Busemann 1990, :114ff]). Eleganz der Beschreibung, Effizienz der Verarbeitung und Transparenz der Regelsysteme gehören zu den wichtigeren Merkmalen. Dies sind Gründe, aus denen Grammatik-Formalismen auch unabhängig von einer bestimmten linguistischen Theorie bestehen, z.B. Functional Unification Grammar [Kay 1979] oder PATR-II [Shieber *et al.* 1983].

Teil I
Definition und Motivation von GPSG

2 Zur Entwicklung von GPSG

Die linguistischen Arbeiten zur GPSG begannen Ende der Siebziger Jahre und erwiesen sich als sehr stimulierend in einer Zeit, in der die Entwicklung moderner, unifikationsbasierter Grammatikformalismen (außer GPSG u.a. LFG [Kaplan/Bresnan 1982] und Categorial Unification Grammar (CUG) [Uszkoreit 1986c]) rasch voranschritt. Noch bevor GKPS erschien, eröffnete Pollard mit seiner Dissertation [Pollard 1984] bereits neue Perspektiven, die viele Ideen von GPSG aufgreifen und gleichzeitig den „Trend zum Lexikon" verstärken, der auf teilweise ähnliche Art durch LFG und C(U)G begründet worden war. Die linguistische Theoriebildung schritt in unvermindertem Tempo weiter und führte inzwischen zur Entwicklung von HPSG [Pollard/Sag 1987]. Gleichzeitig wurden aufgrund von GKPS Erweiterungen und Verbesserungen des GPSG-Formalismus vorgeschlagen, die sehr viele Ideen ausbauen, die in GKPS nur angedeutet sind. Zu keiner Zeit gab es also eine Art stabiler Plattform, die GPSG als ausgereifte Theorie mit langfristigem Bestand präsentiert, so wie dies sehr lange durch die grundlegende Arbeit von Kaplan und Bresnan zu LFG der Fall war.[3]

Die Anfänge der Arbeiten zu GPSG in den späten Siebziger Jahren entstanden parallel zu der Entwicklung der Transformationsgrammatik (TG) in Richtung auf die Government-Binding-Theorie (GB), die sich durch eine Beschränkung der Transformationskomponente auszeichnet. GPSG vollzieht einen radikaleren Bruch und verzichtet ganz auf Transformationen. Die rigide Beschränkung der generativen Kapazität von GPSG auf kontextfreie Sprachen ermöglichte den Nachweis, daß Transformationen für die Beschreibung vieler sprachlicher Phänomene nicht zwingend notwendig sind. Zudem hoffte man, so die bekannten, mathematischen Eigenschaften kontextfreier Grammatiken und der entsprechenden Parsingalgorithmen ausnutzen zu können.[4] Die Arbeit an GPSG weist einen im Vergleich zu anderen linguistisch motivierten Theorien ungewöhnlich hohen Grad der Formalisierung auf und stimulierte daher die Untersuchung mathematischer Eigenschaften natürlicher Sprachen.

GPSG ist, anders als TG, eine *monostratale* Theorie. GKPS versuchen zu zeigen, daß *eine* Ebene der syntaktischen Repräsentation genügt, um eine Reihe komplexer sprachlicher Phänomene zu beschreiben, die in TG-Ansätzen auf verschiedenen Ebenen getrennt behandelt werden mußten.

[3]Die Schattenseite einer solchen Stabilität ist ihre offensichtliche Selbstverstärkung; beim gegenwärtigen Stand der linguistischen Theoriebildung könnte sich Ekklektizismus (wie in HPSG) als innovativer erweisen als ein Verharren im Paradigma.

[4]Diese Hoffnung ist grundsätzlich unberechtigt, denn kontextfreie generative Kapazität garantiert nicht die Existenz eines effizienten Verarbeitungsverfahrens (Näheres hierzu folgt in Abschnitt 5).

[Evans 1987] verweist auf eine Anzahl erster, miteinander noch inkompatibler Ideen, die er als „frühe GPSG" kennzeichnet [Gazdar 1980, Gazdar 1981b, Gazdar 1981a, Gazdar/Sag 1981, Gazdar 1982, Gazdar *et al.* 1982]. Eine erste Formalisierung der gesamten Theorie, der „mittleren GPSG", wird in [Gazdar/Pullum 1982] beschrieben. Abschnitt 2.2 skizziert die wesentlichen Komponenten dieses Ansatzes. Die „späte GPSG" wird durch GKPS repräsentiert und von dem mittleren Ansatz in Abschnitt 2.3 abgegrenzt. Einen Ausblick auf die aktuellen, lexikon-zentrierten Entwicklungslinien gibt Abschnitt 2.4.

2.1 Zur Beschreibung natürlicher Sprachen mit kontextfreien Grammatiken

Daß natürliche Sprachen kontextfrei beschreibbar seien, widersprach der zu Anfang der GPSG-Entwicklung vorherrschenden Meinung, die sich auf die Behandlung einer Reihe sprachlicher Phänomene gründete wie z.B. Subjekt-Verb-Kongruenz, Konstruktionen mit *respectively*, englische Komparative, holländische infinitivische Verbalphrasen, *such that*-Konstruktionen usw. Alle damals bekannten Argumente dieser Art wurden in [Pullum/Gazdar 1982] diskutiert und entweder formal oder empirisch widerlegt. Damit war der Weg frei für die Verwendung kontextfreier Beschreibungsmodelle für größere Fragmente zahlreicher natürlicher Sprachen.

Das Ergebnis war provozierend genug, so daß die Diskussion fortgesetzt wurde, und nunmehr, da sich der Staub ein wenig gelegt hat, kristallisieren sich einige wenige Phänomene heraus, die tatsächlich nicht kontextfrei beschreibbar zu sein scheinen. Nach [Gazdar/Pullum 1985] gibt es für den Fall überkreuzweiser Abhängigkeiten bei infinitivischen VPs des Zürcher Dialektes einen überzeugenden Beweis [Shieber 1985].

Welche Folgerungen sind hieraus zu ziehen? Zunächst einmal ist GPSG offensichtlich zu schwach für eine Syntax des Zürich-Deutschen (und möglicherweise für die Beschreibung einer Reihe weiterer Phänomene, von denen noch präzise zu beweisen wäre, daß sie nicht kontextfrei beschreibbar sind). Dies bedeutet andererseits, daß von fast allen natürlichen Sprachen keine syntaktischen Phänomene bekannt sind, die nicht mit GPSG beschrieben werden könnten.

Die Frage steht jedoch im Räume, welche Klasse von Grammatiken denn gerade ausreicht, um natürliche Sprachen zu beschreiben. Die sogenannten schwach kontextsensitiven Grammatiken [Joshi 1986] erlauben die Behandlung der genannten überkreuzweisen Abhängigkeiten und zeichnen sich durch ein beschränktes Längenwachstum der Ableitungen aus. Sie haben jedoch nicht die Mächtigkeit kontextsensitiver Grammatiken. Formalismen für schwach kontextsensitive Grammatiken sind *Tree Adjoining Grammars* [Joshi *et al.* 1975, Joshi 1985] und *Head Grammars* [Pollard 1984].

Untersucht werden in diesem Hinblick auch *Indexed Grammars* [Aho 1968], die ebenfalls geringere Mächtigkeit als kontexsensitive Grammatiken haben, aber mächtiger sind als schwach kontextsensitiv (vgl. [Gazdar 1985]).

Es scheint mir, als werde der Frage, *ob* ein syntaktisches Phänomen in einem bestimmten Rahmen beschrieben werden kann, zu viel Bedeutung beigemessen und stattdessen die Frage vernachlässigt, *wie einfach* es darin modelliert werden kann.

Es ist ganz offensichtlich, daß die letzte Frage von viel entscheidenderer praktischer Bedeutung ist [Shieber 1988]. Die direkte Explizierbarkeit sprachlicher Fakten ist gerade ein Grund für die Entwicklung der Metasprache in GPSG.

2.2 Der indirekte, metagrammatische Ansatz

Die mittlere, metagrammatische GPSG definiert eine kontextfreie Grammatik, die wiederum—auf bekannte Weise—eine Menge von Endketten beschreibt. Diesen Ansatz möchte ich nach [Evans 1987] als *indirekt* bezeichnen. In der späten GPSG wird keine kontextfreie Grammatik definiert, sondern es werden Kriterien für die Zulässigkeit von Bäumen angegeben. Die Strukturbeschreibungen für Endketten werden somit *direkt* definiert.

In keinem Fall wird eine kontextfreie Grammatik zur Beschreibung natürlicher Sprachen direkt verwendet, denn die linguistische Aussagekraft solcher Grammatiken ist geringer als z.B. die transformationeller Ansätze, da eine Klassifikation von Teilmengen der Regeln zur Beschreibung sprachlicher Phänomene immer nur willkürlich erfolgen kann und nicht aus der Grammatik selbst hervorgeht.

Aus diesen Gründen war die frühe und mittlere GPSG als *metagrammatische Theorie* konzipiert, die auf einer abstrakten, metagrammatischen Ebene unterschiedliche Generalisierungen über Grammatikregeln explizit beschrieb. Von der metagrammatischen Ebene wurde eine Abbildung in Phrasenstruktur-Regeln (PS-Regeln) und damit in die *Objektgrammatik* definiert. Auf diese Weise stellte die (unverminderte) Komplexität der Objektgrammatik linguistisch kein Problem mehr dar, denn die interessierenden theoretischen Aussagen erfolgten auf der Metaebene.

2.2.1 Komplexe Kategorien, Extension und Unifikation

Eine wichtige Eigenschaft von GPS-Grammatiken ist die interne Struktur ihrer Kategorien, die sie von normalen kontextfreien Grammatiken, die monadische Kategoriensymbole haben, unterscheiden. Andere Komponenten (Regeln) können auf Teile dieser Struktur Bezug nehmen. Die Struktur ist aus Merkmal-Wert-Paaren aufgebaut, wobei Merkmalwerte wiederum komplexe Strukturen sein dürfen. Um die Menge der Kategorien endlich zu halten, darf ein Merkmalname niemals im zugehörigen Merkmalwert enthalten sein.

Kategorien können, müssen jedoch nicht vollständig spezifiziert sein, d.h. zu jedem Merkmal der Grammatik eine Spezifikation enthalten. Die Möglichkeit, aufgrund unterspezifizierter Kategorien Regeln zu definieren, stellt einen wesentlichen Vorteil komplexer Kategorien dar. Die monadischen Kategorien N, V, A und P lassen sich in Anlehnung an [Chomsky 1970] dekomponieren mithilfe zweier binärer Merkmale n und v. Auf diese Weise kann z.B. durch eine Kategorie, die nur (v : +) enthält, sowohl auf V als auch auf A Bezug genommen werden. Mehr noch, diese Kategorie umfaßt zusätzlich VP, AP und S; sie ist also stark unterspezifiziert. Für eine eindeutige Beschreibung einer der genannten Kategorien müßten weitere Merkmale spezifiziert werden. Die vorhegende Beziehung von wenig spezifizierten zu stärker spezifizierten Kategorien wird mithilfe des Begriffs der Extension definiert: Die *Extension* einer Kategorie *C* ist eine Kategorie, die mindestens alle Spezifikationen aus

	⟨n : +⟩	⟨n : −⟩
⟨v : +⟩	A	V
⟨v : −⟩	N	P

Abbildung 1: Merkmalspezifikationen zur Beschreibung von N, V, A und P

C enthält. Stärker spezifizierte Kategorien sind oft durch *Unifikation* gegeben: Die Unifikation entspricht (in der mittleren GPSG) der Vereinigung zweier Kategorien, wenn es zu diesen eine gemeinsame Extension gibt.

2.2.2 Feature Cooccurrence Restricions und Feature Coefficient Defaults

In GPSG können Aussagen über die innere Struktur von Kategorien formuliert werden. In Anlehnung an die Markiertheitskonventionen der Generativen Phonologie (Gazdar und Pullum zitieren [Chomsky/Halle 1968, Kap. 9]) werden Beschränkungen der in einer Kategorie zugelassenen Kombinationen von Merkmalspezifikationen definiert. Die *Feature. Cooccurrence Restrictions* (FCRs) zeichnen *legale* Kategorien aus. Nur legale Kategorien dürfen die Knoten im Phrasenstrukturbaum etikettieren.

Die Feature Coefficient Defaults, die in der späten GPSG *Feature Specification Defaults* (FSDs) genannt werden, verlangen das Auftreten bestimmter Merkmal-Wert-Paare, wenn das Merkmal nicht bereits aus anderen Gründen spezifiziert ist.

2.2.3 Das ID/LP-Format

Aus Beziehungen zwischen den komplexen Kategorien werden die Regeln gebildet. Anstelle von PS-Regeln, die sowohl Information über unmittelbare Dominanz (zwischen Mutter und Töchtern) als auch über lineare Präzedenz (zwischen Schwestern) enthalten, ermöglicht das *ID/LP-Format* (Immediate Dominance/Linear Precedence) die Trennung dieser Informationen und damit deren explizite Kennzeichnung in der Metagrammatik. *ID-Regeln* spezifizieren eine Mutterkategorie (linke Seite) und eine Multimenge von Tochterkategorien, über deren Abfolge in Phrasenstrukturbäumen sie nichts aussagen. *LP-Aussagen* legen Beschränkungen über die Reihenfolge von Schwester-Kategorien fest. Mithilfe einer LP-Aussage können Generalisierungen über die Konstituentenabfolge in unterschiedlichen ID-Regeln explizit gemacht werden, die in kontextfreien Grammatiken nicht ausgedrückt werden können. Ein praktischer Vorteil des ID/LP-Formats liegt in der Regelersparnis bei Sprachen mit relativ freier Wortstellung, für die einfach weniger Regeln anzugeben sind.

Die Verwendung des ID/LP-Formats macht eine wichtige Annahme über die Grammatiken, nämlich die der Konstanz der Konstituentenordnung (Exhaustive Constant Partial Ordering; kurz: ECPO-Eigenschaft). Wenn eine kontextfreie Grammatik *G* zwei PS-Regeln enthält, in denen zwei Töchter A und B auftreten, jedoch abhängig vom Vorkommen weiterer Töchter in unterschiedlicher Reihenfolge, so läßt sich zu *G* keine stark äquivalente ID/LP-Grammatik angeben.[5]

2.2.4 Lexikalische Subkategorisierung und Metaregeln

Davon ausgehend, daß Verben (und andere Lexeme) festlegen, welche syntaktischen Kategorien sie subkategorisieren, werden ID-Regeln, die präterminale Kategorien einführen, als Satzbaumuster verwendet, indem die vom Lexem subkategorisierten Kategorien als Schwestern der präterminalen Kategorie auftreten. Das Satzbaumuster ist durch eine Spezifikation des Merkmals subcat an der präterminalen Kategorie gekennzeichnet. Im Lexikon sind Kategorien, wenn sie andere Konstituenten subkategorisieren, ebenfalls für subcat spezifiziert. Man kann dies am besten als Zeiger auf die ID-Regeln verstehen, die die entsprechenden präterminale Kategorie, d.h. Klassen von Lexemen, einführen.

Ähnlich wie in GPSG Beziehungen zwischen Kategorien ausgedrückt werden können, lassen sich auch Relationen zwischen ID-Regeln explizit machen. Beispielsweise können passivierbare transitive Verben auch in dem entsprechende Satzbaumuster für Passivsätze auftreten; es besteht eine systematische strukturelle Beziehung zwischen den ID-Regeln, die Aktiv- und Passivkonfigurationen bilden. Diese Beziehungen werden mithilfe von *Metaregeln* beschrieben, die Funktionen von ID-Regeln in (Mengen von) ID-Regeln sind. Es wird sichergestellt, daß der Abschluß der ID-Regeln unter Metaregel-AnWendung endlich ist (vgl. Abschnitt 4.4).

2.2.5 X-Bar-Syntax und Head-Feature-Konvention

Kennzeichnend für GPS-Grammatiken ist die Verwendung einer X-Bar-Syntax [Chomsky 1970, Jackendoff 1977]. Die in [Gazdar/Pullum 1982] verwendete Version der X-Bar-Theorie geht von drei phrasalen Ebenen aus, die durch numerische Werte des Merkmals bar gekennzeichnet werden (0, l, 2).

Innerhalb einer ID-Regel ist es möglich (und gewöhnlich der Fall), daß die Mutter und eine oder mehrere Töchter dieselben Spezifikationen für n und v haben. Diejenige unter diesen Töchtern mit dem kleinsten bar-Wert, der kleiner oder gleich dem bar-Wert der Mutter ist, wird üblicherweise *Head* genannt.

Diese Auszeichnung vor anderen Töchtern ist nützlich, da Mutter und Head (unabhängig von ihrer jeweiligen Spezifikation für n und v) viele weitere Eigenschaften teilen. In GPSG drückt sich dies in der gleichen Spezifikation weiterer Merkmale an Mutter und Head aus.

Diese Kospezifikation wird auf der metagrammatischen Ebene durch das Merkmalinstantiierungsprinzip *Head Feature Convention* (HFC, [Gazdar *et al.* 1982]) erreicht. Eine ID-Regel genügt der HFC, wenn ihre Mutter und ihr Head bezüglich einer Menge von Head-Merkmalen übereinstimmen.

[5]Siehe Abschnitt 4.3 für weitere Diskussion.

Durch HFC als universelles Prinzip wird eine weitere Generalisierung erreicht, die mit PS-Regeln nicht darstellbar ist.

2.2.6 Das Control-Agreement-Prinzip

Auf der Grundlage des *Control-Agreement-Prinzips* (CAP) wird angestrebt, die Kongruenzbeziehungen in natürlichen Sprachen zu erklären. Kongruenz läßt sich, übereinzelsprachlich gesehen, nicht an bestimmten syntaktischen Kategorien oder Merkmalen festmachen, sondern muß unter Berücksichtigung semantischer Eigenschaften beschrieben und erklärt werden. Ausgehend von einem oberflächennahen Semantikkonzept mit Funktor-Argument-Beziehungen postulieren Gazdar und Pullum

- Funktoren können mit nominalen Argumenten kongruieren

und berufen sich dabei auf Edward Keenan [Keenan 1974]. Welche Funktoren tatsächlich an Kongruenzrelationen beteiligt sind, variiert von Sprache zu Sprache.

Der GPSG-Ansatz verallgemeinert Keenans semantisch motiviertes Prinzip und folgt Bach und Partee [Bach/Partee 1980], indem anstelle der Explizierung der Funktor-Argument-Verhältnisse der Begriff der *Kontrolle* verwendet wird. Eine Argument-Kategorie kann die Funktor-Kategorie kontrollieren, oder die Funktor-Kategorie kann die Kontrollbeziehung zwischen zwei Argument-Kategorien vermitteln. Wann eine Kontrollbeziehung tatsächlich vorhegt, muß durch die Grammatikschreiberin festgelegt werden. Das CAP wird als Bedingung über ID-Regeln definiert: Eine ID-Regel genügt dem CAP, wenn Kategorien, zwischen denen eine Kontrollbeziehung besteht, bezüglich einer Menge von Kontrollmerkmalen übereinstimmen.

Unter der Voraussetzung, daß Kontroll- und Head-Merkmale geeignet definiert sind, bewirken CAP und HFC gemeinsam Kongruenz in komplexen Phrasenstrukturbäumen. Gazdar und Pullum betonen, daß eine solche Beschreibung von Kongruenzverhältnissen nicht auf einzelsprachliche Regularitäten zurückgreift:

> The facts simply follow from interaction of two universal principles (the HFC and the CAP) with the form of the syntactic and semantic rules which are motivated quite independently of the facts of agreement. [Gazdar/Pullum 1982, S. 32f]

2.2.7 Das Foot-Feature-Prinzip

Eine weitere Gruppe von syntaktischen Phänomenen läßt sich auf der Ebene der Metagrammatik in Form eines dritten universellen Instantiierungsprinzips erfassen. Diese Phänomene betreffen, grob gesagt, Beziehungen zwischen unbestimmt weit voneinander entfernt stehenden Konstituenten eines Phrasenstrukturbaumes. Dazu zählen Topikalisierung[6], Fragewörter mit Konstituentenstatus, „Missing-Objecf-Konstruktionen usw.

[6]In dieser Arbeit wird der Begriff Topikalisierung nur in seiner technischen Lesart (d.h. Positionierung einer Konstituente aus ihrer kanonischen Stellung heraus an den Satzanfang) verwendet, nicht in der pragmatischen Bedeutung. Davon unberührt bleibt die Tatsache, daß die pragmatisch motivierte Kennzeichnung eines Satzelements als „Topic" zu einer Topikalisierung fuhren kann.

Kennzeichnend für sie ist, daß sie nicht durch HFC erfaßt werden, da sie nicht nur Beziehungen zwischen Mutter und Head ausdrücken, sondern auch zwischen der Mutter und anderen Töchtern. Die Beziehung wird durch Gleichheit von Merkmalspezifikationen ausgedrückt. Die betroffenen Merkmale heißen Foot-Merkmale.

Die grundlegende Idee, die Beziehung auf prinzipielle Weise zu beschreiben, besteht darin, die beiden betreffenden Kategorien in den ID-Regeln auszuzeichnen, indem sie eine Spezifikation eines Foot-Merkmals erhalten. Im Phrasenstrukturbaum existiert ein Pfad zwischen ihnen mit unbestimmt vielen Knoten. Bezüglich der sie etikettierenden Kategorien ist über die Werte des Foot-Merkmals in den jeweils zugrunde liegenden ID-Regeln nichts ausgesagt. Die Beziehung zwischen den beiden ausgezeichneten Knoten wird nun dadurch hergestellt, daß die Foot-Merkmale an den Kategorien der Knoten auf dem Pfad gleiche Werte haben. Dies wird mithilfe des *Foot-Feature-Prinzips* (FFP) sichergestellt, das wie HFC und CAP über ID-Regeln operiert.

Eine ID-Regel genügt dem FFP, wenn die Mutter bezüglich einer Menge von Foot-Merkmalen mit den Töchtern übereinstimmt, die Knoten auf dem Pfad etikettieren können.

Dies schließt den Überblick über die metagrammatische Ebene der mittleren GPSG ab. Im folgenden wird das intendierte Zusammenwirken der einzelnen Komponenten skizziert.

2.2.8 Die Objektgrammatik

Die Definitionen auf der metagrammatischen Ebene sollen gemeinsam eine Menge von PS-Regehi erzeugen. Daher dürfen sie einander nicht grundsätzlich widersprechen, indem sie die Erzeugung von PS-Regeln verhindern. Natürlich ist es einfach, etwa durch FCRs Merkmale so zu instantiieren, daß die Merkmalinstantiierungsprinzipien (MIPs) nicht erfüllt werden. Ein solches Vorgehen bei der Formulierung einer Metagrammatik widerspräche aber den linguistischen Motivationen des Formalismus. Der Grundgedanke, durch das Zusammenwirken der Komponenten bestimmte Merkmalspezifikationen in Kategorien zu erzwingen und die anderen Merkmale *frei instantiierbar* zu lassen, zieht sich durch alle Theorien, die komplexe Kategorien verwenden.

Die Frage bleibt letztlich offen, ob das Zusammenwirken der Komponenten des GPSG-Formalismus die Formulierung eines Regelsystems für ein Fragment einer beliebigen natürlichen Sprache ermöglicht. GKPS verwenden viel Mühe darauf, die wechselseitigen Einflüsse der Komponenten formal zu beschreiben (vgl. Abschnitt 4).

Um das intuitive Konzept der FCDs zu verwirklichen, nämlich die Instantiierung von Merkmalen nur dann zu fordern, wenn diese ansonsten frei instantiierbar wären, beschränken Gazdar und Pullum den Anwendungsbereich der FCDs entsprechend. Die Merkmale, die FCDs nicht betreffen, heißen *privilegiert.* Nun wird in zwei Schritten die Beziehung zwischen Metagrammatik und Objektgrammatik definiert (vgl. Abbildung 2). Der erste Schritt beschreibt die Auswirkungen von FCRs, MIPs und FCDs auf ID-Regeln. Der zweite Schritt behandelt das Zusammenwirken von Metaregeln, ID-Regeln und LP-Aussagen.

1. Eine *instantiierte Extension* einer ID-Regel ist eine ID-Regel, in der jede Ka-

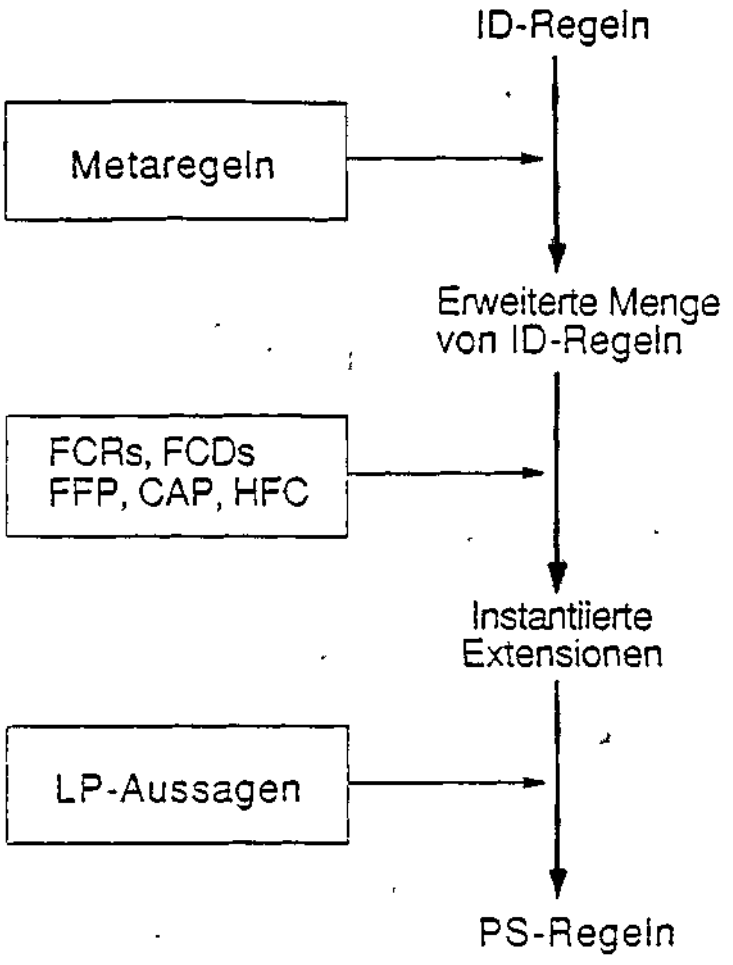

Abbildung 2: Die Organisation der mittleren GPSG

tegorie legal ist, die der HFC, dem GAP und dem FFP genügt und bei der in jeder Kategorie alle nicht privilegierten Merkmale den FCDs genügen.

2. Aus dem endlichen Abschluß der ID-Regeln unter Metaregel-Anwendung resultiert aufgrund des ersten Schrittes eine Menge instantiierter Extensionen. Für jede instantiierte Extension entsprechen die Mutter und diejenigen Permutationen der Töchter, die allen LP-Aussagen genügen, einer Menge von PS-Regeln. Die Vereinigung dieser Mengen bildet eine zur Metagrammatik stark äquivalente Objektgrammatik.

Dieser Korrespondenz liegen eine Reihe von Annahmen über Eigenschaften natürlicher Sprachen zugrunde. In [Gazdar/Pullum 1982] wird genannt:

- Metaregeln können sich nur auf Spezifikationen beziehen, die ursprünglich in ID-Regehi kodiert wurden;
- Metaregeln oder Instantiierungsprinzipien können sich nicht auf Ordnungsrelationen zwischen Töchtern beziehen;
- Ordnungsrelationen können von Merkmalen bestimmt werden, die durch andere Komponenten der Metagrammatik instantiiert werden;

- Ordnungsrelationen treten nur zwischen Schwesterkategorien auf.

2.3 Der direkte, constraint-basierte Ansatz

Bei dem direkten Ansatz der späten GPSG wird zwar keine kontextfreie Grammatik mehr definiert, doch geht der unmittelbare Bezug zu PS-Regeln nicht verloren. Er ist über die gemäß den Komponenten des Formalismus zulässigen lokalen Bäumen (d.h. Bäume der Tiefe 1) gegeben, die aufgrund einer kontextfreien Grammatik erzeugt werden könnten.

2.3.1 GPSG als constraint-basierte Theorie

In der späten GPSG operieren sämtliche Komponenten des Formalismus über lokalen Bäumen statt über ID-Regeln. Im Mittelpunkt steht daher die Frage, unter welchen Bedingungen ein lokaler Baum zulässig ist. GKPS beantworten diese Frage auf strikt deskriptive Weise, in dem sie eine Menge von Eigenschaften formulieren, die zulässige lokale Bäume charakterisieren. Der Formalismus von GKPS enthält keinerlei prozedurale Information; z.B. gibt er kein Verfahren an, wie man ausgehend von einer ID-Regel einen lokalen Baum so verändert (beispielsweise durch Instantiierung von Merkmalen in Kategorien, die seine Knoten etikettieren), daß er zulässig wird.

Es muß betont werden, daß auch die mittlere GPSG die Kriterien für instantiierte Extensionen von ID-Regeln als Menge von Bedingungen beschreibt; allerdings trägt der zweite Schritt bei der Definition der Objektgrammatik doch stark prozedurale Züge.

Die deskriptive Vorgehensweise folgt aus einigen methodologischen Grundannahmen (vgl. [Gazdar *et al.* 1985, S. 2]). Ausgehend von den Voraussetzungen des generativen Paradigmas der Linguistik sehen GKPS in der Grammatik ein formales System, das genau die zu einer Sprache gehörenden Ausdrücke beschreibt und jedem eine Struktur und eine Interpretation zuweist. . -

- Um einen Aspekt der Organisation natürlicher Sprachen zu *erklären,* müssen Beschreibungen der relevanten Phänomene vorliegen, die einen postulierten Sachverhalt stützen.

- Der Grammatikformalismus kann und soll als formale Sprache aufgebaut sein, in der eine bestimmte Art von Grammatiken formulierbar ist.

- Universelle Eigenschaften natürlicher Sprachen sind idealerweise *Konsequenzen* aus Eigenschaften der Theorie und nicht explizit in der Theorie ausgedrückt.

Die Syntax und Semantik natürlicher Sprachen soll durch den Formalismus beschrieben (und, wenn möglich, erklärt) werden. Der Formalismus selbst hat als formale Sprache eine Syntax und eine Semantik und stellt eine Metasprache dar, die Endketten, Bäume und Kategorien der Objektsprache beschreibt.

(3) S —> NP, VP

Dies mag das folgende Beispiel verdeutlichen: Ein Ausdruck wie (3) folgt der Syntax der Metasprache, die festlegt, daß links vom Pfeil eine komplexe Kategorie stehen darf und rechts vom Pfeil eine Folge komplexer Kategorien, die durch Kommas getrennt sind (die Symbole S, NP und VP in (3) sind lediglich Abkürzungen). Die Syntax der komplexen Kategorien ist ebenfalls Teil der Metasprache. Der Ausdruck kann gemäß der Semantik der Metasprache als ID-Regel interpretiert werden (etwa aufgrund einer Formalisierung der im vorigen Abschnitt skizzierten Intuition). Dann beschreibt er Dominanzbeziehungen in der Syntax einer Objektsprache (z.B. Englisch).

Die Erklärung linguistischer Universalien ist ein sehr hoch gestecktes Ziel. Selbst wenn man glaubt, eine Eigenschaft aller natürlicher Sprachen gefunden zu haben und man diese formal repräsentiert, ist sie damit noch nicht erklärt. Man kann dann, im Falle eines Gegenbeispiels, die formale Repräsentation in geeigneter Weise verändern ohne weitere Auswirkung auf den Rest des Formalismus. Nach GKPS sollte eine Erklärung auf der umgekehrten Denkweise beruhen:

> The explanatory task has not even begun when a constraint or a generalization is merely stated. Only when it can be shown to be a nontrivial consequence of the defmition of the notion 'possible grammar' can it be regarded äs explained [...] [Gazdar *et al.* 1985, S. 3]

In dem Fall, da sich eine solchermaßen postulierte Behauptung aufgrund sprachlicher Daten als unzutreffend erweist, hätte dies schwerwiegende Konsequenzen für den gesamten Formalismus.

Ein Beispiel ist die Annahme der ECPO-Eigenschaft für Grammatiken natürlicher Sprachen; also die Behauptung, daß die Anordnung von Schwesterkonstituenten nicht vom strukturellen Kontext abhängt, d.h. es gibt keine Notwendigkeit für eine ID/LP-Syntax, in der z.B. nach einem V die Abfolge VP vor NP gefordert ist, jedoch nach einer von V verschiedenen Konstituente eine NP *vor* einer VP stehen muß. Erwiese sich diese Behauptung als falsch, so wäre das ID/LP-Format nicht adäquat zur Beschreibung von Dominanz- und Präzedenzrelationen. Die Annahme hat weitreichende Konsequenzen für die Formulierung von Grammatiken, die in Abschnitt 4.3 erörtert werden.

Ein anderes Beispiel betrifft den Status bestimmter Merkmale. Die HFC macht nur Sinn im Zusammenhang mit einer X-Bar-Syntax, und infolgedessen ist das Merkmal bar Bestandteil jeder GPS-Grammatik. Die Präsenz anderer Merkmale wie n, v oder cas (Kasus) ist nicht durch die Theorie gefordert.

Dieser prinzipielle Anspruch, Universalien zu identifizieren, läßt sich leicht kritisieren. Hendriks [Hendriks 1986, S. 14ff] bemerkt, daß eine Universalie, die aus einer Metasprache *MI* als Theorem folgt, nicht dadurch „uninteressanter" wird, daß man sie in einer (reichhaltigeren) Metasprache M_2 explizit kodieren kann.

Gegen die Nützlichkeit des Konzepts sprachlicher Universalien wird oft vorgebracht, daß es empirisch unmöglich sei, nachzuweisen, daß zu einer als sprachlichen Universalie deklarierten Behauptung keine Gegenbeispiele existieren. Aber gerade daraus folgt, quasi als Metauniversalie, daß beim gegenwärtigen Stand der linguistischen Forschung Theorien notwendig tentativ sein müssen (was teilweise erklärt, warum es so viele verschiedene Theorien gibt, über deren Erklärungsfähigkeit der-

zeit kein abschließendes Urteil möglich ist). Angesichts dessen erscheint der Ansatz von GKPS, als *Versuch* interpretiert, methodisch vielversprechend; als *endgültige Aussage* über natürliche Sprachen jedoch vermessen.

Da sich GKPS in ihrem Buch nicht mit psychologischen Fragestellungen wie Spracherwerb und menschlicher Sprachverarbeitung beschäftigen, ergibt sich im Hinblick auf das Ziel einer rigorosen Formalisierung der Metasprache kein Anlaß, prozedurale (und damit zusätzlich zu motivierende) Elemente einzubeziehen. Die Metasprache wurde stattdessen als eine Menge von Axiomen formuliert, aus denen die Bedingungen für zulässige Strukturen als Theoreme resultieren. Diese Vorgehensweise wird gelegentlich auch als *axiomatisch, deskriptiv* oder *constraint-basiert* bezeichnet. Für die vorliegende Diskussion spielen die in diesen Attributen ausgedrückten Unterschiede in der Perspektive keine Rolle.

2.3.2 Zum Verhältnis zwischen Syntax und Semantik

Im Unterschied zur mittleren GPSG ist die Kopplung zwischen Syntax und Semantik in der späten GPSG ein eigenes Thema. GKPS präsentieren eine monostratale Theorie, deren Repräsentationsebene die der syntaktischen Strukturen ist. Die semantische Interpretation erfolgt indirekt im Stile Montagues, indem einer syntaktischen Struktur ein Ausdruck in intensionaler Logik (IL) zugewiesen wird. Die IL-Ausdrücke sind nicht Teil der linguistischen Beschreibung, sondern könnten auch durch eine direkte Interpretation einer Endkette in einer Modelltheorie ersetzt werden. Infolgedessen ist nicht beabsichtigt, IL-Ausdrücke als semantische Filter zu verwenden, die über die Wohlgeformtheit einer syntaktischen Struktur entscheiden.

Der Begriff der Wohlgeformtheit (hier: Zulässigkeit) ist im wesentlichen syntaktisch begründet.[7] Daher werden Sätzen, die unter allen Interpretationen (die ich mir vorstellen kann) semantisch leer sind—etwa *Farblose grüne Ideen schlafen wütend*—durchaus zulässige Strukturen assoziiert.

2.3.3 Von der mittleren zur späten GPSG

Die *Metagrammatik* der mittleren GPSG und die *Metasprache* der späten GPSG spezifizieren beide das bereits eingeführte Instrumentarium (Metaregeln, ID/LP-Format, FCR, FSD, HFC, CAP, FFP, usw.), jedoch aus verschiedenen Sichtweisen heraus. Man kann sich auf den Standpunkt stellen, daß der Unterschied künstlich sei (in [Hukari/Levine 1986] etwa wird weiterhin von Metagrammatik gesprochen), denn aus der Menge der zulässigen lokalen Bäume, die die Metasprache charakterisiert, läßt sich (durch geeignete Umbenennungen der Knotenetiketten) eine Menge von PS-Regeln gewinnen, die man als Objektgrammatik auffaßt. Gegen diesen Standpunkt ist einzuwenden, daß dieser Schritt von GKPS nicht intendiert ist, geschweige denn im Formalismus vollzogen wird.[8] Der Unterschied läßt sich folgendermaßen charakterisieren.

[7]Die Verwendung semantischer Typen bei der Etablierung von Kongruenzrelationen (vgl. Abschnitt 4.7) eröffnet den Zugang zu Information über Funktor-Argument-Verhältnisse.

[8]Die einzige metagrammatische Komponente der späten GPSG sind die Metaregeln; sie spezifizieren ID-Regeln und tragen somit nicht direkt zum Begriff des zulässigen Baumes bei.

- Die Metagrammatik stellt ein System von aufeinander abgestimmten Komponenten dar, das eine kontextfreie Grammatik charakterisiert, indem es linguistische Generalisierungen explizit ausdrückt.

- Die Metasprache stellt eine formale Sprache mit separater Syntax und Semantik zur Beschreibung linguistischer Generalisierungen dar. Ihre generative Kapazität entspricht der kontextfreier Grammatiken.

Im folgenden wenden wir uns den inhaltlichen Unterschieden zwischen den Komponenten der mittleren und der späten GPSG zu.

Metaregeln sind in ihrer Bedeutung eingeschränkt worden. Waren sie in der frühen GPSG u.a. für Generalisierungen über Arten des Merkmaltransports verwendet worden, so operieren sie nunmehr lediglich über lexikalischen ID-Regeln, d.h. über solchen, die Kategorien mit einer Spezifikation für subcat enthalten.

ID-Regeln dienen in der späten GPSG dazu, lokale Bäume zu projizieren. Die Menge der lokalen Bäume muß auf geeignete Weise auf die zulässigen lokalen Bäume eingeschränkt werden. Dazu dienen FCRs, FSDs, LP-Aussagen und die MIPs. Nur die zulässigen lokalen Bäume sind zu den zulässigen syntaktischen Strukturen kombinierbar.

Die HFC und das CAP haben wesentliche Änderungen erfahren, die ihre Definitionen (und infolgedessen auch die der FSDs) dramatisch verkomplizieren. Zwei Hauptursachen hegen in dem Versuch, die Koordinationsphänomene in natürlichen Sprachen zu erklären und in dem Anspruch, die Kontrollbeziehungen zwischen Konstituenten aufgrund eines Systems semantischer Typen zu beschreiben. Die HFC übernimmt zusätzlich die Aufgabe, bestimmte Merkmale aus verschiedenen Teilen koordinierter Strukturen zusammenzuführen. Infolgedessen kann mehr als eine Head-Tochter im lokalen Baum vorkommen. Das CAP basiert auf semantischen Typen, zu deren Definition auch Merkmalspezifikationen beitragen können.

Kompliziert ist die Interaktion der MIPs. Zwei Andeutungen sollen hier genügen: Zum einen wird, hauptsächlich zur Beschreibung bestimmter Phänomene mit parasitären Lücken, das Foot-Merkmal slash auch als Head-Merkmal definiert. Wie in Abschnitt 4.7 begründet wird, ist slash zugleich auch ein mögliches Kontrollmerkmal. Zum ändern basiert HFC—per Definition—auf der Wirkung von FFP und CAP. CAP wiederum benötigt Spezifikationen von Head-Merkmalen zur Bestimmung der semantischen Typen. Es wird eingehend zu untersuchen sein, welche Konsequenzen aus einer möglichen wechselseitigen logischen Abhängigkeit zwischen den MIPs hervorgehen (vgl. Abschnitt 5).

Das Zusammenwirken der einzelnen Komponenten wird definiert durch den Kernbegriff der Zulässigkeit von Bäumen, der an die Stelle des Verfahrens zur Erzeugung einer kontextfreien Grammatik tritt (vgl. Abbildung 3). Die genannten, mit jenem Verfahren verbundenen Annahmen über Eigenschaften natürlicher Sprachen resultieren ebenfalls aus der Metasprache.

2.4 GPSG und das Lexikon

Ein chronisch defizitärer Gegenstandsbereich in allen Versionen der GPSG ist die Organisation des Lexikons. Während in anderen unifikationsbasierten Grammatiktheo-

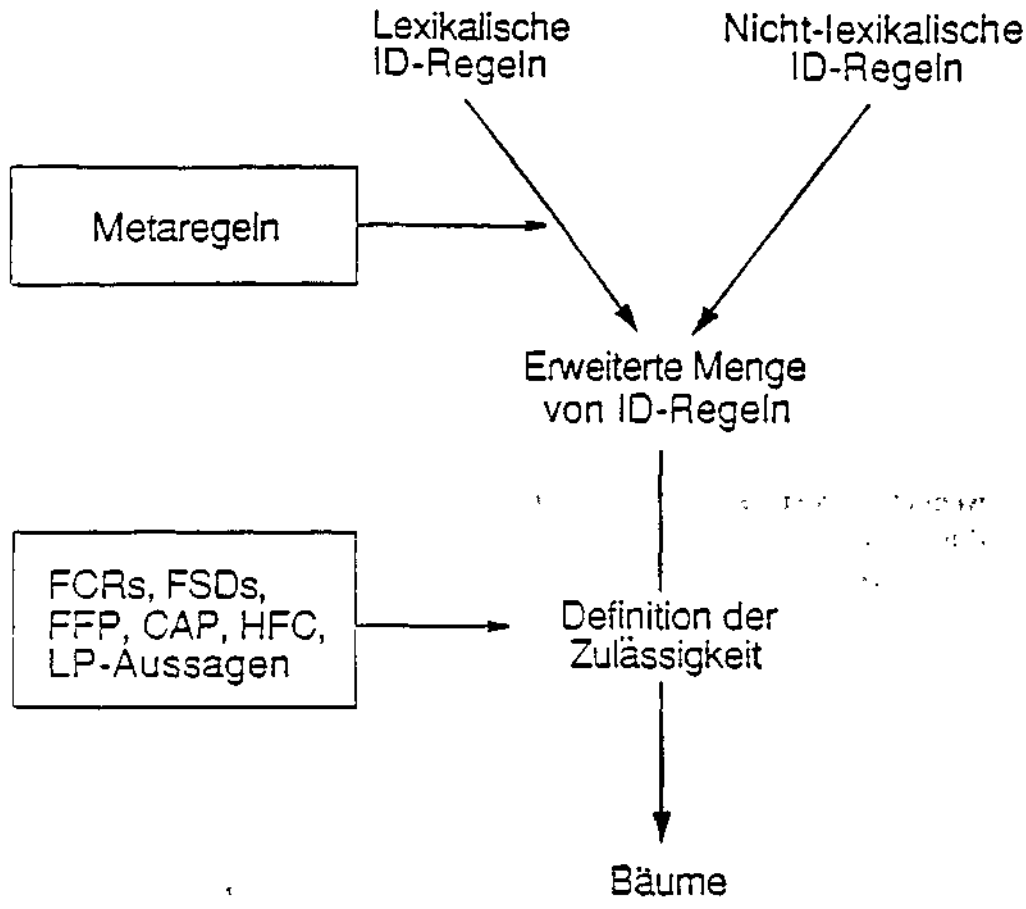

Abbildung 3: Die Organisation der späten GPSG

rien (z.B. LFG und vor allem CUG) die strikte Trennung zwischen der Syntax und dem Lexikon aufgehoben ist und das Lexikon eine zentrale Rolle für die jeweiligen linguistischen Aussagen spielt, gehen in GPSG nur wenige Annahmen über den Aufbau und den Inhalt des Lexikons in die Metasprache ein: die durch das Lexikon zugelassenen terminalen Bäume besitzen als Mutter eine lexikalische Kategorie und als einzige Tochter ein Terminalsymbol. Sie werden grundsätzlich als zulässig betrachtet.

Lexikalische Kategorien enthalten eine (numerische) Spezifikation für subcat. GKPS führen für ihr Fragment des Englischen 48 verschiedene Klassen ein und benötigen ebensoviele ID-Regeln, um die entsprechenden Heads einzuführen. Insgesamt enthält ihr Fragment 58 ID-Regeln, d.h. zehn Regeln haben einen nichtlexikalischen Head.

In Weiterentwicklungen wie *Japanese Phrase Structure Grammar* (JPSG, [Gunji 1987]) und *Head-Driven Phrase Structure Grammar* (HPSG, [Pollard/Sag 1987]) ist die Anzahl der notwendigen ID-Regeln drastisch reduziert. Gunji kommt zur Beschreibung seines Fragments des Japanischen mit einer einzigen Regel aus, die besagt, daß eine Mutter einen Head und eine weitere Tochter dominiert. Pollard und Sag geben für die nicht invertierten Head-Komplement-Strukturen des Englischen zwei Regehl an. Die erste expandiert gesättigte Konstituenten in einen nichtlexikalischen Head und eine weitere Tochter. Die zweite expandiert ungesättigte

phrasale Kategorien in einen lexikalischen Head und eine Anzahl weiterer Töchter, über die nichts ausgesagt wird. Eine dritte Regel beschreibt Inversion und eine letzte die Einführung von Adjunkten.

Der Begriff der Sättigung ist eng mit einem von GPSG abweichenden Konzept der Subkategorisierung verbunden. Während in GPSG Subkategorisierung nur für lexikalische Kategorien definiert ist, können in JPSG und in HPSG auch phrasale Kategorien Information über Subkategorisierung ausdrücken. Das Merkmal subcat hat als Wert eine Menge von Kategorien[9], die gemäß dem Subkategorisierungsprinzip bestimmt wird. Dies besagt im wesentlichen, daß der subcat-Wert der Mutter gleich dem des Heads ist abzüglich denjenigen Elementen, die mit weiteren Komplement-Töchtern unifizieren. Gesättigt heißen dann Kategorien, deren subcat-Wert die leere Menge ist (z.B. S, NP). Ungesättigt heißen alle anderen Kategorien, die für subcat spezifiziert sind (z.B. VP mit (subcat : {NP})).

Das Konzept der Subkategorisierung in HPSG ermöglicht eine flexible Verteilung von Information auf universelle Prinzipien und idiosynkratische Lexikoneinträge (an den lexikalischen Kategorien enthält der Wert von subcat alle Komplemente). Dies macht die Einführung von unterschiedlichen Subkategorisierungen vermittels ID-Regeln überflüssig.

Der Trend, mehr Information im Lexikon zu kodieren und weniger in Syntax-Regeln geht einher mit einer geeigneten Strukturierung des lexikalischen Wissens. Pollard und Sag schlagen eine hierarchische Ordnung von *lexikalischen Typen* vor, die geeignet ist, redundante Kodierung von Information zu vermeiden [Pollard/Sag 1987, S. 191ff]. Die Beziehung zwischen Wortstämmen und flektierten Formen soll mithilfe von *lexikalischen Regeln*[10] ausgedrückt werden.

Zur Repräsentation strukturierter Lexika bieten sich auch andere Formalismen mit einer Subsumptionshierarchie an wie z.B. KL-ONE-artige Systeme [Brachman/Schmölze 1985]. Die Semantik der Subsumption in solchen Systemen ist eng verwandt mit der Semantik der Extension und Unifikation in GPSG und Nachfolgern (vgl. zu LFG und KL-ONE [Nebel/Smolka 1989]).

Diese linguistischen Weiterentwicklungen und die Perspektiven, die sie eröffnen, prägen die gegenwärtige linguistische Forschung im Bereich der Unifikationsgrammatiken und beeinflussen zunehmend auch computerlinguistische Arbeiten (vgl. die Diskussion der Lexikalisierung anhand graphentheoretischer Beschreibungsmittel in [Uszkoreit 1986b]).

In dieser Arbeit spielt das Lexikon nur eine Nebenrolle, was nicht zuletzt darauf zurückzuführen ist, daß ich versuche, möglichst nahe an den Vorgaben der GPSG-Theorie zu bleiben.

[9] In HPSG handelt es sich um eine Liste, da die Ordnung der gespeicherten Kategorien wesentlich ist.

[10] Diese sind nicht zu verwechseln mit lexikalischen ID-Regeln (die einen lexikalischen Head einführen).

3 Eine formale Definition von GPSG

Ziel dieses Abschnitts ist es, anhand einer formalen Definition von GPSG einen Überblick über wichtige, in vielen GPSG-Formalismen verwendete Komponenten der Metasprache zu geben. Die linguistische Motivationen der Komponenten werden in den nachfolgenden Abschnitten dargestellt.

Es wird der Formalismus von GKPS zugrunde gelegt, wobei viele Komponenten anders (wie ich glaube, präziser und vollständiger) formalisiert werden. Hinsichtlich der MIPs kann an dieser Stelle jedoch nur ein ungefährer Eindruck vermittelt werden. Sie werden im Zusammenhang mit einer ausführlichen linguistischen Motivation in den Abschnitten 4.6 bis 4.7 definiert. An dieser Stelle werden die MIPs zusammengefaßt als Prinzipien der Kookkurrenz von Merkmalspezifikationen bei direkten Ableitungsschritten.

Die Definitionen basieren auf den folgenden mathematischen Basiskonstrukten: Alphabet, Wort, Menge, Potenzmenge (geschrieben *P(M)),* Relation, Funktion und Permutation. Darüber hinaus wird der Begriff der Multimenge benötigt.

Definition 1. Eine *Multimenge* ist eine Menge, in der Elemente mehrfach auftreten können. Sei *M* eine Menge und M_{mul} eine Multimenge. Dann heißt *M Basis von* M_{mul}, gdw. jedes Element von *M* mindestens einmal in M_{mul} vorkommt. $'P_{mul}(M)$ ist die Menge aller Multimengen auf der Basis von *M.*

Ein grundlegender Baustein für die Komponenten der Metasprache ist die *komplexe Kategorie.* Eine komplexe Kategorie ist eine Menge von *Merkmalspezifikationen.* Eine Merkmalspezifikation ist ein geordnetes Paar bestehend aus einem *Merkmalnamen* und einem *Merkmalwert.* Ein Merkmalwert kann entweder atomar sein oder eine komplexe Kategorie. Komplexe Kategorien sind partielle Funktionen von Merkmalnamen in Merkmalwerte. In dem Wert w einer kategorienwertigen Merkmalspezifikation (m : w) und in jeder tiefer in w eingebetteten kategorienwertigen Merkmalspezifikation tritt eine Spezifikation (m : w') nicht auf.

Damit ist die Einbettungstiefe durch die Zahl der kategorienwertigen Merkmale beschränkt und die Anzahl der Kategorien endlich. Dies ist eine notwendige Voraussetzung für die Kontextfreiheit von GPSG.

Zunächst werden die Grundbestandteile einer Kategorie definiert, nämlich Merkmalnamen, atomare Merkmalwerte und atomwertige Merkmalspezifikationen. Eine Definition von kategorienartigen Merkmalwerten erfordert die Definition der Kategorie selbst; daher wird zunächst als Hilfskonstrukt die Menge der mengenartigen Merkmalwerte eingeführt, über die anschließend bestimmte Restriktionen definiert werden.

Definition 2. Sei *M* eine endliche Menge von *Merkmalnamen* und *W* eine endliche Menge von *atomaren Merkmalwerten.* Seien M_{atom} und M_{cat} Mengen von Merkmalnamen mit $M = M_{atom}$ U M_{cat} und M_{atom} H $M_{cat} = 0$.

Ein geordnetes Paar (m : w) mit m € M_{atom} und w € *W* heißt *atomwertige Merkmalspezifikation.* S_{atom} ist die Menge der atomwertigen Merkmalspezifikationen.

Sei $MW = S_{atom} \cup \{(\mathsf{m} : \mathsf{w}) | \mathsf{m} \in M_{cat}, \mathsf{w} \in \mathcal{P}(MW)\}$ die Menge der mengenartigen Merkmalwerte.

Die Elemente von MW sind entweder atomwertige Merkmalspezifikationen oder Merkmalspezifikationen, deren Werte Elemente aus MW sind. Offensichtlich ist die Kardinalität von MW unendlich.

MW wird von dem Prädikat τ benutzt, das erfüllt ist, wenn ein Merkmalname auf derselben oder einer tieferen Einbettungsebene erneut auftritt. Genau dann verletzt der mengenartige Merkmalwert die Bedingungen für eine komplexe Kategorie. Daher werden komplexe Kategorien als diejenigen mengenartigen Merkmalwerte definiert, für die τ *nicht* zutrifft.

Definition 3. Sei $\tau : M_{cat} \times MW \mapsto \{\text{true}, \text{false}\}$ ein Boole'sches Prädikat mit

$$\tau(\mathsf{m}, \mathsf{w}) = \text{true} \leftrightarrow [\exists w' \subset MW : (\mathsf{m} : \mathsf{w}') \in \mathsf{w}] \vee [\exists (\mathsf{m}' : \mathsf{w}') \in \mathsf{w} : \tau(\mathsf{m}, \mathsf{w}')]$$

Sei $C \subset MW$. C heißt *komplexe Kategorie* gdw. gilt:

- $\forall\ (\mathsf{m} : \mathsf{w}), (\mathsf{m}' : \mathsf{w}') \in C : \mathsf{m} = \mathsf{m}' \supset \mathsf{w} = \mathsf{w}'$
- $\forall\ (\mathsf{m} : \mathsf{w}) \in C$ mit $\mathsf{m} \in M_{cat}$: w ist komplexe Kategorie, und $\neg\tau(\mathsf{m}, \mathsf{w})$.

Für GPSG ist die Extensionsrelation zwischen zwei Kategorien zentral. Eine Kategorie ist eine Extension einer anderen, wenn sie mindestens dieselben Merkmalspezifikationen enthält wie diese.

Definition 4. Seien C_i, C_j komplexe Kategorien. Dann heißt C_j *Extension* von C_i ($C_i \sqsubseteq C_j$) gdw. gilt:

- $\forall (\mathsf{m} : \mathsf{w}) \in C_i, \mathsf{m} \in M_{atom} : (\mathsf{m} : \mathsf{w}) \in C_j$
- $\forall (\mathsf{m} : \mathsf{w}) \in C_i, \mathsf{m} \in M_{cat} : (\mathsf{m} : \mathsf{w}') \in C_j$ mit $\mathsf{w} \sqsubseteq \mathsf{w}'$

Mithilfe der Extension wird nun die Unifikation definiert. Für Teilmengen von Kategorien, die einen Halbverband mit Supremum bilden, ist die *Unifikation* die kleinste obere Schranke; d.h. sie ist die „∪-Operation" des Halbverbandes.

Definition 5. Sei K eine Menge von komplexen Kategorien und $C \in K$. Dann ist C eine *obere Schranke* von K gdw. für alle $C' \in K$ gilt: $C' \sqsubseteq C$.

C ist die *Unifikation* von K (falls $K = \{C_1, C_2\}$, so schreiben wir $C_1 \sqcup C_2$) gdw. C obere Schranke von K ist und für alle oberen Schranken C' von K gilt: $C \sqsubseteq C'$.

Nach Einführung dieser grundlegenden Definitionen wird jetzt die *Syntax der Metasprache* definiert. Die Semantik der Metasprache, d.h. die Verwendungsweise ihrer Komponenten, wird anschließend durch den Ableitungsbegriff definiert.

Definition 6. Eine *Generalisierte Phrasenstrukturgrammatik* G ist ein geordnetes 13-Tupel $G = (M, W, T, K, X,$ FOOT, HEAD, CONTROL, *Lex, ID, LP, Meta, FCR)*, wobei gilt:

- M und W sind endliche Mengen von Merkmalnamen und Merkmalwerten (wie oben).
- T ist eine endliche Menge von *Terminalsymbolen.*
- K ist die Menge der komplexen Kategorien zu M und W.
- $X \subset K$ ist eine Menge von *Startkategorien.*
- FOOT $\subset M$, HEAD $\subset M$ und CONTROL $\subset M_{cat}$ sind die für die MIPs relevanten Merkmalmengen. Sie können einen nichtleeren Durchschnitt besitzen.
- *Lex* ist eine endliche Menge von geordneten Paaren aus $K \times T$, den *Lexikoneinträgen.* Man schreibt einen Lexikoneintrag als $C_0 \longrightarrow t$.
- ID ist eine endliche Menge von geordneten Paaren aus $\mathbf{ID} = K \times \mathcal{P}_{mul}(K)$. Ein Paar $(C_0, \{C_1, \ldots, C_n\}_{mul})$ (geschrieben $C_0 \longrightarrow C_1, \ldots, C_n$) heißt *unmittelbare Dominanzregel* (immediate dominance rule; kurz: ID-Regel).
- LP ist eine asymmetrische, transitive Relation aus $K \times K$. Ein geordnetes Paar $(C_1, C_2) \in LP$ (geschrieben $C_1 \prec C_2$) heißt *lineare Präzendenz-Aussage* (linear precedence statement; kurz: LP-Aussage).
- *Meta* ist eine endliche Menge von partiellen Funktionen m von ID-Regeln in Mengen von ID-Regeln: $m : \mathbf{ID} \mapsto \mathcal{P}(\mathbf{ID})$.[11] Eine solche Funktion heißt *Metaregel.*
- FCR ist eine endliche Menge von Boole'schen Prädikaten $p : K \mapsto \{\text{true}, \text{false}\}$. FCR ist die Menge der *Feature Cooccurrence Restrictions* (kurz: FCRs). Sei $\mathsf{m} \in M$, $\mathsf{w} \in W \cup K$. FCRs sind wie folgt aufgebaut.[12] p ist von der Form (m) oder $(\mathsf{m} : \mathsf{w})$. Wenn p_1 eine FCR ist, so auch $\neg p_1$. Wenn p_1 und p_2 FCRs sind, so auch $p_1 \wedge p_2, p_1 \vee p_2, p_1 \supset p_2$ und $p_1 \equiv p_2$.

Vor der Definition der Semantik der Metasprache betrachten wir die Anwendung von Metaregeln, die die Existenz von zusätzlichen ID-Regeln sicherstellen.

Die durch die Anwendung der Metaregeln gegebene Menge von ID-Regeln muß endlich sein. Dies wird sichergestellt, indem jede Metaregel höchstens einmal bei der Erzeugung einer ID-Regel angewendet werden darf.

Um dies formal darzustellen, benötigen wir eine Definition der Anwendung einer Metaregel auf eine beliebige Menge von ID-Regeln. Metaregeln sind als *partielle* Funktionen eingeführt. Daher setzen wir sie als totale Funktionen fort mithilfe einer Funktion f.

Definition 7. Sei $f : [\mathbf{ID} \mapsto \mathcal{P}(\mathbf{ID})] \times \mathbf{ID} \mapsto \mathcal{P}(\mathbf{ID})$ eine totale Funktion von Metaregeln und ID-Regeln in die Potenzmenge der ID-Regeln mit

$$f(m, r) = \begin{cases} S & \text{falls } m(r) = S \\ \{r\} & \text{falls } m(r) \text{ undefiniert} \end{cases}$$

[11] Tatsächlich werden Ursprung und Bild durch *ID-Regel-Schemata* dargestellt (vgl. Abschnitt 4.4).

[12] Genau dieselbe Syntax haben *Feature Specification Defaults* in GKPS.

Das Resultat der Anwendung einer Metaregel ist die Eingabe für die Anwendung der nächsten. Die Reihenfolge der Anwendung einer Menge von Metaregeln ist also kritisch. Wir benutzen im folgenden Permutationen der Elemente von beliebigen Teilmengen der Metaregeln, um alle Reihenfolgen zu erfassen.

Definition 8. Sei $\mathcal{M}$ eine Menge von Metaregeln und $m_1, \ldots, m_n \in \mathcal{M}$.

$\Pi_{\mathcal{M}}$ ist die Menge aller Permutationen $\langle m_1, \ldots, m_i \rangle$ der Elemente von jeder Menge in $\mathcal{P}(\mathcal{M})$. Für die leere Menge sei $\langle\rangle \in \Pi_{\mathcal{M}}$.

Sei ferner $R_{\langle\rangle}$ eine Menge von ID-Regeln. $R_{\langle\rangle}$ heißt *Basismenge*.

Für alle $\langle m_1 \ldots m_n \rangle \in \Pi_{\mathcal{M}}$ gilt:

$$R_{\langle m_1 \ldots m_n \rangle} = R_{\langle m_1 \ldots m_{n-1} \rangle} \cup \bigcup_{r \in R_{\langle m_1 \ \ldots \ m_{n-1} \rangle}} f(m_n, r)$$

Der *Abschluß* R einer Basismenge von ID-Regeln unter $\mathcal{M}$ ist

$$R = \bigcup_{\pi \in \Pi_{\mathcal{M}}} R_\pi.$$

Die Verwendungsweise der ID-Regeln, LP-Aussagen und FCRs wird im Rahmen eines Ableitungsbegriffs für Strukturen definiert. Der Ableitungsbegriff umfaßt außerdem Beschränkungen über Kookkurrenzen von Merkmalspezifikationen in direkten Ableitungsschritten, die mithilfe der MIPs kodiert sind. Die MIPs heißen *Foot Feature Principle* (FFP), *Control Agreement Principle* (CAP) und *Head Feature Convention* (HFC). Die jeweils betroffenen Merkmale sind in den Mengen FOOT, CONTROL und HEAD zusammengefaßt.

Die nachstehenden Definitionen der MIPs sind stark vereinfacht. Dies ist an dieser Stelle erforderlich, um die Definitionen überschaubar zu halten. Im Unterschied zu den in nachfolgenden Abschnitten (4.6 bis 4.7) gegebenen Definitionen werden hier für jeden direkten Ableitungsschritt eigene *Teilmengen* von FOOT, CONTROL und HEAD angenommen, für die die MIPs gelten. Es bleibt an dieser Stelle offen, welche Merkmale in diesen Teilmengen jeweils enthalten sind. Somit wird lediglich ausgesagt, daß es Merkmalspezifikationen gibt, die auf eine bestimmte Weise in ihren Werten übereinstimmen. In den genannten Abschnitten werden für die Merkmale in FOOT, CONTROL und HEAD Bedingungen angegeben, unter denen die MIPs eine solche Übereinstimmung fordern.

Definition 9. Sei G eine GPS-Grammatik wie oben definiert, sei ID_{meta} der Abschluß von ID unter *Meta* und seien $P = \gamma C_0' \delta$ und $Q = \gamma \beta \delta$ Wörter über $K \cup T$. Q heißt *direkte Ableitung* von P in G ($P \overset{G}{\Longrightarrow} Q$), gdw. gilt:

- Wenn $\beta \in T$, dann gibt es $C_0 \longrightarrow \beta \in Lex$ mit $C_0 \sqsubseteq C_0'$.[13]
- Wenn $\beta = C_1' \ldots C_n'$, $(C_i' \in K)$, dann gelten die folgenden Beschränkungen $(1 \leq i, j \leq n)$:

[13]Diese Definition beruht auf der Annahme eines Vollformenlexikons. Abschnitt 10 geht auf äquivalente Alternativen ein.

ID-Regeln. Es gibt eine ID-Regel $C_0 \longrightarrow C_1, \ldots, C_n \in ID_{meta}$ mit $C_0 \sqsubseteq C'_0$, und es gibt eine Permutation π mit $\pi(C_1, \ldots, C_n) = \langle C''_1 \ldots C''_n \rangle$, so daß gilt: $C''_i \sqsubseteq C'_i$.

LP-Aussagen. Für alle Kategorien $C'_i, C'_j \in \beta$ mit $i < j$ (d.h. C'_i steht vor C'_j) gilt: Es gibt keine Kategorien $C_i, C_j \in K$ mit

$$C_i \sqsubseteq C'_i \quad \wedge \quad C_j \sqsubseteq C'_j \quad \wedge \quad C_j \prec C_i \in LP$$

FCRs. Alle C'_i $(0 \leq i \leq n)$ sind *legal*; d.h. für alle C'_i und alle $p \in FCR$ gilt:

$$(\mathsf{m})(C'_i) = \text{true} \leftrightarrow \exists \mathsf{x} \in W \cup K : \{(\mathsf{m} : \mathsf{x})\} \sqsubseteq C'_i$$

$$(\mathsf{m} : \mathsf{w})(C'_i) = \text{true} \leftrightarrow \{(\mathsf{m} : \mathsf{w})\} \sqsubseteq C'_i$$

$$\neg p(C'_i) = \text{true} \leftrightarrow p(C'_i) = \text{false}$$

Seien p_1 und p_2 FCRs, und sei „o" eine Operatorvariable mit $\mathsf{o} \in \{\vee, \wedge, \supset, \equiv\}$. Dann gilt:

$$p_1(C'_i) \,\mathsf{o}\, p_2(C'_i) = \text{true} \leftrightarrow (p_1(C'_i) = \text{true}) \,\mathsf{o}\, (p_2(C'_i) = \text{true})$$

FFP. Es gibt eine Menge $K' \subset \{C'_1, \ldots, C'_n\}$ und eine Menge $M' \subset$ FOOT, so daß für alle $\mathsf{m} \in M'$ gilt:

$$\{(\mathsf{m} : C'_0(\mathsf{m}))\} = \sqcup\{\{(\mathsf{m} : C'_i(\mathsf{m}))\} | C'_i \in K'\}$$

CAP. Es gibt i, j $(i \neq j)$ und eine Menge $M' \subset$ CONTROL, so daß für alle $\mathsf{m} \in M'$ gilt:

$$C'_i(\mathsf{m}) \sqsubseteq C'_j$$

HFC. Es gibt ein i und eine Menge $M' \subset$ HEAD, so daß für alle $\mathsf{m} \in M'$ gilt:

$$C'_0(\mathsf{m}) = C'_i(\mathsf{m})$$

Aufgrund der Definition des direkten Ableitungsschritts in einer GPS-Grammatik kann nun der Ableitungsbegriff formuliert werden. Abschließend wird die von einer GPS-Grammatik generierte Sprache definiert.

Definition 10. Sei G eine GPS-Grammatik und seien $\alpha_1, \ldots, \alpha_n$ Wörter über $K \cup T$. Dann heißt α_n *Ableitung* von α_1 in G $(\alpha_1 \overset{*G}{\Longrightarrow} \alpha_n)$ gdw. gilt:

$$\alpha_1 \overset{G}{\Longrightarrow} \alpha_2,\ \alpha_2 \overset{G}{\Longrightarrow} \alpha_3,\ \ldots,\ \alpha_{n-1} \overset{G}{\Longrightarrow} \alpha_n.$$

Definition 11. Sei G eine GPS-Grammatik. Sei T^+ die Menge der Wörter über T. Die von G erzeugte Sprache $L(G)$ ist

$$L(G) = \{\alpha \mid \alpha \in T^+,\ C_0 \overset{*G}{\Longrightarrow} \alpha,\ C_0 \in X\}.$$

4 Die axiomatische Version von GPSG

In diesem Abschnitt werden die Komponenten der Metasprache von GKPS beschrieben, die im Rahmen des Berliner GPSG-Formalismus rekonstruiert werden.

Abschnitt 4.1 führt Merkmale und Kategorien ein sowie die Begriffe der Extension und der Unifikation. In Abschnitt 4.2 werden Aussagen über Kategorien in Form von FCRs und FSDs definiert. Anschließend folgt in Abschnitt 4.3 die Definition des ID/LP-Formats und eine Diskussion der ECPO-Eigenschaft, die Zusammenhänge zwischen ID-Regeln und LP-Aussagen aufdeckt. Sie bilden den Anstoß für ein erweitertes LP-Konzept, das [Uszkoreit 1986a] und [Hauenschild 1988] beschreiben. Danach definiert Abschnitt 4.4 Metaregeln in ihrer eingeschränkten Rolle und kritisiert das Metaregelkonzept. Alternativen werden anhand von Vorschlägen von Kilbury aufgezeigt.

Den Zusammenhang zwischen ID-Regeln und lokalen Bäumen präzisiert Abschnitt 4.5, in dem der zentrale Begriff des *zulässigen Baumes* definiert wird. Dieser Begriff ist zu dem des direkten Ableitungsschritts in Abschnitt 3 formal gleichwertig.

In den Abschnitten 4.6-4.8 schließlich werden die MIPs zwar informell, aber vollständig definiert.

4.1 Merkmalspezifikationen und Kategorien

Komplexe Kategorien sind Mengen von Merkmalspezifikationen, die einigen zusätzlichen Bedingungen genügen. Merkmalspezifikationen sind geordnete Paare (m : w) bestehend aus einem Merkmalnamen m und einem Merkmalwert w. Der Merkmalname ist ein atomares Symbol; der Merkmalwert ist entweder ein atomares Symbol oder eine Kategorie. Das Merkmal Numerus z.B. hat den Namen plu und die Werte + und -.[14] Kategorien enthalten zu jedem Merkmal höchstens eine Spezifikation. Sie können daher als (partielle) Funktionen von Merkmalnamen in Merkmalwerte angesehen werden. Z.B. gilt für Pluralkategorien *C :* C'(plu) = +.

Die Endlichkeit der Menge der möglichen Kategorien wird wie in der mittleren GPSG sichergestellt (vgl. S. 10).

Es gibt verschiedene Vorschläge zur Definition komplexer Kategorien. GKPS geben eine komplexe, induktive Definition über die Elemente von M_{cat}- Die in Definition 3 auf S. 23 gegebene Version ist wesentlich einfacher; sie benutzt ein Prädikat r, das nach wiederholten Vorkommen eines Merkmalnamens auf den unterschiedlichen Einbettungstiefen „sucht". *T* operiert über mengenartigen Merkmalwerten, die als Hilfskonstrukt ausschließlich für die Definition komplexer Kategorien eingeführt wurden. In komplexen Kategorien ist *r* nicht erfüllt. Die Grundidee wird auch von anderen Autoren benutzt (vgl. [Gazdar *et al.* 1986] und [Hukari/Levine 1986, S. 216-221]).

In [Naumann 1988, S. 11] wird ebenfalls versucht, ohne Induktion auszukommen, doch die dort verwendete transitive Hülle der Element-Relation ($\in^+$) leistet nicht das Geforderte, nämlich abwechselnd mengenwertige Merkmalspezifikationen

[14][Naumann 1988, S. 9] weist zu Recht darauf hin, daß oft (so auch in GKPS) die Begriffe Merkmal und Merkmalname nicht klar unterschieden werden.

und mengenartige Merkmalwerte zu betrachten. Die Verwendung von $\in^{+}$ in GKPS (S. 36ff) wird in [Hukari/Levine 1986] aus demselben Grunde kritisiert.

Es folgen einige Beispiele für Kategorien. Die Merkmalspezifikationen orientieren sich an der Grammatik für Englisch in GKPS.

(4) {(n:+), (v:-), (bar:2)}

(5) {(n:-), (v:+), (bar:2), (subj:+)}

(6) {(n:-), (v:+), (bar:2), (subj:+),
(slash:{(n:+), (v:-), (bar:2)})}

(4) stellt eine NP dar, (5) ein S und (6) ein S, dem eine NP „fehlt".

An dieser Stelle sind einige Notationsvereinbarungen zu treffen, die später noch ergänzt werden. Ich halte mich an die in GPSG üblichen Konventionen, verwende aber nicht alle von ihnen. Eine Merkmalspezifikation (m : w) wird auch als [m:w] geschrieben; bei w € {+, -} verwende ich [wm]. Bestimmte Kategorien werden in Kurzform notiert (vgl. die obigen Beispiele). Beide Schreibweisen werden kombiniert, so daß eine Nominativ-NP durch NP[cas:nom] dargestellt wird. Wo der Merkmalwert eindeutig auf den Merkmalnamen schließen läßt, kann er ausgelassen werden; z.B. NP(nom). Statt C[slash:C'] verwende ich die allgemein übliche Notation C / C'.

Im Unterschied zu GKPS verwende ich *nicht* die Notation [m], um auszudrücken, daß das Merkmal m eine wie auch immer geartete Spezifikation besitzt. Diese Notation gibt Anlaß zu Verwechslungen: Bedeutet [plu] die Spezifikation eines Merkmals Numerus mit Plural oder die Aussage, daß Numerus mit dem Merkmalnamen plu kodiert wird und spezifiziert ist?

Die Kategorien in den obigen Beispielen sind *unterspezifiziert*, denn *M* enthält mehr Merkmalnamen als n, v, bar, subj und slash. Die systematische Beziehung zu stärker spezifizierten Kategorien wurde unter dem Begriff der Extension eingeführt (Definition 4 auf S. 23). Sie stellt eine partielle Ordnung über der Menge der Kategorien dar. Die maximal unterspezifizierte Kategorie ist die leere Menge. Die Kategorien in (4)-(6) sind allesamt Extensionen von [bar:2]. In GPSG wird ausgiebig von der Möglichkeit Gebrauch gemacht, mit unterspezifizierten Kategorien über deren Extensionen zu generalisieren. Mit [bar:2] beispielsweise kann man sich auf phrasale Kategorien beziehen.

Die Unifikation spielt in GKPS eine nebengeordnete Rolle (sie wird lediglich für die Definition des FFP benötigt). Dies mag überraschen, da die Theorie der komplexen Kategorien offenbar besonders gute Voraussetzung für die Verwendung der Unifikation verspricht. Der Grund liegt in der strikt axiomatischen Definition durch GKPS, die die prozeduralen Effekte der Unifikationsoperation einfach nicht braucht. Ich komme noch ausführlich auf diesen Punkt in Teil II zu sprechen.

Da zu einer beliebigen Teilmenge der Kategorien nicht immer eine kleinste obere Schranke existiert, ist die Unifikation von Kategorien nur partiell definiert (vgl. Definition 5 auf S. 23).

4.2 FCRs und FSDs

Die Kategoriendefinition erlaubt linguistisch unerwünschte Kategorien wie z.B. [vform:fin, nom][15], die mithilfe von FCRs ausgeschlossen werden können. FCRs beschränken die Menge der möglichen Kategorien auf die *legalen,* d.h. diejenigen, die in syntaktischen Strukturen vorkommen dürfen. FCRs sind Prädikate über Kategorien, die gewöhnlich als Implikationen oder Äquivalenzen kodiert werden. Es gibt FCRs, die als universelle Prädikate gelten, wie etwa (7), worin ausgedrückt wird, daß legale Kategorien, die für subcat spezifiziert sind, keine Spezifikation für slash enthalten dürfen. Außerdem gibt es FCRs, die einzelsprachliche Verhältnisse ausdrücken, wie etwa (8) für das Englische. Darin wird festgestellt, daß ein Verb am Satzanfang (d.h. in invertierten Sätzen) ebenfalls als Hilfsverb und als finites Verb ausgewiesen sein muß.

(7) {subcat) *D* -{slash)

(8) {inf: +) D ({aux : +) A {vform : fin))

Die Notation der FCRs und ihre Interpretation ergibt sich aus den Definitionen 6 (S. 23) und 9 (S. 25).[16]

(9) ({n : +) A {v : -) A {bar: 2)) = {cas : acc)

Die FSDs verlangen das Auftreten bestimmter Merkmal-Wert-Paare, wenn keine anderen Komponenten der Metasprache dem entgegenstehen. Damit kann beispielsweise ausgedrückt werden, daß NPs im Englischen gewöhnlich im Akkusativ stehen (vgl. (9)). Im Falle von Subjekts-NPs würde eine Kasusspezifikation aus einer ID-Regel verhindern, daß ein FSD {cas : acc) spezifizieren könnte.

Die Syntax von FSDs ist zwar identisch mit der der FCRs, ihre Semantik ist jedoch sehr komplex, wenn sie in der Form von Constraints dargestellt werden soll. Da im Berliner GPSG-System keine FSDs verwendet werden, wird auf eine Diskussion verzichtet.

[15]Das Merkmal Verbform (denotiert durch vform) kann durch *fin, inf, prp* und *pas* spezifiziert werden. Sie stehen für finite und infinite Form, Perfekt-Partizip und Passiv-Partizip.

[16]GKPS schreiben FCRs irreführenderweise als Boole'sche Ausdrücke über Kategorien; z.B. (statt (8)) „[+inv] D [+aux, fin]". Die Interpretation, die GKPS angeben, ist als Folge der irreführenden Notation fehlerhaft, wie [Hendriks 1986] zeigt. Für (8) übersetzen GKPS:

VC : [C(inv) = +] D [(C(aux) = +) A (C(vform) = fin)]

Die Verwendung der Gleichheitsrelation ergibt Probleme bei kategorienwertigen Merkmalen. Hendriks führt die FCR

[fin, agr : NP] D [agr : NP[nom]]

und ihre Interpretation nach GKPS an (zu dem kategorienwertigen Merkmal agr siehe Abschnitt 4.7):

V<7 : [C(vform) = fin A C(agr) = NP] D C(agr) = NP[nom]

Dies ist äquivalent zu der (nicht intendierten) Formel

VC : *[C(*vform) ^ fin] V [C(agr) / NP].

Anstelle der Gleichheitsrelation muß die Extensionsbeziehung verwendet werden, wie in den hier gegebenen Interpretationsvorschriften.

4.3 Das ID/LP-Format

Eine ID-Regel der Form Co —> C_1, C_2, C_3 läßt sechs lokale Bäume zu, die sich durch die Reihenfolge ihrer Töchter unterscheiden. ID-Regeln sind insofern *permissiv.* LP-Aussagen beschränken die Menge der lokalen Bäume, die von ID-Regeln zugelassen werden, durch partielle Aussagen über die Reihenfolge der Töchter. Sie sind insofern *restriktiv.* Eine LP-Aussage der Form *C\ -< €3* läßt von den sechs lokalen Bäumen nur drei zu.

Man beachte, daß bei der Schreibweise von ID-Regeln—im Unterschied zu der von PS-Regeln—die Töchter durch Kommata getrennt werden (vgl. Definition 6 auf S.23).

Die Definition der LP-Aussage *(C -< C')* besagt, daß in zulässigen lokalen Bäumen Extensionen von *C* vor (d.h. links von) Extensionen von *C'* auftreten müssen, wenn sie Schwestern sind.

Die LP-Aussage (12) regelt somit die Abfolge von NP und VP[inf] in der ID-Regel (10) sowie von A und VP[inf] in der ID-Regel (11).

(10) VP—> V,NP,VP[inf]

(11) AP^A,VP[inf]

(12) [+n] X VP

Wie in Abschnitt 2.2 angedeutet, macht die Verwendung des ID/LP-Formats eine Annahme über Grammatiken natürlicher Sprachen, die nicht für alle kontextfreien Grammatiken zutrifft. GKPS behaupten nämlich, daß nicht jede kontextfreie Grammatik in das ID/LP-Format überführbar sei:

> There are some PSGs that are not expressible in ID/LP format. [GKPS, S. 48]

Die notwendige und hinreichende Bedingung für die Überführbarkeit bestehe in der Eigenschaft der Töchter in einer PS-Regel, eine partielle Ordnung einzuhalten, die von den Töchtern in allen anderen PS-Regeln eingehalten wird. In der Menge von PS-Regeln (13) ist diese Bedingung verletzt.

(13) {A—>BC, D—>ECB}

Da LP-Aussagen Ordnungsrelationen über Schwesterknoten beliebiger lokaler Bäume etablieren, können sie nicht ausdrücken, daß z.B. B auf C folgen muß, wenn ihnen E vorangeht; daß sonst jedoch C auf B folgen muß.

Wenn eine kontextfreie Grammatik der genannten Bedingung genügt, dann hat sie die ECPO-Eigenschaft (ECPO steht für Exhaustive Constant Partial Ordering). GKPS behaupten, daß eine kontextfreie Grammatik genau dann in ID/LP-Format überführt werden kann, wenn sie die ECPO-Eigenschaft besitzt [GKPS, S. 49].

Mit der Verwendung des ID/LP-Formats in GPSG geht somit die Behauptung einher, daß ECPO eine universelle Eigenschaft von Grammatiken für natürliche Sprachen ist. Diese Behauptung ist statistisch unerwartet und daher sehr restriktiv. Sie könnte sich als interessante (im Sinne von GKPS) linguistische Universalie erweisen.

4.3.1 Zur Relevanz der ECPO-Eigenschaft

Die Relevanz der ECPO-Eigenschaft muß genauer untersucht werden. Da wohl selten jemand in der Praxis vorhat, eine gegebene kontextfreie Grammatik in eine ID/LP-Grammatik zu überführen, scheint die ECPO-Eigenschaft lediglich theoretische Relevanz im Hinblick auf die generative Kapazität der GPSG-Theorie zu haben. Dies ist nur auf den ersten Blick richtig. Sie hat durchaus Konsequenzen für die Beziehung zwischen Formalismus und Grammatik.

Wie [Shieber 1984, S. 145] zeigt, kann jede kontextfreie Sprache mit einer ID/LP-Grammatik beschrieben werden. Damit ist die zitierte Behauptung von GKPS, es gebe kontextfreie Grammatiken, die nicht im ID/LP-Format ausgedrückt werden könnten, formal widerlegt. Shieber geht noch einen Schritt weiter und behauptet:

> In fact, a stronger theorem can be proved: every context-free grammar ist *structurally equivalent* to an ID/LP grammar, that is, it is strongly equivalent up to a renaming of nonterminal labels [...] [Shieber 1984, S. 153, Fn. 7]

Naumann gibt ein Verfahren auf der Basis von Shiebers Beweis dafür an [Naumann 1988, S. 30fj. Dabei werden alle Vorkommen gleicher Kategorien auf den rechten Seiten der PS-Regeln umbenannt[17]; z.B. würden B und C in der ersten Regel von (13) in B! bzw. G! und in der zweiten Regel in B_2 bzw. C_2 umbenannt werden. Danach lassen sich LP-Aussagen über die Abfolge aller Kategorien formulieren. Naumann behauptet jedoch, anders als Shieber, daß mit seinem Verfahren aus einer kontextfreien Grammatik *G* eine zu *G stark äquivalente* ID/LP-Grammatik *G'* erzeugt werde. Dieser Behauptung liegt folgende Definition zugrunde.

> Zwei Syntaxen G_1 und G_2 sind stark äquivalent gdw.
>
> (i). $L(G_1) = L(G_2)$ und
> (ii). G_1 und G_2 allen Sätzen isomorphe Strukturen zuordnen.
>
> [Naumann 1988, S. 30, Fn. 22]

Diese Definition weicht ab von der allgemein üblichen, in der nicht nur Strukturäquivalenz sondern auch Isomorphie der Knotenbezeichner gefordert wird. Letzteres erfüllen Grammatiken ohne ECPO-Eigenschaft nicht. Die Behauptung von GKPS stimmt also nur, wenn man unter „Überführbarkeit in das ID/LP-Format" die Isomorphie der Knotenbezeichner mitversteht.

Die Isomorphie der Knotenbezeichner ist aus linguistischen Gründen wünschenswert, denn das ID/LP-Format sieht schließlich gerade die *Trennung* der unmittelbaren Dominanzbeziehungen von den linearen Präzedenzbeziehungen vor. Bei einer Grammatik, die nicht ECPO ist, werden, wie das obige Beispiel zeigt, zusätzliche nonterminale Kategorien benötigt, mithin auch zusätzliche Dominanzbeziehungen und—infolgedessen—auch weitere Präzedenzbeziehungen. Die Existenz dieser zusätzlichen Objekte ist nicht auf linguistische, sondern ausschließlich auf formale Gründe zurückzuführen. Eine solche Vermischung konterkariert die Leitidee

[17] In der Praxis heißt dies im Falle komplexer Kategorien, weitere Merkmalspezifikationen einzuführen.

von GPSG, in der Metasprache genau die linguistisch motivierten Regelhaftigkeiten explizit zu machen.

So gesehen, stellt die ECPO-Eigenschaft nur sicher, daß eine unabhängig motivierte Verwendung von Kategorien (im Falle komplexer Kategorien: von Merkmal-Spezifikationen) direkt in das ID/LP-Format übertragen werden kann.

4.3.2 Zum Zusammenhang zwischen ID und LP

Andererseits legt das ID/LP-Format eine Beschreibung sprachlicher Fakten nahe, die die ökonomischen Darstellungsmöglichkeiten ausnutzt: Bei total freier Wortstellung von *n* Konstituenten liegt es z.B. nahe, eine einzige ID-Regel anzusetzen, die die n Kategorien als Töchter hat (für *n* = 3 etwa S —> A, B, C). Es gibt auch Alternativen, etwa eine binär verzweigende Konstituentenstruktur, doch ist die zugrundeliegende ID/LP-Syntax aufwendig und schließt zusätzliche nonterminale Symbole und LP-Aussagen ein, um Mehrfachanalysen zu verhindern. Für drei Konstituenten A, B und C leistet u.a. folgende Syntax das Gewünschte:

(14)
$$\begin{array}{ll} S \longrightarrow A, T_1 & T_1 \longrightarrow B, C \\ S \longrightarrow B, T_2 & T_2 \longrightarrow A, C \\ S \longrightarrow C, T_3 & T_3 \longrightarrow A, B \\ A \prec T_1,\ B \prec T_2,\ C \prec T_3 & \end{array}$$

Angesichts dessen wundert es nicht, daß die meisten GPSG-Analysen des deutschen Mittelfelds flache Strukturen ansetzen (vgl. [Nerbonne 1982, Uszkoreit 1984, Russell 1985, Preuß 1987]), obwohl die linguistische Diskussion über die Vorzüge flacher oder binär verzweigender Strukturen keineswegs zu eindeutigen Resultaten geführt hat.

4.3.3 ECPO und partielle Freiheit der Wortstellung

Eine wichtige Konsequenz der ECPO-Eigenschaft für die Formulierung von ID/LP-Syntaxen liegt darin, daß bestimmte partielle Ordnungen nur unter Einführung von zusätzlichen nonterminalen Kategorien beschreibbar sind. Die Behauptung, daß kontextfreie Grammatiken für natürliche Sprachen die ECPO-Eigenschaft besitzen, sagt voraus, daß diese Kategorien unabhängig motiviert sind.

Geht man von einer Menge von ID-Regeln aus, die die Dominanzbeziehungen für ein Fragment einer natürlichen Sprache beschreiben, so reflektieren die LP-Aussagen für dieses Fragment direkt die ECPO-Eigenschaft. In [Uszkoreit 1986a] werden die im ID/LP-Format ausdrückbaren Beziehungen unterteilt in total freie Wortstellung (Abwesenheit von LP-Aussagen) und vollkommen festgelegte Wortstellung (Anwesenheit von LP-Aussagen). In den meisten Sprachen ist die Wortstellung *teilweise frei* insofern, als eine Interaktion verschiedener Ordnungsprinzipien vorliegt, die für ein Paar von Kategorien unterschiedliche Reihenfolgen verlangen können. Uszkoreit zeigt am Beispiel der Abfolge von Satzgliedern im Mittelfeld deutscher Sätze, für das er eine flache Struktur zugrunde legt (vgl. [Uszkoreit 1984]), daß die teilweise freie Wortstellung nicht direkt im ID/LP-Format kodierbar ist. Er schlägt daher eine Erweiterung der Definition von LP-Aussagen vor, die der Interaktion gerecht wird. Folgende Prinzipien werden identifiziert:

(15) (i) [agent] steht tendenziell vor [theme]
(ii) [agent] steht tendenziell vor [goal]
(iii) [goal] steht tendenziell vor [theme]
(iv) [-focus] steht tendenziell vor [+focus]
(v) [+prn] steht tendenziell vor [-prn]

Wie [Lenerz 1977] feststellt, sind Sätze grammatisch, wenn mindestens eines der Prinzipien erfüllt ist. Uszkoreit schlägt *komplexe LP-Aussagen* vor, die aus mehreren Klausen im Format gewöhnlicher LP-Aussagen bestehen. Ein lokaler Baum genügt einer komplexen LP-Aussage, wenn jedes Paar von Schwestern mindestens einer anwendbaren Klause genügt. Auf diese Weise gelingt es, die gefundenen sprachlichen Fakten *direkt* in der Metasprache zu kodieren.

Wie [Hauenschild 1988] nachweist, sind die Merkmale wiederum voneinander abhängig. Durch die komplexe LP-Aussage auf der Basis von (15) werden der Satz (16) nicht zugelassen, wenn das indirekte Objekt nicht eine Extension von [+focus] ist (obwohl er auch dann grammatisch sein kann) und die Sätze (17)-(18) nicht zurückgewiesen, obwohl sie (von vielen deutschen Muttersprachlerinnen) zumindest als schlecht eingestuft werden. Offensichtlich ist die Wortstellung zweier pronominaler Konstituenten strikter geregelt, während sie bei Beteiligung nichtpronominaler NPs teilweise frei ist.

(16) Sie gab es [theme] ihm [goal].

(17) ? Sie gab ihm [goal, +prn] es [theme, +prn].

(18) ? Sie gab dem Mann [goal, -prn] es [theme, +prn].

Hauenschilds Argumentation läuft darauf hinaus, daß in einem rein syntaktischen Formalismus die Bedeutung der Merkmale für thematische Rolle und Fokus unzureichend erklärt werden können. Sie schlägt anhand eines Mehrebenen-Modells für MÜ vor, die nicht strikten, semantisch-pragmatischen Aspekte der Wortstellung auf einer anderen Repräsentationsebene zu kodieren als die strikten, syntaktischen Aspekte. Wortstellung erhält damit eine wohlfundierte Dimension, die quer durch die traditionellen Ebenen der Syntax, Semantik und Pragmatik verläuft und wohl eine notwendige Voraussetzung für eine adäquate Behandlung dieses Phänomenkomplexes darstellt. Eine detaillierte Ausarbeitung eines solchen Ansatzes steht aber noch aus.

4.3.4 Verletzung **der ECPO-Eigenschaft**

Pollard gibt Beispiele an, die die Annahme, eine Grammatik für Englisch habe die ECPO-Eigenschaft, infrage stellen [Pollard 1984, S. 60, 66]. Die Verben *appear* und *appeal* können in Sätzen zusammen mit den gleichen Konstituenten auftreten (19)-(22). Nach GKPS werden sie dann durch dieselbe ID-Regel (z.B. (23)) eingeführt.

(19) Kim appeared to Sandy to be optimistic.

(20) Kim appeared to be optimistic to Sandy.

(21) Kim appealed to Sandy to be optimistic.

(22) * Kim appealed to be optimistic to Sandy.

(23) VP—>V,PP[to],VP

Es gibt keine Menge von LP-Aussagen, die die Grammatikalität von (19)-(21) und zugleich die Ungrammatikalität von (22) beschreibt. Pollard behauptet, daß der entscheidende Punkt die *grammatische Funktion* der PP *to Sandy* sei (obliques Objekt bzw. indirektes Objekt). Auf die grammatische Funktion können LP-Aussagen jedoch keinen Bezug nehmen, wenn sie nicht in den Kategorien kodiert ist.[18]

[Sag 1986] schlägt zur Behebung des Problems im Rahmen von HPSG vor, neben den gewöhnlichen LP-Aussagen solche zu haben, die die Position einer Konstituente lediglich hinsichtlich den Schwestern beschränkt, die höher in der Obliqueness-Hierarchie stehen. Auf diese Weise kommt ein Begriff der grammatischen Funktion ins Spiel, der das von Pollard benannte Problem löst.

Andererseits ist die Verletzung der ECPO-Eigenschaft nur dann ein ernstes Problem, wenn man tatsächlich beide Verben durch dieselbe ID-Regel einführt. Dazu ist man, technisch gesehen, nicht gezwungen. Die syntaktische Funktion von Konstituenten läßt sich ohne weiteres durch Merkmalspezifikationen unterscheiden.

Wir stehen mithin vor der Frage, welchen Stellenwert Merkmale in GPSG haben. Offenbar legt die Metasprache die Menge der Merkmale nicht fest. Einige Merkmale kommen sinnvollerweise in jeder GPS-Grammmatik vor (z.B. bar), aber nirgends wird die Menge der Merkmale beschränkt. Der inflationäre Gebrauch von Merkmalen kann also nur durch Argumente beschränkt werden, die außerhalb der Metasprache von GKPS verankert sind. Hinsichtlich der Kodierung der grammatischen Funktion als Merkmal mag man etwa einwenden, daß dies theoretisch nicht begründet sei und nur dazu diene, Beispiele wie (22) auszuschließen.

Ich kann diese Frage nicht grundsätzlich beantworten; es soll jedoch betont werden, daß der Formalismus viele technische Möglichkeiten bietet, sprachliche Fakten zu beschreiben und daß nicht unbedingt die Anzahl der beschriebenen Sätze ein Maß für die linguistische Adäquatheit einer Grammatik darstellt.

4.4 Metaregeln

Metaregeln sind in den verschiedenen Versionen von GPSG starken formalen Änderungen unterworfen gewesen. In diesem Abschnitt wird die bisher am stärksten eingeschränkte Fassung vorgestellt. Da Metaregeln im Berliner GPSG-System nur eine marginale Rolle spielen, gibt der vorliegende Abschnitt lediglich einen Überblick und verweist auf die einschlägige Literatur.

Der Effekt von Metaregeln ist die Explizierung systematischer Beziehungen zwischen ID-Regeln:

> [...] they amount to nothing more than a novel type of rule collapsing convention for rules [...] [GKPS, S. 66]

Formal lassen sich Metaregeln als Funktionen von ID-Regeln in (Mengen von) ID-Regeln auffassen. Eine Metaregel besteht aus zwei ID-Regel-Schemata, die Mengen

[18]Die Notwendigkeit, die Funktion von Konstituenten einzubeziehen, wird auch bei der Zuweisung semantischer Typen zu Kategorien offensichtlich. GKPS geben keine befriedigende Auskunft, wie das direkte und das indirekte Objekts in der Regel S —«• V, NP.NP zu unterscheiden sind.

von ID-Regeln denotieren. Ein ID-Regel-Schema ist ein Ausdruck der Form a_0 —> $a_1,\ldots,\ a_n,$ wobei die a, bis auf eines Kategorien sind; ein *a,* ist eine Multimenge W von Kategorien. Das Format für Metaregeln sieht folgendermaßen aus:[19]

(24) $$\begin{array}{l} a_0 \longrightarrow \alpha_1, \ldots, \alpha_n \\ \qquad\quad \Downarrow \\ ßo \longrightarrow \beta_1, \ldots, \beta_k \end{array}$$

Der Pfeil bedeutet, daß Eigenschaften einer ID-Regel, die das obere Schema matcht, die Existenz einer ID-Regel bewirken, die das untere Schema matcht. Am Beispiel der Passiv-Metaregel aus GKPS wird dies verdeutlicht:

(25) $$\begin{array}{l} \mathrm{VP} \longrightarrow \quad \mathrm{W,\ NP} \\ \qquad\quad \Downarrow \\ \mathrm{VP[pas]} \longrightarrow \mathrm{W,\ (PP[pform{:}by])} \end{array}$$

Sie besagt, daß zu jeder ID-Regel, die eine VP in eine NP und weitere Konstituenten expandiert, eine andere ID-Regel existiert, in der eine VP[pas] genau die weiteren Konstituenten sowie, optional, eine PP dominiert, die durch die Präposition *by* eingeleitet wird. Dabei bleibt die Subkategorisierung unverändert. Z.B. existiert aufgrund der ID-Regel (26) auch die ID-Regel (27).

(26) $\mathrm{VP} \longrightarrow \mathrm{V[6],\ NP}$

(27) $\mathrm{VP[pas]} \longrightarrow \mathrm{V[6],\ (PP[pform\ :\ by])}$

Jacobson zeigt, daß in GKPS eine Metaregel für Passiv nicht nur aus metasprachlichen Erwägungen existiert, sondern daß sie unumgänglich notwendig ist [Jacobson 1987]. Die Aufgabe, die von Metaregeln wahrgenommen wird, erledigen in anderen Theorien (wie LFG, HPSG) lexikalische Regeln, indem sie angeben, wie die Subkategorisierungseigenschaften sich unterscheiden. Auf der Basis des Konzepts von Subkategorisierung in GKPS ist dies nicht möglich. Allein aufgrund der Spezifikationen für subcat im Lexikon besteht keine Möglichkeit, genau diejenigen Verben zu identifizieren, die ein Objekt subkategorisieren und infolgedessen ggf. Passivformen bilden. Darüber hinaus ist es so nicht möglich, Passivformen regelhaft eine andere Subkategorisierung zuzuweisen. Die erforderliche Information ist nur in den ID-Regeln enthalten. Insofern würden Einwände gegen das Konzept der Metaregel eine grundlegende Änderung des Konzepts der Subkategorisierung erfordern.[20]

Im folgenden nenne ich einige Einwände. Metaregeln werden von GKPS nur eingesetzt, um ID-Regeln mit lexikalischem Head in ID-Regeln mit lexikalischem Head abzubilden. [Jacobson 1987] kritisiert, daß die Beschränkung nicht aus der Natur der Metaregeln hervorgeht, sondern „von außen" motiviert wird. Im Falle der Passiv-Metaregel ist sie durch die Eigenschaft der Grammatik in GKPS gegeben, daß alle

[19]GKPS machen hierzu noch weitere Einschränkungen, die hier aber nicht von Belang sind; vgl. hierzu [GKPS, S. 67-72]

[20]Auf einen Vorschlag von Kilbury, der das Subkategorisierungskonzept von GKPS übernimmt und die Komplemente eines Verbs mithilfe semantischer Typen rekonstruiert, gehe ich weiter unten ein.

infrage kommenden ID-Regeln ein für subcat spezifiziertes (und damit lexikalisches) Head besitzen.[21]

Eine weitgehend ungeklärte Frage ist nach Jacobson die generative Kapazität von Formalismen mit Metaregeln. Es ist vielfach betont worden, daß Metaregeln sich grundlegend von Transformationen unterscheiden. Transformationen operieren auf Strukturen, Metaregeln auf ID-Regeln; und eine endliche Menge von ID-Regeln kann nur kontextfreie Sprachen beschreiben. GKPS erwähnen eine Vermutung von Joshi, derzufolge die Anwendung einer Metaregel mit höchstens einer Variable auf PS-Regeln nur zu kontextfreien Sprachen führen kann. Uszkoreit zeigte jedoch, daß unter diesen Bedingungen eine unendliche Menge von PS-Regeln erzeugbar ist, falls nämlich eine Metaregel auf das Resultat ihrer Anwendung anwendbar ist. Es gibt kontextfreie Grammatiken mit einer unendlichen Regelmenge, die eine nicht kontextfreie Sprache beschreiben ([Gazdar/Pullum 1982] und [Naumann 1988] geben ein elegant formuliertes Beispiel von Culy wider). Nach [Peters/Uszkoreit 1982] läßt sich durch unendliche Mengen von PS-Regeln jede rekursiv aufzählbare Menge beschreiben. Welche Klasse von Sprachen bei Metaregel-Anwendung auf ID-Regeln, in denen ja keine Präzedenzrelationen kodiert sind, generiert wird, ist laut [Jacobson 1987] nicht bekannt.

Um den Formalismus mit Sicherheit kontextfrei zu halten, fordern GKPS, daß der Abschluß der ID-Regeln unter Metaregel-Anwendung endlich bleibt. Dies wird erreicht durch die Beschränkung, daß im Verlauf der Ableitung einer ID-Regel jede Metaregel höchstens einmal angewendet werden darf [Thompson 1981]. Dies wird durch die Definitionen 7 (S. 24) und 8 (S. 25) sichergestellt.

Diese Beschränkung ist nach Jacobson ebenfalls unnatürlich, denn sie geht nicht aus den formalen Eigenschaften der Metaregeln selbst hervor. Jacobson plädiert dafür, das Metaregelkonzept ganz aufzugeben zugunsten einer lexikalischen Behandlung der Subkategorisierungsmöglichkeiten.

4.4.1 Category-Cooccurrence Restrictions

Einen anderen Ansatz, bei dem Metaregeln durch eine Rekonstruktion der ID-Regeln als Beschränkungen über das gemeinsame Vorkommen von Kategorien obsolet werden, schlägt Kilbury im Rahmen einer neuen, metasprachlichen Komponente vor [Kilbury 1986, Kilbury 1988], die er in Analogie zu existierenden Komponenten (FCRs) einführt. Wie Kilbury zeigt, gelangt man damit zu einer viel einheitlicheren Formulierung der Metasprache.

Ich werde diesen Ansatz im folgenden darstellen, um einerseits zu zeigen, wie Metaregeln umgangen werden können und andererseits ein Beispiel dafür zu geben,

[21] Hinsichtlich der Korrektheit der Passiv-Metaregel sind einige Anmerkungen notwendig. Offensichtlich werden auch Strukturen mit nicht passivierbaren (transitiven) Verben zugelassen. Kilbury betont in [Kilbury 1986], daß die Hauptaufgabe der Passivbildung einer Regel im Lexikon überlassen bleibt [GKPS, S. 219], die einem Verb eine Passivform zuordnet. Wenn im Lexikoneintrag keine Passivform zu einem Verb vorgesehen ist, kann die durch die Passiv-Metaregel geforderte ID-Regel diese Form nicht einführen. Soweit ich verstehe, erfaßt die Regel im Lexikon jedoch keine idiosynkratischen Verbeigenschaften—sie regelt vielmehr die unterschiedliche Semantik der Formen—und ist daher nicht in der Lage, Sätze wie *Ein Buch wird bekommen* auszuschließen.- [Kilbury 1986] und [Preuß 1989] nennen einige weitere Probleme.

wie die metasprachlichen Komponenten von GPSG aus *linguistischen* Überlegungen heraus modifiziert werden (Modifikationen aufgrund informatischer Erwägungen werden in Kapitel II beschrieben).

Kilbury weist darauf hin, daß in einer ID-Regel zwei verschiedene Arten von Information kombiniert sind:

- die Mutter *M* dominiert unmittelbar eine Tochter *D*
- eine Tochter *DI* kommt als Schwester einer Tochter D_2 im lokalen Baum vor

Indem diese Information separat repräsentiert werden, lassen sich Generalisierungen ausdrücken, die im ID/LP-Format nicht formuliert werden können; z.B.:

- Jeder lokale Baum mit Mutter *M* hat *D* als Tochter.
- Kein lokaler Baum mit D_1 als Tochter hat ebenfalls D_2 als Tochter.

Ausdrücke dieser Art nennt Kilbury *Category Cooccurrence Restrictions* (CCRs). Die in einer Menge von ID-Regeln enthaltene Information kann durch die Menge der Äste verbunden mit geeigneten CCRs dargestellt werden.

(28)
$S \longrightarrow NP, VP$ $\qquad$ $VP \longrightarrow V, VP$
$S \longrightarrow AUX, NP, VP$ $\qquad$ $VP \longrightarrow V, NP$
$VP \longrightarrow AUX, VP$ $\qquad$ $VP \longrightarrow V, NP, VP$

(29) (S, NP), (S, VP), (S, AUX), (VP, V), (VP, VP), (VP, AUX), (VP, NP)

Für (28) (vgl. [GKPS, S. 48 (9i)]) sind die Äste in (29) dargestellt. Kilbury gibt die folgenden CCRs an.[22]

- Jeder lokale Baum mit S als Mutter hat sowohl NP als auch VP als Tochter.
- Jeder lokale Baum mit AUX als Tochter hat ebenfalls VP als Tochter.
- In jedem lokalen Baum mit VP als Mutter kommen entweder AUX oder V als Tochter vor.
- In jedem lokalen Baum mit VP als Mutter, in dem AUX als Tochter vorkommt, kommt NP nicht als Tochter vor.
- Jeder lokale Baum mit V als Tochter hat VP oder NP (oder beide) als Tochter.

Die Rekonstruktion einer umfangreichen Menge von ID-Regeln führt zu einer sehr unübersichtlichen Situation. Die Menge der lexikalischen ID-Regeln wird jedoch auf eine generelle Weise erfaßt, indem man an ihren Heads ein syntaktisches Merkmal typ spezifiziert, dessen Wert eine Folge von Kategorien ist.[23] Diese Kategorien nimmt

[22]Er verwendet dazu eine formale Notation, die hier nicht eingeführt wird.
[23]Dies ist stark vereinfacht. Zum Konzept der semantischen Typen siehe Abschnitt 4.7.

das Head als Komplemente. Auf diese Weise wird die Relation zwischen Subkategorisierung und Komplementen aus den ID-Regeln in die lexikalischen Kategorien verlagert. Mithilfe des *Komplement-Typ-Prinzips* (CTP) lassen sich die Komplemente eines lexikalischen Heads rekonstruieren. Das CTP wird durch schematische CCRs definiert. Vereinfacht ausgedrückt, besagt es:

- Jeder Head X[bar:0, typ:$\langle\delta_1, \ldots, \delta_{n-1}, \delta_n\rangle$] hat in einem lokalen Baum die Mutter X[typ:$\langle\delta_n\rangle$].
- Wenn in einem lokalen Baum ein Head [bar:0, typ:$\langle\delta_1, \ldots, \delta_{n-1}, \delta_n\rangle$] vorkommt, so kommen auch X[typ:$\langle\delta_i\rangle$] ($1 \leq i < n$) als Töchter vor und keine weiteren.

Damit besteht die Reformulierung einer Menge von ID-Regeln aus den Ästen, dem CTP und geeigneten CCRs für nicht-lexikalische ID-Regeln.

- Die Beschreibung lexikalischer ID-Regeln durch das CTP kann nun ausgenutzt werden, um Metaregeln zu eliminieren. Dies soll nun am Beispiel der Passiv-Metaregel skizziert werden. Der Effekt der Passiv-Metaregel ist, wie gesagt, zusammen mit einer Typanpassungsregel im Lexikon zu betrachten. Kilbury ersetzt beide durch eine „metalexikalische" Regel. Metalexikalische Regeln sind Ausdrücke der Form a =ß, wobei *a* und *ß* Kategorienschemata sind, die Variablen in den Merkmalwerten enthalten dürfen. Eine metalexikalische Regel besagt: Wenn im Lexikon ein Eintrag *a'*—> *w* existiert, wobei *a* und *a'* matchen (d.h. Variablen in a werden—analog etwa zur Prolog-Unifikation—durch entsprechende Konstanten in *a'* gebunden), dann existiert auch eine Eintrag *ß'* —> *w,* wobei *ß* und *ß'* matchen. Natürlich muß sichergestellt sein, daß die korrekten Wortformen vorliegen; Kilbury setzt dafür separate morphologische Regeln an (vgl. Abschnitt 2.4).

Die metalexikalische Regel für Passiv nimmt die entsprechenden Anpassungen des Merkmals typ vor.

(30) V[−pas, agr:δ_n, typ:$\langle\delta_1, \ldots, \delta_{n-1}, \delta_n$,S$\rangle$]

$\Downarrow$

V[+pas, agr:δ_{n-1}, typ:$\langle\delta_{n'}, \delta_1, \ldots, \delta_{n-1}$,S$\rangle$]

und

V[+pas, agr:δ_{n-1}, typ:$\langle\delta_{n'}, \delta_1, \ldots, \delta_{n-1}$,S$\rangle$]

wobei

(i) $\delta_{n-1}, \delta_n \in$ {NP, S}

(ii) wenn δ_n = NP, dann $\delta_{n'}$ = PP[by]; sonst $\delta_{n'}$ = S

Dabei wird ein binäres Merkmal pas zugrundegelegt, das Aktiv und Passiv unterscheidet. d_{n_i} und d_n sind die Kategorien des direkten Objekts bzw. des Subjekts der aktiven Form. Das CTP besagt nun, daß V[-pas] als Tochter einer VP mit dem Typ (d_n,S) genau die Komplemente d_i (l < *i* < *n* — 1) besitzt, während V[+pas] als Tochter einer VP mit dem Typ {d_{n-1},S) genau die Komplemente 6, (l ≤ *i* ≤ *n* — 2) und (optional) d_n, besitzt.

Der wesentliche Unterschied zwischen diesem Vorschlag und der Argumentation von Jacobson, welche mit einem HPSG-ähnlichen Ansatz vereinbar ist, besteht darin, daß Kilbury Metaregeln eliminiert, ohne das Subkategorisierungskonzept von GKPS anzutasten. Kilbury zeigt nebenbei die Redundanz auf, die hinsichtlich der

Kodierung von Komplementen eines lexikalischen Heads in ID-Regeln und semantischen Typen steckt. Er setzt allerdings voraus, daß die Werte von typ auf konsistente Weise instantiiert werden können—eine Annahme, die in GKPS nicht ohne weiteres haltbar ist (vgl. Abschnitt 4.7).

Im Berliner GPSG-System werden Metaregeln (in der oben beschriebenen, beschränkten Form) aus rein praktischen Gründen verwendet, nämlich als Arbeitserleichterung für das Regelschreiben. Ich komme darauf in Teil II zurück.

4.5 Zulässige Bäume

In Definition 9 wurde eine Reihe von Bedingungen formuliert, die eine *direkte Ableitung* definieren. Dieses Konzept wurde verwendet, weil es aus der Theorie formaler Sprachen bekannt ist. In diesem Abschnitt wird stattdessen wie in GKPS mengentheoretisch vorgegangen und die Menge der lokalen Bäume definiert. Infolge der genannten Bedingungen wird diese Menge dann eingeschränkt auf die Menge der *zulässigen lokalen Bäume.* In einem weiteren Schritt wird, analog zur Definition 10 (Ableitung), die Menge der *zulässigen Bäume* definiert.

Zunächst soll der Zusammenhang zwischen ID-Regeln und lokalen Bäumen genauer bestimmt werden. Eine ID-Regel läßt alle lokalen Bäume zu, deren Kategorien Extensionen der entsprechenden Kategorien in der ID-Regel sind. Zusätzlich müssen für jede Kategorie in diesen Bäumen alle FCRs wahr sein. Dann heißt jeder dieser Bäume *projiziert* durch die ID-Regel.

Definition 12. Sei $r = \langle C_0, \{C_1, \ldots, C_n\}_{mul}\rangle$ eine ID-Regel, und sei ferner $t = \langle C'_0, \langle C'_1, \ldots, C'_n\rangle\rangle$ ein etikettierter lokaler Baum. Dann heißt eine bijektive Funktion $\phi : \{C_0, \ldots, C_n\}_{mul} \rightarrow \{C'_0, \ldots, C'_n\}_{mul}$ *Projektion* gdw.

- $\phi(C_0) = C'_0$
- $\forall i, 0 \leq i \leq n : \phi(C_i)$ ist legal
- $\forall i, 0 \leq i \leq n : C_i \sqsubseteq \phi(C_i)$

Man beachte, daß die Information in lokalen Bäumen sich von der in ID-Regeln auf zwei Weisen unterscheidet. Einmal enthalten die Kategorien gewöhnlich mehr Merkmalspezifikationen und zum ändern existiert eine Reihenfolge der Töchter (m.a.W. erhält nicht die Ordnung der Indizes der Töchter aufrecht).[24]

Definition 13. Sei *r* eine ID-Regel. Dann heißt ein lokaler Baum *t lokal zulässig durch r* gdw. er die folgenden Eigenschaften aufweist:

- Es gibt eine Projektion mit *(r)* = *t.*[25]
- Für alle Kategorien *C* € *r* erfüllt *(C)* alle FSDs.

[24][Hendriks 1986] kritisiert, daß dies nicht aus der Definition der Projektionsfunktion folgt. Es wird lediglich eine Abbildung von Multimengen in Multimengen definiert, nicht aber von Multimengen in Bäume. Auch wenn man eine Bijektion zwischen Elementen des Bildbereichs und lokalen Bäumen als gegeben voraussetzt, bleibt ein Problem: die Töchter in Bäumen sind geordnet, die Elemente einer Multimenge jedoch nicht. Für eine Lösung verweise ich auf [Hendriks 1986].

[25]Diese etwas freizügige Schreibweise übernehme ich von GKPS. Gemeint ist: ^ ist eine Projektion von einer ID-Regel *r* auf einen lokalen Baum *t.*

- erfüllt die MIPs (d.h. das FFP, das GAP und die HFC).
- induziert eine Folge von Töchtern, die zulässig gemäß LP ist.

Leider ergeben sich aus der in GKPS vorgenommenen Aufgabenverteilung eine Reihe von Abhängigkeiten, die diese einfache Menge von Bedingungen zu einem extrem komplexen und unübersichtlichen Netz von Constraints werden lassen. Ich werde die formalen Definitionen in dieser Arbeit nicht alle einführen, sondern auf die Schwierigkeiten eingehen, die sich aus den Aufgaben der MIPs ergeben. An dieser Stelle sei darauf hingewiesen, daß die Anwendungsbedingungen der FSDs ebenfalls relationalen Charakter erhalten insofern, als sie die Effekte der MIPs nachspielen müssen. Die Aufgabe von Defaults ist ja gerade, nicht den anderweitig geforderten Merkmalspezifikationen zu widersprechen vgl. [GKPS, S. 101-104]).

Im folgenden wird der zentrale Begriff der GPSG definiert, nämlich der des zulässigen Baums. Ein Baum heißt *terminiert* gdw. jedes Blatt mit einem Terminalsymbol etikettiert ist. GKPS gehen einfach davon aus, daß terminierte lokale Bäume aufgrund des Lexikons existieren, zulässig sind und der Syntax zur Verfügung stehen.

Definition 14. Sei *R* eine Menge von ID-Regeln. Ein Baum *t* heißt *zulässig in R* gdw. *t* terminiert ist und jeder lokale Teilbaum in *t* entweder terminiert ist oder lokal zulässig durch ein r € *R.*

In die Definition der lokalen Zulässigkeit wurden die MIPs einbezogen. Die folgenden Abschnitte veranschaulichen die intendierte Funktionsweise und weisen auf theoretische Probleme hin.

4.6 Das Foot-Feature-Prinzip

In den letzten Abschnitten wurden metasprachliche Komponenten beschrieben, die eine eigene Syntax besitzen: Merkmalspezifikationen, komplexe Kategorien, FCRs, ID-Regeln, LP-Aussagen und Metaregeln. In diesem und den folgenden beiden Abschnitten werden die MIPs beschrieben. Als universelle Prinzipien, die die Spezifikation bestimmter Merkmale in lokalen Bäumen regeln, beschränken sie die Menge der zulässigen lokalen Bäume. Die MIPs besitzen keine Syntax und sind daher ausschließlich durch ihre Semantik, d.h. durch ihre Wirkung auf lokale Bäume bestimmt.

In Abschnitt 3 wurden die MIPs stark vereinfacht definiert, indem lediglich jeweils eine Menge von kospezifizierten Merkmalen in den Kategorien eines lokalen Baumes gefordert wurde. Nun werden diese Definitionen präzisiert und linguistisch motiviert. Es werden die Bedingungen beschrieben, unter denen ein Merkmal aus FOOT, CONTROL oder HEAD dem jeweiligen MIP genügen muß bzw. von ihm ausgenommen ist.

Dieser Abschnitt geht auf das FFP ein. Die Foot-Merkmale werden in GKPS in der Menge FOOT zusammengefaßt. In der Grammatik des in GKPS behandelten englischen Fragments sind drei kategorienwertige Merkmale, nämlich wh, re und slash. Die ersten beiden werden benutzt, um bestimmte Frage-, Relativ- und Reflexivpronomina zu behandeln. Mit slash werden Kategorien beschrieben, die nicht an ihrer kanonischen Position in der syntaktischen Struktur auftreten. Auf diese Weise

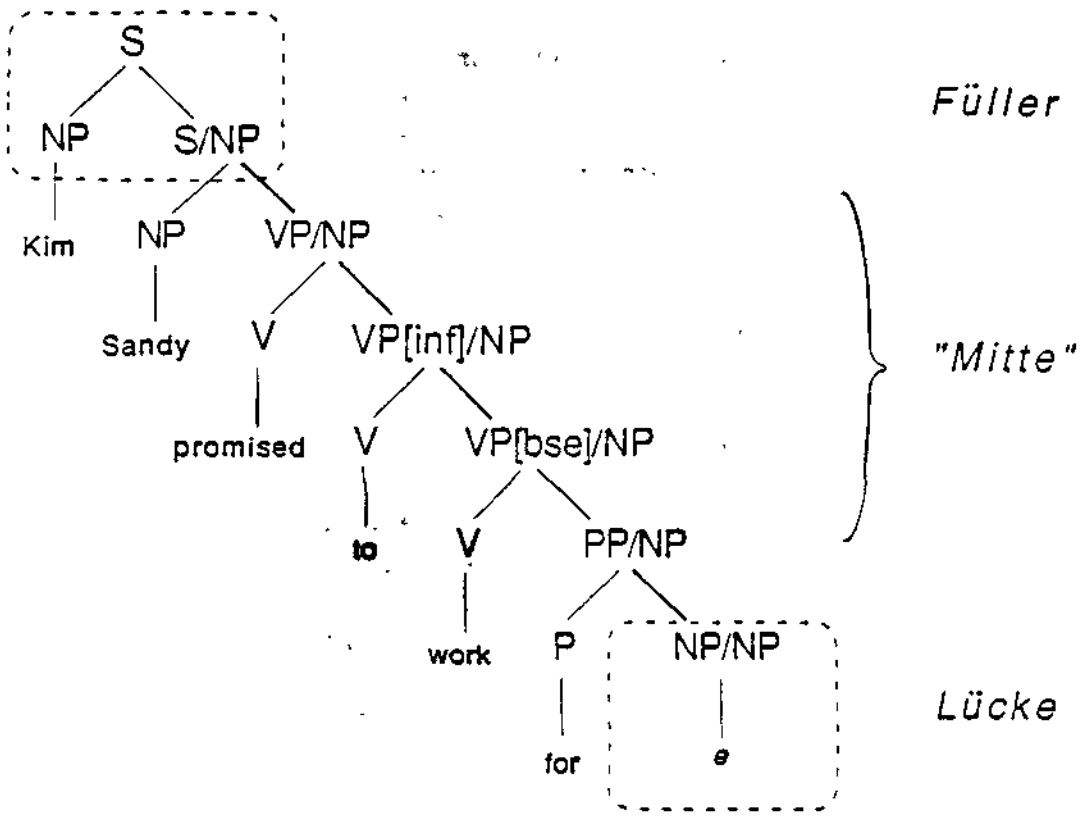

Abbildung 4: Topologische Bereiche bei Füller-Lücke-Konstruktionen

wird in GKPS die Gruppe der Füller-Lücke-Konstruktionen (engl. *filler-gap constructions)* behandelt. Ein Beispiel soll die Wirkungsweise des FFP verdeutlichen. Anschließend folgt eine Definition des FFP.

Die ID-Regeln in (32)-(37) ermöglichen Strukturbeschreibungen für Sätze wie (31). Der Lexikoneintrag (38) führt die Lücke *e* ein, die durch die erste Tochter von Regel (32) „gefüllt" wird. Der Füller ist eine phrasale Kategorie, die in relevanten Merkmalspezifikationen identisch mit dem slash-Wert an der Schwester ist. Es wird versucht, dies durch Kontrollmechanismen sicherzustellen, über die in Abschnitt 4.7 zu sprechen sein wird.

(31) Kim Sandy promised to work for.

(32) S—> [bar: 2], S/[bar: 2]

(33) S—>[bar:2],VP

(34) VP—>V,VP[inf]

(35) VP—>V,VP[bse]

(36) VP—>V,PP

(37) PP—>P,NP

(38) X / X —> e

Abbildung 4 zeigt eine Struktur, in der man topologisch drei Bereiche unterscheiden kann. Zwei davon werden durch die beiden Bäume gebildet, die Lücke und Füller einführen. Die sie projizierenden ID-Regeln bzw. Lexikoneinträge enthalten eine Spezifikation von slash; der Lexikoneintrag an der Mutter und die ID-Regel an einer Tochter. Der dritte Bereich liegt, topologisch gesehen, in der Mitte und wird durch eine unbestimmte Anzahl von ID-Regeln erzeugt, in denen slash nicht spezifiziert ist (vgl. (32)-(37)).

Eine Merkmalspezifikation (m : w) in einer Kategorie *(C)* heißt *ererbt* gdw. (m : w) € *C.* Merkmalspezifikationen in (C), die nicht ererbt sind, heißen *instantiiert.*

Wie in Abbildung 4 ersichtlich, sind in den lokalen Bäumen des mittleren Bereichs an den Kategorien auf dem „Pfad" zwischen Lücke und Füller Spezifikationen für slash instantiiert. Diese Spezifikationen explizieren die Korrespondenz zwischen Füller und Lücke. Sowohl ihre Präsenz wie auch ihre Identität werden durch das FFP erzwungen. Die ererbten slash-Werte an Füller und Lücke werden vom FFP jedoch nicht berührt.

Grundsätzlich kann eine Lücke mehrfach auftreten (bei parasitären Lücken im Englischen oder bei Koordination). Das FFP soll auch diese Fälle erfassen. Es muß sicherstellen, daß die Pfade vom jeweils unteren Bereich an einem Knoten im mittleren Bereich zusammentreffen und daß von dort aus nur ein Pfad bis zum oberen Bereich führt. Der Treffpunkt ist an der Mutter eines lokalen Baumes, und das FFP muß die Bedingungen über das Zusammentreffen ausdrücken.

Dies wird in der folgenden Definition angestrebt (eine formale Darstellung findet sich in [GKPS, S. 82]).

Definition 15. Ein lokaler Baum genügt dem FFP, wenn der Wert jedes instantiierten FOOT-Merkmals m an der Mutter und die Unifikation aller instantiierten Spezifikationen von m an den Töchtern identisch sind.

Die lokalen Bäume in Abbildung 4 genügen dem FFP. Bei lexikalischen Kategorien ist die Instantiierung von slash in GKPS durch die FCR (7) auf Seite 29, die hier als (39) wiederholt wird, in Verbindung mit der FCR (40) verhindert.[26]

(39) (subcat) *D* -.(slash)

(40) (bar: 0) = (n) A (v) A (subcat)

Bei nicht-lexikalischen Töchtern ohne ererbte slash-Spezifikation fordert das FFP keine instantiierte Spezifikation, läßt aber jede zu, die mit denen der anderen Töchter unifizierbar ist (vgl. [GKPS, S. 140ff]). Dies kann u.a. bei flachen Strukturen

[26]Eine Anmerkung zum Begriff der *lexikalischen Kategorie* in GKPS: FCR (40) besagt, daß die legalen Extensionen von bzgl. subcat spezifizierter Kategorien genau diejenigen sind, die für n und v spezifiziert sind und (bar : 0) enthalten. Die Bezeichnung *lexikalisch* trifft somit nicht auf alle Mütter terminaler Bäume zu, insbesondere auf Det nicht. Das ist aber offensichtlich nicht die Intention von GKPS, wenn sie schreiben:

> A lexical item belongs to a minor lexical category if and only if it is specified for subcat, but not specified for bar. [GKPS, S. 35]

Daher verwende ich *lexikalisch* weiterhin für alle Kategorien, die ein Terminalsymbol dominieren können.

zu unerwünschten Analysen führen, wenn die mögliche Spezifikation von FOOT-Merkmalen nicht eingeschränkt wird (etwa durch FCRs). ID-Regeln mit mehr als einer Tochter, die kein lexikalischer Head ist (41)-(43), können zulassen, daß mehrere Pfade aufeinander treffen. Damit würden Sätze wie (44)-(46) als grammatisch klassifiziert.[27]

(41) VP—> V,NP,VP[inf]

(42) VP—>V,NP,PP

(43) s —> V[+aux],NP,VP[bse]

(44) * Sandy Kim promises *e* to wash *e.*

(45) * Fido Kim gives *e* to *e.*

(46) * Who did a picture of *e* bother Mary?

GKPS möchten einzelsprachliche Koreferenz-Beschränkungen (im Sinne von [Engdahl 1980]) heranziehen, um ähnliche Sätze als ungrammatisch zu kennzeichnen [GKPS, S. 166].

Ein weiteres Problem ist, daß die *Unifikation* der FOOT-Spezifikationen unerwünschte Mehrfachanalysen bei mehrfach auftretenden Lücken zuläßt. Die instantiierten Spezifikationen im slash-Wert können in vielfältiger Weise auf die unterschiedlichen Vorkommen der Lücke verteilt sein. Die Intuition, daß *eine* Lücke mehrfach auftreten kann, würde genauer verwirklicht durch die rigidere Forderung im FFP nach *Identität* von instantiierten FOOT-Spezifikationen.

Die Wirkung des FFP ist andererseits zu strikt, indem lokale Bäume, die instantiierte slash-Spezifikationen für verschiedene Lücken enthalten, aufgrund der Kategoriendefinition ausgeschlossen sind. Es ist nicht schwer, englische Beispiele zu finden, für die es naheliegt, Konstituenten mit verschiedenen Lücken anzunehmen (für Beispiele aus skandinavischen Sprachen zitieren GKPS [Engdahl 1980]). In Abbildung 5 liegt eine vereinfachte Analyse im Stile von GKPS vor (wobei es für das Argument keine Rolle spielt, welche Konstituenten *argue* subkategorisiert). Die verschiedenen NPs werden nur zur Unterscheidung indiziert.[28]

Die technische „Lösung" besteht in der Annahme eines zusätzlichen kategorienwertigen FOOT-Merkmals. Offen bleibt, wie dessen Spezifikationen instantiiert werden sollen, so daß keine unakzeptablen Sätze von der Grammatik analysiert werden.

4.7 Das Control-Agreement-Prinzip

Das CAP ist neben den FSDs die wohl umstrittenste Komponente der Metasprache, denn es ist nicht nur wegen seiner barocken Definition schwer durchschaubar sondern auch inadäquat hinsichtlich einer Reihe sprachlicher Fakten.

Keenans semantisch motiviertes Prinzip, daß, grob vereinfachend, Argumente

[27]Beispiel 46 stammt aus [Preuß 1989, S. 35].

[28]Der Beispielsatz stammt von Hukari und Levine, die ausführlich verschiedene Probleme mit parasitären Lücken diskutieren, die das Zusammenwirken von FFP und CAP betreffen [Hukari/Levine 1986, S. 238ff].

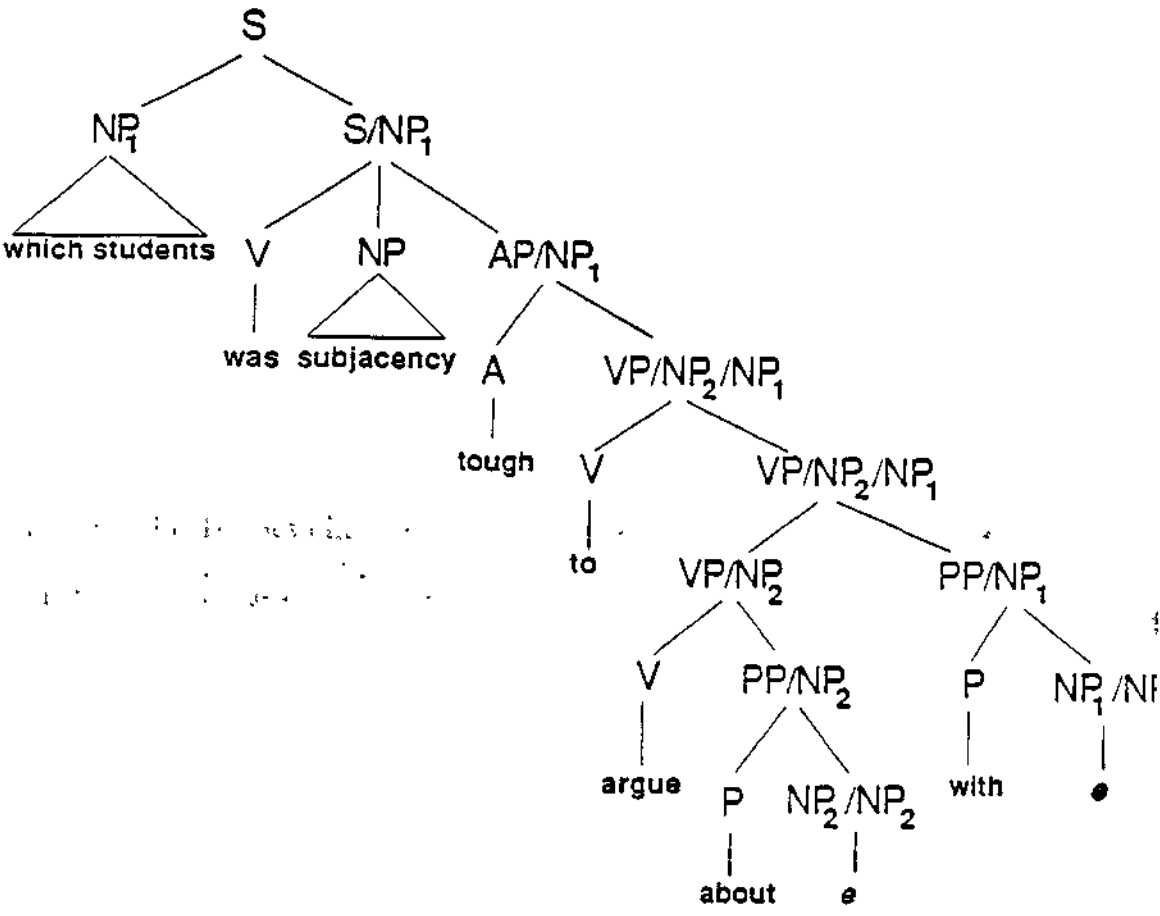

Abbildung 5: Unerlaubtes Zusammentreffen von slash-Spezifikationen

Funktoren kontrollieren können, wird in GPSG mithilfe semantischer Typen formalisiert und verallgemeinert. Die Typen ermöglichen die Rekonstruktion von Funktor-Argument-Verhältnissen auf der syntaktischen Ebene. Die Typen bilden die Grundlage für ein Konzept von *Kontrolle* als Beziehung zwischen der kontrollierenden Kategorie (Controller) und der kontrollierten Kategorie (Controllee).[29] Diese Kategorien *können* miteinander kongruieren. Das CAP bildet die Grundlage für eine Beschreibung, welche Kategorien *tatsächlich* miteinander kongruieren. 1

Zu den Phänomenen, die GKPS auf der Basis des CAP behandeln, gehören die Kongruenz zwischen grammatischem Subjekt und finitem Verb sowie zwischen logischem Subjekt (im Sinne von [Bußmann 1983]) und Objekt in Equi- und Raising-Konstruktionen.[30] Es ist wichtig zu beachten, daß GKPS in den Strukturen für (48)-(49), in denen *Ann* das logische Subjekt von *wash* darstellt, keine eigene syntaktische Repräsentation ansetzen; ebenso im Falle von (50), wo *Jim* als logisches Subjekt von *wash* verstanden wird. Die Verben *persuade* und *promise* haben die gleichen Schwestern (vgl. Abbildung 6), kombinieren aber semantisch auf verschiedene Weise mit ihnen.

Dies wird durch unterschiedliche semantische Typen bewirkt. Das CAP soll in

[29]Ich übernehme die englischen Termini in Ermangelung besserer Alternativen.
[30]Diese Begriffe sind im Sinne der TG zu verstehen.

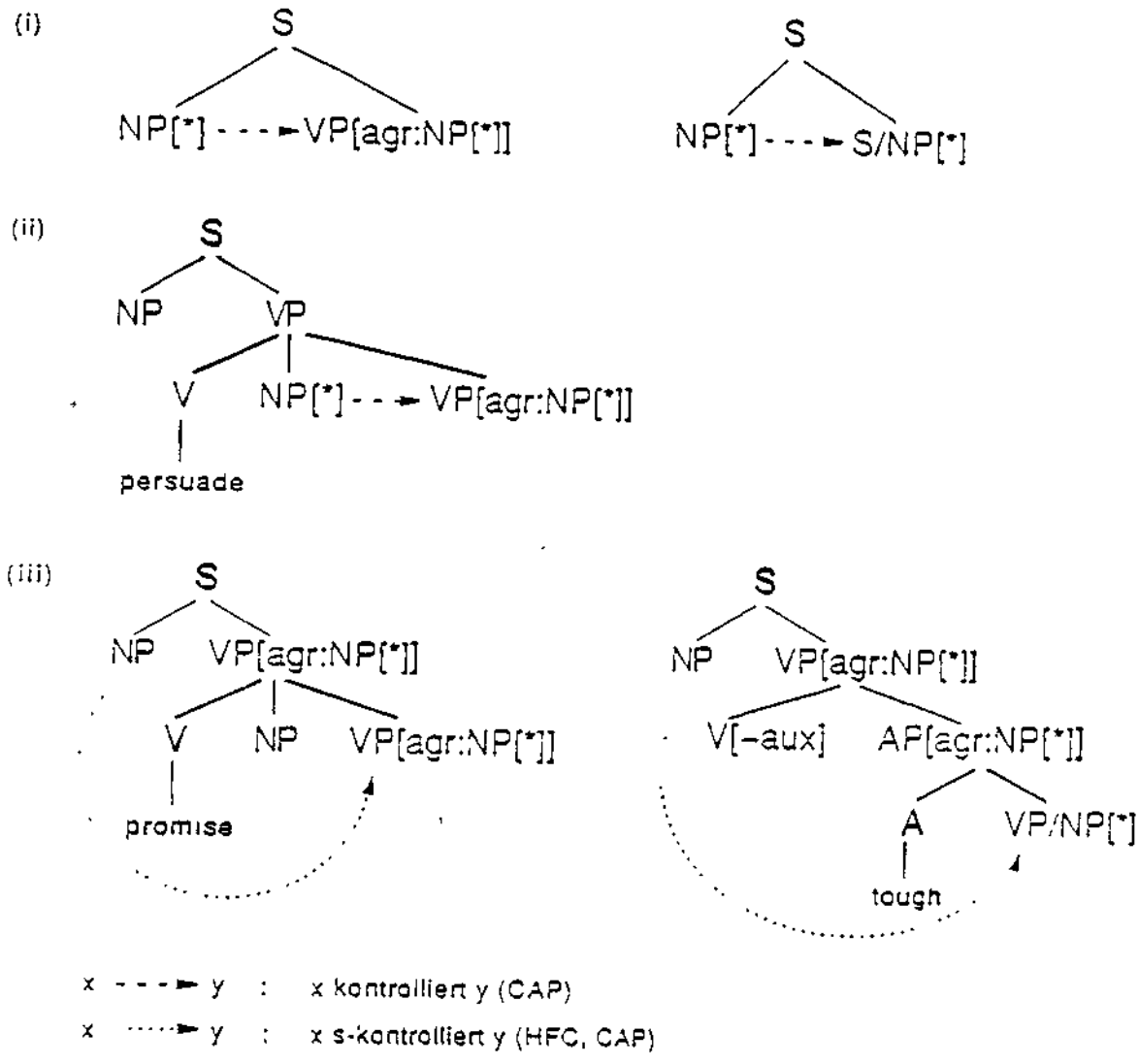

Abbildung 6: Kontrollbeziehungen zwischen Kategorien in lokalen Bäumen

diesem Abschnitt anhand der folgenden Beispiele beschrieben werden, die die wesentlichen Kontrollbeziehungen enthalten.

(47) Jim washes his car.

(48) Ann Jim persuades to wash herseif.

(49) Jim persuades Ann to wash herseif (* himself).

(50) Jim promises Ann to wash himself (* herseif).

(51) Jim is easy to please.

Ich führe erst die semantischen Typen ein, definiere dann Kontrolle und gebe schließlich eine Definition des CAP.[31] Danach werden eine Reihe von Problemen mit diesem Ansatz diskutiert.

[31]Ich orientiere mich an der Definitionsweise von [Jacobson 1987], die ich für transparenter halte als die Originalversion von GKPS.

4.7.1 Semantische Typen

GKPS übernehmen das Konzept einer modelltheoretischen Semantik nach Montague. Obwohl sie die direkte Abbildung von syntaktischen Kategorien in Mengen möglicher Denotate für möglich halten, bevorzugen sie (wie auch Montague) eine indirekte Vorgehensweise mithilfe einer formalen Sprache, nämlich intensionaler Logik (IL). Dabei steht das Problem der Abbildung natürlichsprachlicher Ausdrücke in IL-Ausdrücke im Mittelpunkt, denn IL ist so definiert, daß die Beziehung zu modelltheoretischen Interpretationen bekannt ist.

Semantische Typen sind die syntaktischen Kategorien von IL. Sie werden benutzt, um die semantische Rolle der GPSG-Kategorien darzustellen. Die primitiven Typen von IL sind „e" für *entity* und „t" für Wahrheitswerte. Komplexe Typen werden induktiv definiert:

- Wenn *a* ein Typ ist, dann auch (s,a).
- Wenn a und *b* Typen sind, dann auch (a, 6).

Jeder syntaktischen Kategorie wird—irn Unterschied zu Montague nicht eindeutig—ein Typ zugewiesen. S hat z.B. den Typ *(s, t)* und NP (5, {{e, *),*)) (vgl. [GKPS, S. 188ff]). Man schreibt dies mithilfe der Funktion *TYP*, die aus der Menge syntaktischer Kategorien in die Typen von IL abbildet.

Typen bestimmen, wie Denotate komplexerer IL-Ausdrücke sich als Funktion aus den Denotaten ihrer Konstituenten ergeben. Infolgedessen ist z.B. *TYP(VP) = (TYP(NP), TYP(S))*, oder, abgekürzt VP' = (NP', S'). VP' ist eine Funktion, die Objekte aus NP' als Argumente nimmt und Objekte aus S' als Werte liefert.

4.7.2 Kontrollbeziehungen in GPSG

Semantische Typen werden für die Definition der Kontrollbeziehung herangezogen. In einer Kontrollbeziehung ist die Funktor-Kategorie Controllee und die Argument-Kategorie Controller. Die Funktor-Kategorie heißt, unabhängig von der Existenz einer Kontrollbeziehung, *Zielkategorie* (der Kongruenzrelation).

Die Kontrollbeziehung wird für zwei verschiedene Situationen definiert. Betrachten wir zuerst binäre lokale Bäume; sie haben eine Funktor-Kategorie *C* und eine Argument-Kategorie C_2 (ein Beispiel ist (i) in Abbildung 6). Es gilt $C\{ = (C'_2, C'_0)$ für eine Mutter-Kategorie C_0. Dann wird C_1 von C_2 kontrolliert. In dem genannten Beispiel ist VP die Funktor-Kategorie und wird von NP kontrolliert.

Die zweite Situation betrifft sogenannte *Kontrollprädikate.* Ein Kontrollprädikat ist ein lexikalischer Head, der Komplemente subkategorisiert, welche durch prädikative Kategorien (z.B. VPs) dargestellt werden. Diese haben den Typ (NP', C'), denotieren also Funktionen von NP-Denotaten. In Abbildung 6 werden die fett gezeichneten lokalen Bäume in (ii) und (iii) durch die ID-Regeln (41) auf Seite 43 (hier als (52) wiederholt) und (53) projiziert. Die VP-Töchter bzw. die AP sind prädikative Kategorien (vom Typ (NP', S')).

(52) VP—> V,NP,VP[inf]

(53) AP—>A,VP/NP

Kontrollprädikate sind Verben und Adjektive, die Hilfsverben sowie Equi- und Raising-Prädikate im Sinne der TG einschließen. Prädikative Komplemente von Kontrollprädikaten sind mögliche Zielkategorien, auch wenn keine Kontrollbeziehung vorliegt. Wir müssen also abermals zwei Situationen unterscheiden und betrachten zunächst die, in der die prädikative Kategorie Controllee ist.

In diesen Fällen *vermittelt* der lexikalische Head die Kontrollbeziehung zwischen einem NP-Controller und der prädikativen Kategorie. Solche Heads sind objektkontrollierte Equi- und „Raising-to-Object"-Verben (vgl. (ii) in Abbildung 6). Nach GKPS haben Kontrollvermittler den Typ (VP', (NP',VP')).

Dies sind die beiden einzigen Fälle, in denen eine Kontrollbeziehung definiert ist. Wir betrachten nun die Situation, in denen prädikative Kategorien nicht kontrolliert werden. Das V, das in (iii) von Abbildung 6 das Terminalsymbol *promise* dominiert, hat den Typ (NP', (VP',VP')); es vermittelt keine Kontrollbeziehung zwischen der NP und der VP, und wird von keinem seiner Argumente kontrolliert.[32]. Analog verhält es sich mit dem A in (iii). Die prädikative Kategorie VP / NP hat den Typ (NP', VP'), aber keinen Controller, denn der Typ von A ist ((NP', VP'), AP'), nicht aber (VP', AP'), wie es zur Herstellung einer Kontrollbeziehung notwendig wäre.

4.7.3 *x*-Spezifikationen

Die Verbindung der Kontrollrelation zu den semantischen Typen bringt eine Komplikation mit sich, da die Typen sich abhängig von instantiierten FOOT-Spezifikationen ändern können. Diese sollen aber keinen Einfluß auf die Definition von Kontrolle ausüben; z.B. soll in (ii) die NP auch dann die VP kontrollieren, wenn sie eine Lücke repräsentiert (d.h. NP / NP). Welche Merkmale sollen nun die semantischen Typen bestimmen, die für die Definition von Kontrolle relevant sind? GKPS geben (kommentarlos) an, daß dies die Head-Spezifikationen seien, die nicht zugleich instantiierte FOOT-Spezifikationen sind sowie zusätzlich die ererbten FOOT-Spezifikationen. Sie verwenden eine Funktion *x*, die zu einer Kategorie diese Teilmenge der Spezifikationen liefert und nennen diese Teilmenge *x-Spezifikationen*.

Die ererbten FOOT-Spezifikationen werden in den semantischen Typen berücksichtigt und sind relevant für die Kontrollbeziehung. Warum aber sind die Head-Spezifikationen für die Kontrollbeziehung relevant? Und warum sind von diesen genau die instantiierten FOOT-Spezifikationen irrelevant?

Mit dem CAP sollen Kongruenzbeziehungen zwischen Terminalsymbolen erklärt werden. Deren morphologische Eigenschaften können wir uns als in den lexikalischen Kategorien repräsentiert vorstellen. Daß diese Information an nonterminalen Kategorien auftritt (z.B. Merkmale für Numerus, Genus und Person außer an N auch an NP), stellt die HFC sicher (vgl. Abschnitt 4.8), die genau für die Head-Merkmale erfüllt sein muß. Unklar ist aber, ob wirklich *alle* Head-Merkmale relevant sind. Auf die Arbeitsteilung zwischen CAP und HFC bei der Realisierung von Kongruenz werde ich gleich zurückkommen.

[32]Der gestrichelte Pfeil drückt eine andere, semantische Kontrollbeziehung aus (in Abbildung 6 S-Kontrolle genannt); man spricht hinsichtlich der VP vom Subjekt des Matrixsatzes als dem „verstandenen Subjekt"

Mit dem Ausschluß instantiierter FOOT-Spezifikationen, die zugleich Head-Spezifikationen sind, erreichen GKPS zwar, daß das CAP in ihrer Grammatik keine instantiierten slash-Werte betrachtet, denn slash ist das einzige FOOT-Merkmal, das zugleich Head-Merkmal ist. Es bleibt aber offen, inwieweit die Eigenschaft, sowohl instantiierte FOOT- als auch Head-Spezifikation zu sein, im allgemeinen etwas mit der Zuordnung semantischer Typen zu Kategorien zu tun hat.

4.7.4 Kontrollbeziehung und Kongruenz

Durch die folgende Definition werden die Fälle der Kontrolle in binären Bäumen von denen der Kontrollvermittlung unterschieden.[33]

Definition 16. In einem lokalen Baum $((C_0, (C_1, \ldots, C_n\}}$ *kontrolliert* eine Kategorie *Ci* eine Kategorie *Cj* $(1 \leq i,j \leq n)$ gdw. entweder

- $TYP(x(C_i)) = (TYP(x(C,),TYP(x(C_0)),oder$
- $TYP(x(C_i))$ = VP' und die Head-Schwester hat den Typ $(VP',(TYP(x(C_J),VP'))$

Kontrollbeziehungen geben an, welche Kategorien miteinander kongruieren *können,* nicht aber, welche es tatsächlich tun. Die Kongruenz wird realisiert mithilfe eines Kontrollmerkmals. Jede Zielkategorie enthält ein kategorienwertiges syntaktisches Merkmal agr oder slash. Diese beiden Merkmale sind Elemente der Menge CONTROL, die eine Teilmenge der Merkmale ist (in der Grammatik von GKPS gilt CONTROL = {agr, slash}). Die Kontrollmerkmale sind auch Head-Merkmale. Dies wird nicht im Formalismus verlangt, doch ist es erforderlich, um die gewünschte Arbeitsteilung zwischen CAP und HFC zu erreichen.

Die Aufgabe der HFC ist, wie oben bereits angedeutet, die folgende: Der Lexikoneintrag für eine Wortform (z.B. *persuades)* enthält eine Extension von [agr:NP[per:3, -plu]]. HFC erzwingt das Vorkommen dieser Spezifikation an der Mutter (VP in (ii) von Abbildung 6). Aufgabe des CAP ist es, sicherzustellen, daß die Mutter, vereinfacht gesagt, mit einer Kategorie übereinstimmt, die als Wert ihrer agr-Spezifikation kodiert ist. Am Beispiel sehen wir, daß *persuades* nur mit einer NP[per:3, -plu] kongruiert. Die Kongruenz bezieht sich ausschließlich auf die *x*-Spezifikationen; alle anderen sind in der agr-Spezifikation nicht enthalten.

4.7.5 Die Definition des CAP und einige Beispiele

Bevor die Definition des CAP gegeben wird, sind die Kriterien für die Auswahl des Kontrollmerkmals zu beschreiben. GKPS definieren:

Definition 17. Sei *C* eine Kategorie in einem lokalen Baum und C(bar) /=/ 0. Das *Kontrollmerkmal von C* slash gdw. slash an *C* ererbt ist; sonst ist es agr.

Definition 18. Ein lokaler Baum (Co, $(C\backslash,\ldots, C_n))$ genügt dem CAP gdw. entweder

[33]Eine formale Definition steht in [Hukari/Levine 1986, S. 164]; die entsprechende Definition von GKPS enthält einige Fehler.

1. *Cj* eine Kategorie C, im binären lokalen Baum kontrolliert und der Wert des Kontrollmerkmals von C_i, mit den *x*-Spezifikationen von *Cj* identisch ist; oder
2. C_k, eine Kontrollbeziehung zwischen *Cj* als Controller und C_i als Controllee vermittelt und der Wert des Kontrollmerkmals von C, mit den *x*-Spezifikationen von *Cj* identisch ist; oder
3. eine prädikative Kategorie C_i, ohne Controller existiert, und der Wert des Kontrollmerkmals von C_i mit dem Wert des Kontrollmerkmals von C_0 identisch ist.

Betrachten wir noch einmal Abbildung 6. Im linken lokalen Baum von (i) ist das Kontrollmerkmal des Controllees agr, da keine ererbte slash-Spezifikation vorliegt. Die erste Klausel des CAP verlangt, daß der Wert von agr (modulo *x)* mit *dem* Controller identisch ist. In Zusammenarbeit mit der HFC wird so z.B. (47) generiert. Im rechten lokalen Baum von (i) ist slash das Kontrollmerkmal des Controllees. Dieselbe Klausel des CAP stellt offenbar sicher, daß Füller und Lücke hinsichtlich der *x*-Spezifikationen gleich sind (vgl. Abschnitt 4.6). Dies ist eine Konsequenz aus der obigen Definition 17. Ein Beispiel hierfür ist (48) auf Seite 45, worauf wir noch zurückkommen.

In (ii) liegt ein Fall der Kontrollvermittlung vor. Infolgedessen müssen gemäß der zweiten Klausel des CAP die *x*-Spezifikationen der NP mit dem agr-Wert der prädikativen Kategorie übereinstimmen (man beachte, daß dies unabhängig von einer an der NP instantiierten slash-Spezifikation gilt; der agr-Wert ist für slash Undefiniert). Auf diese Weise wird der Bezug zwischen dem Objekt und der prädikativen Kategorie hergestellt, der für (51) wesentlich ist.

In (iii) links liegt im fett gezeichneten lokalen Baum keine Kontrollbeziehung vor. Es gibt jedoch eine prädikative Kategorie (VP), so daß die dritte Klausel des CAP Anwendung finden kann. Durch die Kospezifikation der Kontrollmerkmale an Mutter und Tochter wird, sofern das CAP im—topologisch gesehen—höheren lokalen Baum nach Klausel l erfüllt ist, die Beziehung zwischen dem Subjekt des Matrixsatzes und dem logischen Subjekt der durch die prädikative Kategorie dominierten Konstituente hergestellt. Dies ist notwendig, um (50) zu akzeptieren. Auf analoge Weise erfüllt im rechten Baum von (iii) der fett gezeichnete lokale Baum das CAP, doch sind die Kontrollmerkmale hier verschieden. Ein Beispiel ist (51).

Das intendierte Zusammenwirken von CAP und HFC sowie zwischen CAP und FFP faßt Abbildung 7 zusammen, worin eine Struktur für das Beispiel (48) angegeben wird.

GKPS beschreiben nicht, wie aufgrund der agr-Spezifikation an der infiniten VP tatsächlich das Terminalsymbol *herself* (und nicht etwa *himself)* erzwungen wird. Unter der Annahme der in der Abbildung gezeigten Strukturen ergibt sich folgende Situation: Die VP[bse] ist eine prädikative Kategorie, weshalb die dritte Klausel des CAP gleich spezifizierte Kontrollmerkmale fordert. In dem lokalen Baum, in dem das transitive Verb *wash* eingeführt wird, kontrolliert die NP das V nach der Definition für binär verzweigende Bäume. Das V hat als lexikalische Kategorie jedoch kein Kontrollmerkmal, so daß die erste Klausel des CAP nicht anwendbar ist. Die dritte

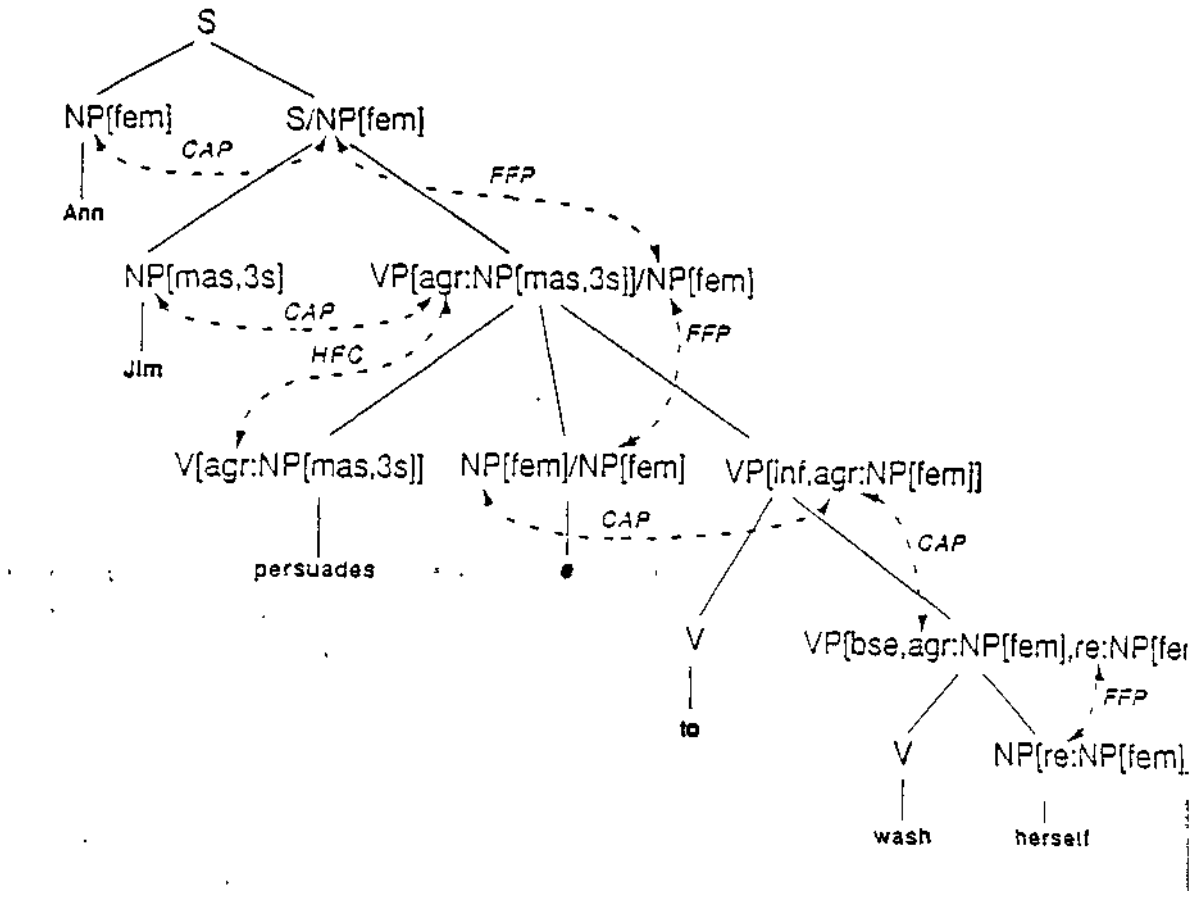

Abbildung 7: CAP und das Zusammenwirken der MIPs

Klausel ist ebenfalls nicht anwendbar, denn die NP ist keine prädikative Kategorie Für diesen lokalen Baum ist das CAP mithin auf eine triviale Weise erfüllt.

Die Eigenschaften des Reflexivpronomens werden in dem kategorienwertige FOOT-Merkmal re kodiert. Diese Spezifikation tritt nach dem FFP an der VP[bs< auf. GKPS erwähnen:

> Agreement principles guarantee that these reflexive or reciprocal elements agree with the agr value of the VPs that contain them (in a significant class of cases). [GKPS, S. 122]

Es ist nun aber nicht ersichtlich, wie die Werte von agr und re miteinander abgegl chen werden können.[34] Darüber hinaus muß untersucht werden, wie das Vorkomme von re zu beschränken ist (das FFP läßt mehr oder minder beliebige Vorkommen de Spezifikation zu). Es bleiben also eine Reihe von Fragen offen, und es ist festzuhalter daß das CAP nicht alle Kongruenzphänomene abdeckt.

4.7.6 Weitere Probleme mit dem CAP

Im folgenden wird eine Reihe von Problemen mit dem CAP diskutiert, die sein linguistische Adäquatheit infrage stellen. Wie in den Beispielen aus Abbildung

[34]Eine re-Spezifikation könnte in der IL-Repräsentation eine geeignete Identität von Variable fordern; ein Vorschlag in dieser Richtung enthält [Pollard/Sag 1983]. Ein eigenständiges Bindungs prinzip wird in [Pollard 1984, S. 180] angenommen.

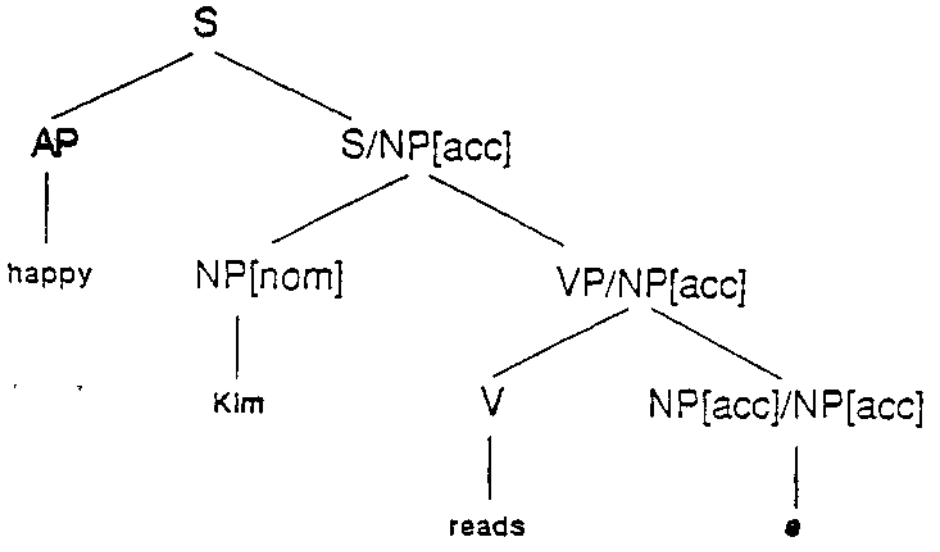

Abbildung 8: Probleme mit der Terminierung von slash

gezeigt wurde, ist das CAP für die Identität der *x*-Spezifikationen an der [bar:2]-Tochter und dem slash-Wert der S-Tochter in der Topikalisierungs-ID-Regel (vgl. (32) auf Seite 41) zuständig. Der semantische Typ von S / [bar:2] ist eine Funktion von Objekten des Typs [bar:2] in Objekte des Typs S. Infolgedessen enthält das Kontrollmerkmal (slash) die *x*-Spezifikationen der jeweiligen Extension von [bar:2]. Jacobson [Jacobson 1987, S. 410ff] zeigt, daß der Gedankengang zirkulär ist: der semantische Typ von S / [bar:2] hängt u.a. von den Spezifikationen für n und v ab. CAP soll diese Werte bestimmen, doch die Anwendbarkeit von CAP hängt von ihnen ab!

Abbildung 8 zeigt, daß lokale Bäume, in denen die erste Tochter eine AP und der slash-Wert eine NP ist, durch das CAP zugelassen werden. Da die AP und die NP verschiedene Typen haben, existiert kein Controller im lokalen Baum.[35]

Jacobson wendet sich gegen die Verwendung von slash als Kontrollmerkmal. Sie hebt den Zusammenhang mit der Einführung von Lücken durch eine Metaregel hervor. Diese Metaregel (Slash Termination Metarule 1) bewirkt zusammen mit einer FCR, daß im lokalen Baum slash-Werte instantiiert werden. Jacobson plädiert dafür, diese Metaregel aufzugeben. Einer ihrer Vorschläge verzichtet auf den die Lücke einführenden Lexikoneintrag (vgl. (38) auf Seite 41) und erzeugt aufgrund einer neuen Metaregel ID-Regeln mit einer *ererbten* slash-Spezifikation an der Mutter (54).[36]

(54) VP/NP—>V,VP

Zwei weitere Probleme nennt Kilbury.[37] Analog zu der Topikalisierungsregel (32) herrscht keine Kontrollbeziehung in bestimmten lokalen Bäumen, die durch die ID-Regel (33) projiziert werden (diese wird hier als (55) wiederholt). Eine AP als Ex-

[35]Analoges gilt für nicht subkategorisierte PPs (vgl. [Hukari/Levine 1986, S. 234]).

[36]Mit diesem Ansatz gerät man wahrscheinlich mit CAP in Schwierigkeiten bei Subjekt-Equi-Verben wie *promise,* denn die VP-Tochter kongruiert mit der dislozierten Konstituente (mit dem Objekt) statt mit dem Subjekt, weil nun slash das Kontrollmerkmal der Mutter ist.

[37]Persönliche Mitteilung, 1986.

tension der [bar:2]-Kategorie ist kein Controller im lokalen Baum, und infolgedessen verlangt das CAP nur gleiche agr-Spezifikationen von Mutter und VP. Dann können Sätze wie **Happy sleeps* analysiert werden. Abhilfe würde natürlich eine weniger unterspezifizierte Variante von (55) schaffen.

(55) S —> [bar : 2], VP

agr-Spezifikationen lexikalischer Controllees zählen nicht als Kontrollmerkmal. Das hat seinen guten Grund, weil nur so die Verb-Objekt-Kongruenz unterbunden werden kann (z.B. im untersten lokalen Baum von Abbildung 7)[38] Bei invertierten Sätzen führt dies zu Schwierigkeiten, denn ihnen wird die folgende ID-Regel zugrundegelegt (vgl. (43)).

(56) S[+inv] —> V[+aux], NP, VP[-aux, bse]

Das CAP ignoriert, welche Spezifikation der Head für agr besitzt, und stellt keine Kontrollbeziehung her. Es ist nicht ersichtlich, wie Analysen für Sätze wie **Does you come?* unterdrückt werden können.

Eine weitere Schwierigkeit ergibt sich mit der Beschränkung des Begriffs der Kontrolle auf solche Fälle, in denen der Funktor nicht mehr als ein Argument hat (mit Ausnahme der Fälle von Kontrollvermittlung).

Dies kann nicht aus der Definition der Kontrolle allein gefolgert werden, geht aber aus dem Zusammenhang mit den Bedingungen für korrekte Typzuweisung hervor, denen syntaktische Strukturen infolge des Prinzips der Funktionalen Realisierung (GKPS, Kap. 10) genügen müssen. Beide Teile der Theorie bewirken, daß ein Funktor nur dann durch ein Argument kontrolliert werden kann, wenn kein weiteres Argument vorliegt. Andernfalls ist der Funktor von einem Typ, der nicht von der Definition von Kontrolle erfaßt wird.

Diese Beschränkung bereitet besonders Schwierigkeiten, wenn flache Strukturen angenommen werden (wie dies aus unabhängigen Gründen in der Berliner Grammatik für das deutsche Fragment der Fall ist; vgl. Abschnitt 8). Dann ist nicht klar, welches der Argumente den Funktor kontrollieren soll. Im Fall der Subjekt-Verb-Kongruenz im Deutschen wäre das Subjekt als Controller zu kennzeichnen, was kaum auf der Grundlage der semantischen Typen alleine geschehen kann. Es gibt keinen semantischen Grund, Subjekt und Objekt durch ihren Typ zu unterscheiden; es sei denn, man will Subjekte als Funktoren betrachten, die über VPs als Argumenten operieren, was die Kontrollbeziehung umkehren und zahlreiche andere Probleme mit sich bringen würde.

Der einzige Weg, der hier gangbar erscheint, verläuft parallel zu der Vorgehensweise von GKPS, eine Argumentreihenfolge zu definieren, mit der die korrekte Interpretation von direktem und indirektem Objekt ermöglicht werden soll (S. 214), doch ist dies ein einzelsprachspezifischer Lösungsversuch und als genereller Ansatz in einem universell gedachten Prinzip untauglich.

Die hier aufgezählten Probleme bilden keine vollständige Liste; weitere, fundamentale Einwände gegen die Formulierung des CAP in GKPS werden in [Jacobson 1987], [Pollard 1984] und [Hukari/Levine 1986] vorgebracht. Der Tenor ist folgender:

[38]Diese Beschränkung läßt sich nur als einzelsprachlich motiviert interpretieren, denn es gibt Sprachen mit Verb-Objekt-Kongruenz.

Das CAP *ist* mit einer solchen Fülle an Aufgaben beladen, daß es eine angemessene, universell gültige Behandlung von Kongruenzphänomenen nicht leisten kann. Die vielen impliziten Abhängigkeiten von der Formulierung der ID-Regeln bis hin zur Arbeitsteilung mit FFP und HFC sind komplex und nicht vollständig durchschaubar. Die Aufgabenverteilung auf die drei verschiedenen Klauseln mutet willkürlich an, und die theoretisch-linguistische Motivation für die dritte Klausel (Gleichsetzung von Kontrollmerkmalen) ist fragwürdig.

Ich möchte diesen Abschnitt mit einem Zitat von Pollard zusammenfassen:

> It seems necessary to conclude that the GPSG-CAP is overly ambitious. The desire to subsume all agreement phenomena under a single unified principle is understandable, but considerably more work will have to be done to bring into focus a version of that principle which is both intuitively satisfying and linguistically motivated. [Pollard 1984, S. 134]

Infolge der zahlreichen grundsätzlichen Schwierigkeiten wird das CAP nicht in den Berliner GPSG-Formalismus übernommen, sondern es wird ein weniger anspruchsvolles Kongruenzprinzip verwendet (vgl. Abschnitt 6).

4.8 Die Head-Feature-Konvention

Das Instantiierungsprinzip HFC trägt ebenfalls zur Definition des zulässigen lokalen Baumes bei und schränkt die Menge der Projektionen weiter ein. GKPS geben eine Folge von Definitionen zunehmender Komplexität für HFC, die schließlich die Behandlung von Koordination mit einbeziehen. Dieser Abschnitt begnügt sich mit einer einfacheren Version und beschreibt einige recht trickreiche Konsequenzen für das Zusammenwirken der MIPs in der Absicht, die grundsätzliche Kritik am CAP zu verallgemeinern und auf die HFC auszudehnen.

4.8.1 Heads und HEAD-Merkmale

Die HFC betrifft eine Teilmenge der Merkmale, die Head-Merkmale, die durch die Menge HEAD definiert sind.

Der Begriff des Heads wurde in Abschnitt 2.2 folgendermaßen eingeführt. Betrachten wir die Töchter in einer ID-Regel mit denselben Spezifikationen für n und v wie die Mutter. Head ist die Tochter mit dem kleinsten bar-Wert, der kleiner oder gleich dem bar-Wert der Mutter ist. Diese Definition von Head basiert auf der Kospezifikation von Merkmalen zwischen der Mutter und einer Tochter. Sie induziert (durch die HFC, die für Heads definiert ist) eine weitere Kospezifikation zusätzlicher Merkmale an Mutter und Head und ist nur deshalb nicht zirkulär, weil die jeweiligen Merkmalmengen disjunkt sind. Diese Tatsache wiederum ist eine rein technische Bedingung für das Funktionieren des Formalismus und insofern aus linguistischer Sicht unbefriedigend.

Daher wählen GKPS eine andere Lösung. Sie verwenden bei der Formulierung von ID-Regeln die metasprachliche Variable H für eine stark unterspezifizierte Kategorie, die die Funktion eines Heads in der ID-Regel hat. Es bleibt damit der HFC (und anderen Komponenten des Formalismus) überlassen, die Spezifikationen von

Heads zu bestimmen. Statt (41) (vgl. Seite 43) schreiben GKPS (57).

(57) $\text{VP} \longrightarrow \text{H[17]}, \text{NP}, \text{VP[inf]}$

Der Head ist für **subcat** spezifiziert und muß daher in jeder Projektion [bar:0] sowie Spezifikationen für **n** und **v** besitzen (durch FCR (40) auf Seite 42). Welche Spezifikationen dies sind, wird durch die HFC bestimmt, denn **n** und **v** sind HEAD-Merkmale.

(58) $\text{HEAD} = \{\textbf{n}, \textbf{v}, \textbf{bar}, \textbf{subcat}, \textbf{subj}, \textbf{per}, \textbf{plu}, \textbf{aux}, \textbf{inv}, \textbf{vform}, \textbf{past}, \textbf{adv}, \textbf{prd}, \textbf{loc}, \textbf{slash}, \textbf{agr}\}$

Wenn wir die Menge HEAD aus GKPS betrachten, die für ein Fragment des Englischen definiert ist, ist unmittelbar klar, daß die HFC in der Version des Abschnitts 2.2 nicht mehr verwendet wird: Die Forderung, daß HEAD-Merkmale an der Mutter und einem Head gleich spezifiziert sein sollen, würde bereits infolge der Head-Definition aufgrund der Spezifikationen von **bar** oder **subcat** bei lexikalischen ID-Regeln unerfüllbar sein.

4.8.2 „Freie" Merkmalspezifikationen und eine Definition der HFC

In der Tat verwenden GKPS die HFC als ein Default-Prinzip, das nur diejenigen HEAD-Spezifikationen im lokalen Baum betrifft, deren Wert nicht durch ID-Regeln, FCRs, das FFP oder das CAP bestimmt ist.[39] Merkmalspezifikationen, die nicht durch ID-Regeln oder FCRs in einer Kategorie verboten sind, heißen *frei* an dieser Kategorie. Die Beschränkung auf Merkmalspezifikationen, die nicht durch FFP oder CAP instantiiert werden, erfolgt in der Definition der HFC selbst.

In dieser Definition wird als Hilfskonstrukt der Begriff der Menge freier Merkmalspezifikationen für eine Kategorie C benutzt. Sei Ψ_r die Menge aller Projektionen ϕ einer ID-Regel r. Die Menge $\psi(C, \Psi_r)$ aller *freien Merkmalspezifikationen an* C in r besteht aus den Spezifikationen, die in mindestens einem $\phi(C)$ aus Ψ_r auftreten.

Die HFC wird nun durch die beiden folgenden komplementären Regeln ausgedrückt:

Definition 19. Sei r eine ID-Regel mit der Mutter C_0 und einer Head-Tocher H. Sei ferner Φ_r die Menge aller Projektionen von r, die das FFP und das CAP erfüllen. Dann erfüllt ein durch $\phi(r) \in \Phi_r$ projizierter lokaler Baum die HFC gdw. gilt:

- Jede HEAD-Spezifikation von C_0 muß an H vorliegen, sofern sie in $\psi(\text{H}, \Phi_r)$ ist.
- Jede HEAD-Spezifikation von H muß an C_0 vorliegen, sofern sie in $\psi(C_0, \Phi_r)$ ist.

Betrachten wir als Beispiel einen durch die ID-Regel (57) projizierten lokalen Baum. Im Hinblick auf die an der Mutter vorhandenen Spezifikationen ist

$$\psi(\text{H}, \Psi_{(57)}) = \{\langle \textbf{bar} : 0\rangle, \langle \textbf{subcat} : 17\rangle, \langle \textbf{n} : +\rangle, \langle \textbf{n} : -\rangle, \langle \textbf{v} : +\rangle, \langle \textbf{v} : -\rangle\},$$

[39] Diese Sichtweise vertritt auch [Gunji 1987, S. 26, Fn. 11].

und im Hinblick auf die am Head vorhandenen Spezifikationen ist

$$\psi(\mathrm{VP}, \Psi_{(57)}) = \{(\mathbf{bar} : 2), (\mathbf{n} : -), (\mathbf{v} : +)\}.$$

Nach der ersten Klausel der HFC werden (n : —} und (v : -f} am Head verlangt. H ist nur für subcat und bar spezifiziert, doch keine dieser Spezifikationen ist frei an der Mutter.

HFC wird in der späten GPSG zusätzlich für die Beschreibung von Koordinationsphänomenen benutzt. Zu diesem Zweck wird die Möglichkeit vorgesehen, daß in einer ID-Regel mehrere Heads auftreten, jedes von ihnen ein Konjunkt. HFC „sammelt" quasi die allen Konjunkten gemeinsame Information an der Mutter. GKPS erfassen dies in ihrer endgültigen Definition der HFC, auf die hier nicht weiter eingegangen wird.[40] Darüber hinaus verlangt die Definition, daß nur solche Kategorien betrachtet werden, die für bar spezifiziert sind. Auf diese Weise wird die HFC direkt an die X-Bar-Syntax gebunden.

Soweit zur Definition der HFC. Es erscheint unklar, warum nur solche Projektionen betrachtet werden, die dem CAP und dem FFP genügen. Ich halte diese Forderung für unglücklich und irreführend. Durch sie wird eine Abhängigkeit der HFC von den anderen MIPs postuliert, die nicht nur eine ungewollte, zeitlich gebundene Anwendungsreihenfolge der MIPs suggeriert („HFC nach FFP und CAP"), sondern auch einfach redundant ist, denn für die Definition der Zulässigkeit (Abschnitt 4.5) wird ohnedies gefordert, daß ein lokaler Baum *allen* MIPs genügt. Bäume, die den beiden Bedingungen in der Definition der HFC genügen, nicht aber dem FFP oder dem CAP, sind mithin unzulässig, ohne daß dies aus der Definition der HFC folgen muß.

4.8.3 Die simultane Wirkung der MIPs am Beispiel der Topikalisierungsregel

Es gibt ein Problem, das die Interaktion der MIPs aufschlußreich beleuchtet. Betrachten wir die lokalen Bäume in Abbildung 9, die durch die Topikalisierungsregel (32) (vgl. Seite 41) projiziert wurden. In dieser Regel ist der Head S / [bar:2]. Es kann nun gezeigt werden, daß nur (iii) zulässig ist (was der Intuition, die mit der Topikalisierungseregel verbunden ist, zuwiderläuft). Das Problem besteht darin, daß slash ein HEAD-Merkmal ist.

(i) ist zulässig gemäß FFP und CAP. Die HFC verlangt nun, daß der slash-Wert des Heads an der Mutter (an welcher slash frei ist) spezifiziert wird (ii). Dieser lokale Baum (ii) verletzt jedoch das FFP, das verlangt, daß *slash* an einer Tochter instantiiert ist, wenn es an der Mutter instantiiert ist (der slash-Wert an der Head-Tochter ist ererbt!). Um das FFP zu erfüllen, muß der lokale Baum dieselbe Spezifikation auch an der anderen Tochter enthalten (iii). Nur diese Projektion erfüllt alle MIPs.

Die folgende Diskussion verfolgt nacheinander zwei Gedanken, die sich aus diesem Problem ergeben. Einmal scheint es sinnvoll, die mit der Topikalisierungsregel verbundene Intuition zu erfassen, indem die HFC darauf beschränkt wird, keine

[40]Der Effekt ist, daß die Mutter diejenigen HEAD-Spezifikationen enthält, die an *sämtlichen* Heads auftreten, sofern sie an der Mutter frei sind und daß jeder Head diejenigen HEAD-Spezifikationen enthält, die an der Mutter auftreten, sofern sie am jeweiligen Head frei sind.

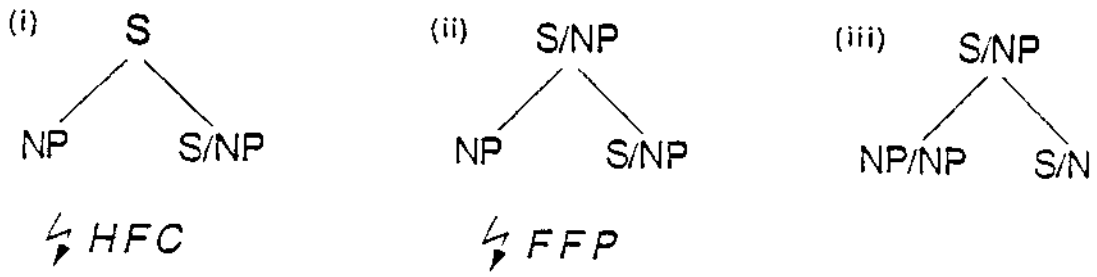

Abbildung 9: Projektionen aufgrund der Topikalisierungsregel

Merkmalspezifikationen zu fordern, die im Widerspruch zu FFP und CAP stehen d.h. der lokale Baum (i) in Abbildung 9 soll zulässig sein. Zum ändern soll ver folgt werden, was für und was gegen die Verwendung von slash als HEAD-Merkmal spricht.

4.8.4 Zur logischen Abhängigkeit zwischen CAP und HFC

In GKPS sind eine Reihe von Hinweisen enthalten, die auf logische Abhängigkeite der MIPs untereinander schließen lassen; der schwierigste Fall betrifft die im vorige Kapitel angedeutete Problematik zwischen CAP und HFC.

- Die HFC scheint die Wirkung des CAP (und des FFP) vorauszusetzen, we: sie keine Merkmalspezifikationen fordern darf, die durch diese Prinzipien aus
- geschlossen werden

- Das CAP setzt die Wirkung der HFC voraus, da es auf semantischen Ty pen beruht, die von HEAD-Merkmalen abhängen, deren Verteilung wiederun durch HFC geregelt ist.

Die simultane Anwendung von CAP und HFC auf lokale Bäume liefert somit in schlimmsten Fall die leere Menge als Menge zulässiger lokaler Bäume.

Ein möglicher Ausweg aus diesem Dilemma wurde von Shieber vorgeschlager [Shieber 1986], doch er beruht auf der Annahme, daß die HEAD-Merkmale in zwe disjunkte Teilmengen aufgeteilt werden können: in diejenigen HEAD-Merkmale, di(Voraussetzung sind für die 'Zuweisung semantischer Typen und insofern für die Anwendbarkeit des CAP und in diejenigen HEAD-Merkmale, die unabhängig von CAF instantiiert werden können. Allerdings ist es nicht sicher, ob eine solche Aufteilung möglich ist.

Natürlich kann man das Problem immer dadurch umgehen, daß man die Kategorien in den ID-Regeln stärker spezifiziert, als dies in GKPS vorgeschlagen ist; doch würde ein solches Vorgehen nicht im Geiste der GPSG sein, wo es darauf ankommt, universelle wie einzelsprachliche Generalisierungen auszudrücken.

4.8.5 Zur Verwendung von slash als HEAD-Merkmal

Zwei wesentliche Gründe für die Verwendung von slash als HEAD-Merkmal sind in der Behandlung von parasitären Lücken und von Koordination zu sehen. Ein Gegenargument basiert auf der Verwendung der folgenden ID-Regel.

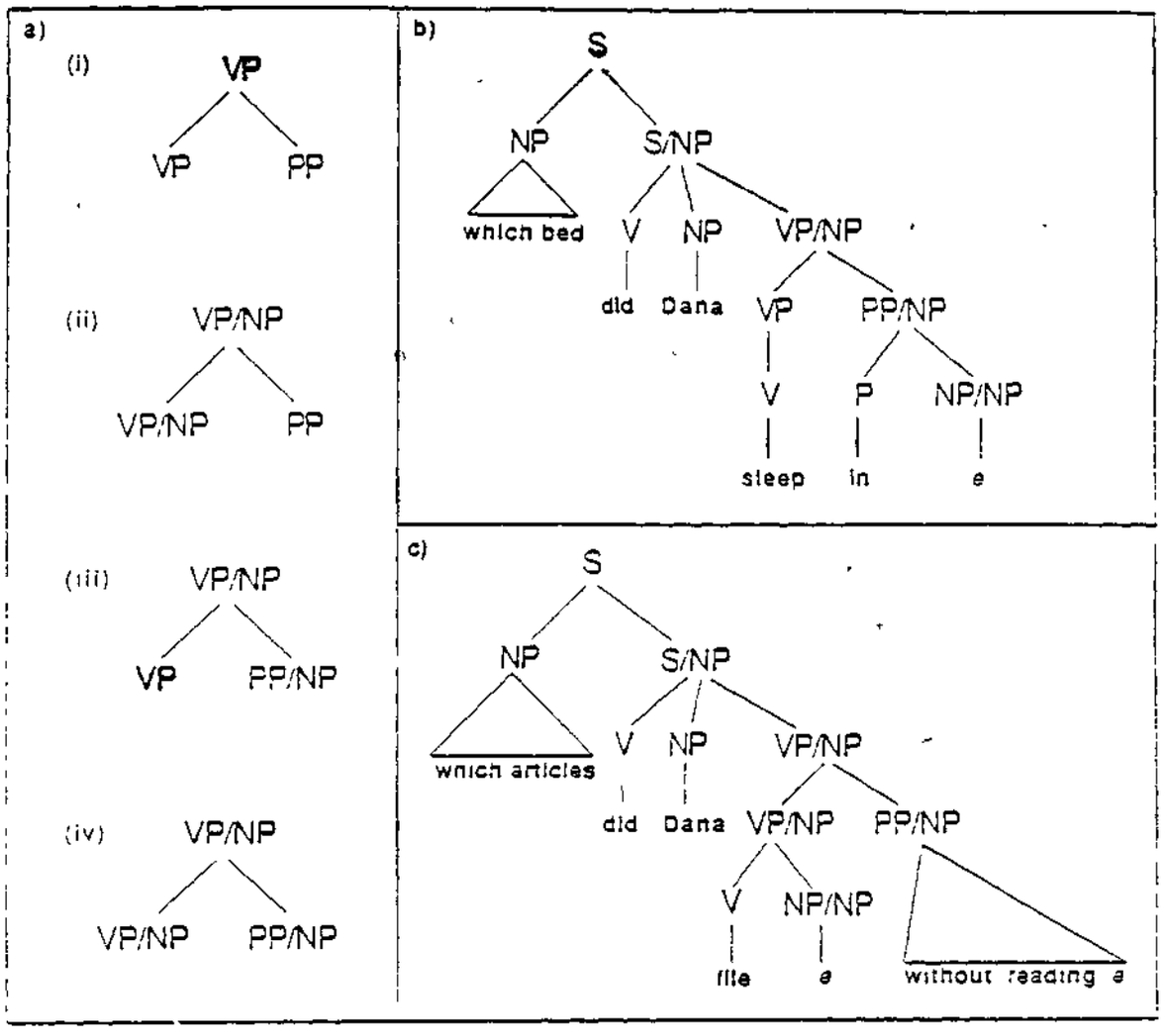

Abbildung 10: HFC und das Zusammenwirken der MIPs

(59) VP—>H,PP

(60) Which bed did Dana sleep in e?

Die ID-Regel (59) ist nicht in der Grammatik von GKPS enthalten, aber sie ist offensichtlich sinnvoll zur Beschreibung von freien Präpositionalergänzungen wie in (60), (die GKPS nicht behandeln). Abbildung lOa zeigt vier Projektionen aus $(59) (die CAP und FFP erfüllen). Offenbar ist (slash : NP) frei an allen drei Kategorien. Nach HFC ist dann der dritte Baum unzulässig, da der Head nicht die slash-Spezifikation der Mutter aufweist. Gerade er aber wird für die Analyse von (60) benötigt (siehe Abbildung lOb).

Die vorliegende Interaktion von HFC und FFP ist jedoch in anderen Fällen beabsichtigt. Bei parasitären Lücken kann sie ungrammatische Sätze wie (63), in denen die parasitäre Lücke nicht aus einem Head stammt, ausschließen, während (61) und (62) (vgl. Abbildung lOc) akzeptiert werden (vgl. GKPS, S. 162ff).

(61) Which articles did Dana file e without reading them?

(62) Which articles did Dana file *e* without reading e?

(63) * Which articles did Dana file dossiers without reading e?

Zwar folgen in diesem Beispiel die korrekten Fakten, doch ist die Behandlung parasitärer Lücken in GKPS fragwürdig, wie [Preuß 1989] nachweist.

Grundsätzliche Einwände gegen slash als HEAD-Merkmal führt Jacobson an. In der Definition des CAP werden die HEAD-Merkmale betrachtet, die nicht instantiierte FOOT-Merkmale sind. In der Grammatik von GKPS führt dies auf einen Ausschluß von slash-Spezifikationen hinaus. Das bedeute ganz einfach, so Jacobson, daß slash nicht die Eigenschaften anderer HEAD-Merkmale besitzt. Jacobson behauptet, daß dies eine Ursache für die Komplexität der letztlichen Definition der HFC sei:

> Moreover, there is absolutely *no* independent motivation for this formulation of the HFC; it is used solely to solve theory-internal problems which arise strictly because slash is defined äs a head feature. [Jacobson 1987, S. 414]

Die Komplexität der HFC im Vergleich zur mittleren GPSG beruht aber auch auf der Definition von Head und der Behandlung von Koordination mit HFC. Beides erfordert, daß HEAD eine Reihe weiterer Merkmale enthalten muß, die in der mittleren GPSG keine Head-Merkmale waren. Diese *problematischen* Merkmale sind es, die HFC zu einem Default-Prinzip machen, slash ist auch im Hinblick auf Koordination HEAD-Merkmal. In (64) wird somit an *beiden* Heads eine slash-Spezifikation erzwungen.

(64) Who did you see *e* and like e?

(65) What did she go and buy e?

In der Grammatik von GKPS sind neben slash nur die Merkmale bar, subj und subcat problematisch. Es wäre lohnend, Wege zu finden, ohne problematische Merkmale auszukommen. Voraussetzung dafür ist wahrscheinlich, daß auf die Behandlung von Koordination durch HFC verzichtet wird.

Von dieser These ausgehend, macht [Preuß 1989] den nächsten Schritt und definiert ein separates *Koordinationsprinzip,* das im wesentlichen die Fälle symmetrischer Koordination behandelt, in denen die Konjunkte Head-Kategorien sind (64). Das Koordinationsprinzip betrifft eine Menge von „Koordinationsmerkmalen", die an der Mutter und den Konjunkttöchtern im lokalen Baum gleich spezifiziert sein müssen. Phänomene asymmetrischer Koordination (65) werden durch die HFC in ihrer rigiden Fassung aus der mittleren GPSG erfaßt. Die Konjunkte werden durch unterschiedlich spezifizierte Heads beschrieben (in (65) V und VP).[41] Die HFC ist damit kein Default-Prinzip, sondern wird in ihrer Form aus der mittleren GPSG benutzt. Preuß gelangt mit ihrem Vorschlag zu einer Entflechtung der Aufgaben, die CAP und HFC zu übernehmen hatten und zu einer klareren Modularisierung der Metasprache.

Im Rahmen des Berliner GPSG-Systems wurde HFC als Default-Prinzip rekonstruiert. Allerdings werden die problematischen Merkmale in den ID-Regeln als spe-

[41] Man beachte, daß die problematischen Merkmale bar und subcat in der mittleren GPSG ebenso wie bei Preuß keine HEAD-Merkmale sind.

zifiziert angenommen, so daß das Konzept der freien Merkmalspezifikationen nicht berücksichtigt werden mußte (vgl. Abschnitt 6).

Teil II
Eine konstruktive Version von GPSG

5 Probleme der Algorithmisierung von GPSG

In diesem Abschnitt wird gezeigt, daß die im ersten Teil vorgestellte axiomatische Formulierung von GPSG in der Praxis nicht implementierbar ist. Die Überlegungen gehen von der Frage aus, welche Berechnungen effektiv durchzuführen wären, wenn die axiomatische Version von GPSG operationalisiert werden soll. Es wird ein Verfahren skizziert, das sich strikt an die deskriptive Art und Weise hält, wie die metasprachlichen Komponenten der GPSG definiert sind. Das Verfahren erweist sich als exponentiell-polynomial aufwendig und als ungeeignet für Sprachverarbeitung.

Diese Vorgehensweise unterscheidet sich von der in den komplexitätstheoretischen Studien in [Barton *et al.* 1987]. Dort wird statt der Komplexität eines *Verfahrens* die Komplexität des *Problems* betrachtet und somit über sämtliche mögliche Algorithmen, Grammatiken und Endketten verallgemeinert, die den Aufwand im Einzelfall beeinflussen können. Auf diese Weise entstehen „worst-case"-Analysen, die für konkrete Verfahren mit konkreten Grammatiken nicht unbedingt von Belang sein müssen. Sie zeigen aber auf, in welchen Komponenten des Formalismus man mit einer Explosion des Aufwandes *rechnen* muß und warum dies der Fall ist.

Zur Illustration mag ein Beispiel genügen. Für den Earley-Algorithmus ist der Analyseaufwand bekanntlich $O(|G|^2 \cdot n^3)$, wobei $\backslash G \backslash$ die Anzahl der PS-Regeln und *n* die Länge der Eingabekette ist. Die Analyse mit einer ID/LP-Grammatik kann das Verfahren von Earley grundsätzlich auf zwei verschiedene Weisen benutzen: Einmal ist eine Kompilation in eine kontextfreie Grammatik vorstellbar, und zum ändern eine direkte Interpretation der ID/LP-Grammatik. Berücksichtigt man nur das ID/LP-Format, so ist eine Kompilation nicht sinnvoll[42], denn die Transformation der komplexen Kategorien in monadische Kategoriensymbole von PS-Regeln ist, wie im folgenden gezeigt wird, außerordentlich aufwendig und erfordert sehr viele PS-Regeln. Da die Größe der Grammatik dann maßgeblich den Aufwand bestimmt, wird das Earley-Verfahren ineffizient. Ein Verfahren zur direkten Interpretation gibt [Shieber 1984]. Es ist so stark an den Earley-Algorithmus angelehnt, daß es den gleichen Zeitaufwand zu haben scheint. Da es mit weniger Regeln auskommt als eine stark äquivalente kontextfreie Grammatik, scheint Shiebers Verfahren günstiger zu sein.

Nun wird aber in [Barton *et al.* 1987] gezeigt, daß ID/LP-Parsing grundsätzlich nicht in polynomialem Zeitaufwand möglich ist. Shiebers Verfahren kann z.B. mit lexikalisch mehrdeutigen Regeln oder solchen, deren rechte Seite aus dem „leeren" Symbol *e* besteht, Probleme bekommen, da diese die Menge der Analysezustände u.U. drastisch aufblähen.

> Informally speaking, the reason why Shieber's parser sometimes suffers

[42] In Abschnitt 12 wird diskutiert, unter welchen Umständen eine Kompilation sinnvoll ist.

from combinatorial explosion *is* that there are exponentially more possible ways to progress through an unordered rule expansion than an ordered one. [Barton *et al.* 1987, S. 199]

Dennoch ist Shiebers Verfahren im „Normalfall" effizienter. Eine präzise Definition der Normalfälle folgt nicht aus dem GPSG-Formalismus, und somit sind die Resultate der komplexitätstheoretischen Analyse zunächst einmal wertvolle Hinweise für die Grammatikschreiberin, die darauf achten muß, daß die Verfahren nicht notwendig ineffizient werden.

Die Resultate von [Barton *et al.* 1987] für GPSG werden nun kurz zusammengefaßt, indem jeweils das untersuchte Problem als Frage formuliert, die Komplexitätsklasse der Lösungen und die wesentliche Ursache für das Resultat angegeben wird. Dabei werden beliebige GPS-Grammatiken angenommen; insbesondere solche, die zu extrem ungünstigen Resultaten führen.

- Ist eine ID-Regel im Abschluß einer Menge von ID-Regeln unter Metaregelanwendung enthalten?

  Resultat: NP-schwierig, d.h. nicht in polynomialem Zeitaufwand auf einer deterministischen Turingmaschine berechenbar.

  Grund: Als Anwendungsfolgen sind sämtliche Permutationen beliebiger Teilmengen der Metaregeln zu verwenden.

- Ist eine Extension einer Kategorie legal (unter Bezug auf die Definitionen der Kategorie und der FCRs aus GKPS ; vgl. die Abschnitte 4.1 und 4.2)?

  Resultat: PSPACE-schwierig; d.h. mit polynomialem 5pez'c/ieraufwand, nicht aber mit polynomialem Zeitaufwand auf einer deterministischen Turingmaschine berechenbar und vermutlich schwieriger als NP-schwierig.

  Grund: Im wesentlichen die hohe Anzahl der Kategorien und die FCRs mit disjunktivem „Consequens" (d.h. FCRs der Form $p \quad p_1 \ V \ldots V p_n$), für die im ungünstigen Fall *n* Fälle berechnet werden müssen.

- Ist *w € L(G)* für eine ID/LP-Grammatik *G!*

  Resultat: NP-vollständig; d.h. in polynomialem Zeitaufwand auf einer m'c/iideterministischen Turingmaschine berechenbar.[43]

- Ist *w € L(G)* für eine GPS-Grammatik G?

  Resultat: EXP-POLY-schwierig; d.h. in exponentiell-polynomialem Zeitaufwand auf einer deterministischen Turingmaschine berechenbar.

  Grund: Nullübergänge in ID-Regeln können zu komplexen Strukturen für die leere Kette führen, und die Verwendung problematischer HEAD-Merkmale bei HFC ist notwendig für die Beweisführung.

Kehren wir zurück zu der Frage, was ein Verfahren, das den Formalismus von GKPS implementiert, mindestens berechnen muß. Um alle zulässigen Bäume für einen gegebenenen Satz zu finden, müßten für jeden lokalen Baum die folgenden Schritte durchgeführt werden:

[43]Zu demselben Resultat gelangt [Ritchie 1987], jedoch auf anderen Wegen.

- *Generiere* jede Extension für jede Kategorie in einer ID-Regel *und teste,* ob sie legal ist. Filtere die anderen aus.

- *Generiere* alle Projektionen *und teste,* ob sie dem FFP, dem CAP, der HFC und den FSDs genügen und keiner LP-Aussage widersprechen. Filtere die anderen aus.

- *Generiere* alle Paare von Bäumen *und teste,* ob einer terminiert ist und ob seine Mutter mit einer Tochter des anderen identisch ist. Dann füge diese Kombination als Baum hinzu (sofern sie nicht mehr Blattknoten enthält als der Satz Wörter). Filtere alle anderen Paare aus.

Der erste Schritt erzeugt alle Kategorien, die nach der Definition der Merkmale und ihrer jeweiligen Wertebereiche möglich sind. Die Komplexität dieser vollständigen Aufzählung des Suchraumes führt zu einer kombinatorischen Explosion der Menge der Kategorien. Dies läßt sich numerisch durch folgende Überlegung beschreiben (vgl. [Ristad 1986]).

Die Anzahl möglicher Kategorien ist N_n (66), wobei N_{atom} die Anzahl der möglichen Kategorien mit atomwertigen Merkmalen und N_{cat} die Anzahl der möglichen kategorienwertigen Merkmalspezifikationen in Abhängigkeit von der Anzahl *n* der kategorienwertigen Merkmale ist. Im folgenden wird zunächst N_{atom} und dann N_{cat} bestimmt.

(66) $N_n = N_{atom} \bullet N_{cat}(n), \ (n > 0)$

(67) $N_0 = N_{atom}$

Natom hängt von der Anzahl *a* atomwertiger Merkmale und der Kardinalität von deren Wertebereiche ab. Sie ist $d > 3$, wenn man mindestens binäre Merkmale und die Möglichkeit des Fehlens ihrer Spezifikation in einer Kategorie betrachtet. Somit gilt (68).

(68) $N_{atom} > 3^a$

Bei *einem* kategorienwertigen Merkmal m (69) gibt es N_{atom} + l Möglichkeiten (m kann selbst Undefiniert sein und ist es notwendigerweise in seinem Wert). Bei zwei kategorienwertigen Merkmalen m, m' sind drei Fälle zu unterscheiden: m kann Undefiniert sein, m hat einen Wert, in dem m' Undefiniert ist, oder m hat einen Wert, in dem m' ebenfalls spezifiziert ist. Bei dem letzten Fall geht N_{atom} quadratisch in $N_{cat}(2)$ ein. Da dies auch umgekehrt für m' gilt, resultieren neun Kombinationen[44], in die N_{atom} bis zur vierten Potenz eingeht (70). Für den allgemeinen Fall gilt (71).

(69) $N_{cat}(1) = 1 + N_{atom}$

(70) $N_{cat}(2) = (1 + N_{atom}(1 + N_{atom})^1)^2$

(71) $N_{cat}(k) = (1 + N_{k-1})^k, \ (1 \leq k \leq n).$

N_n ist von der Größenordnung $O(N_{atom}^{e.n!}), (n > 1)$. Dies kann man sich anhand der

[44]Ristad gibt eine etwas andere Formel an ([Ristad 1986, S. 31, Fn. 4] bzw. [Barton *et al.* 1987, S. 222, Fn. 7]), die für diesen Fall sechzehn Kombinationen verlangt. Da die sieben zusätzlichen Glieder niederwertig sind, ist dieser Unterschied nicht von Bedeutung.

Gleichungen für N_{cat} überlegen. Für die Exponenten gelangt man zu der Folge c (vgl. (72)), woraus mit (73) die Behauptung folgt.

(72) $c = 1 + \frac{n!}{(n-1)!} + \frac{n!}{(n-2)!} + \cdots + \frac{n!}{(n-n)!} = n! \sum_{i=1}^{n} \frac{1}{i!}$

(73) $\lim_{n\to\infty} \sum_{i=1}^{n} \frac{1}{i!} = e \approx 2,718$

Für die Grammatik von GKPS ergeben sich bei vier kategorienwertigen und 25 atomwertigen Merkmalen $N_{atom} \geq 3^{25}$ und damit $N_4 > 3^{1625} > 10^{774}$.[45]

Ristad behauptet, daß diese astronomische Zahl inhärent für den GPSG-Formalismus sei. Dies setzt allerdings voraus, daß die Anzahl von Merkmalen und möglichen Werten als Teil des Formalismus begriffen werden und nicht als grammatikspezifisch. Obwohl diese Voraussetzung generell nicht gilt, ist Ristad zuzustimmen, wenn er meint, daß die Komplexität trotz Kontextfreiheit des zugrundeliegenden Formalismus unvermeidbar sei.

Der erste Schritt, Erzeugung der legalen Kategorien, wird abgeschlossen, indem diejenigen Kategorien, die einer FCR widersprechen, von der weiteren Betrachtung ausgeschlossen werden.

Betrachten wir den zweiten Schritt des Verfahrens. In ihm werden Projektionen erzeugt, was einen weiteren Aufwand von $O(|E|^k \cdot (k-1)!)$ bedeutet, wobei $|E|$ die durchschnittliche Kardinalität der Menge der legalen Extensionen einer Kategorie und k die durchschnittliche Anzahl der Kategorien pro ID-Regel ist. Durch die Permutation der Töchter kommt der Faktor $(k-1)!$ hinzu. Auf die Projektionen werden dann die MIPs angewendet und die unzulässigen lokalen Bäume ausgefiltert. Obwohl die MIPs, FSDs und LP-Aussagen nicht Filter im Sinne Chomskys sind, verhalten sie sich unter operationalen Bedingungen in analoger Weise, indem sie verhindern, daß vorher erzeugte Strukturen als zulässig eingestuft werden.

Der dritte Schritt operiert auf den Ergebnissen des zweiten. Er geht nicht unmittelbar aus GKPS hervor und implementiert zufällig eine Bottom-Up-Strategie, mit der aus der Menge der zulässigen lokalen Bäume die komplexeren Strukturen generiert werden.[46] Die Paare ermöglichen die Kombination zweier Bäume, indem der eine in den andern „eingehängt" wird. Diese Kombination ersetzt das zugrundeliegende Paar und nimmt in einem weiteren Zyklus teil, in dem erneut Paare gebildet werden und so fort, bis keine Paarbildung mehr möglich ist. Damit nur endlich viele Paare erzeugt werden, wird einfach die Anzahl der Blattknoten der Kombination beschränkt. Die Blattknoten der schließlich resultierenden Bäume werden mit der Folge der Wörter des Satzes verglichen, wodurch die erzeugten Strukturen des Satzes identifizierbar sind.

Das skizzierte Verfahren ist natürlich vollkommen unbrauchbar für reale Sprachverarbeitung. Man beachte, daß es zumindest in den ersten beiden Schritten notwendig auf der Generiere-und-Teste-Strategie beruht, die für große Suchräume inhärent ineffizient ist. Das Verfahren erfordert hierfür die explizite Aufzählung des Suchrau-

[45] Für $1625 = 25 \cdot x \cdot 4!$ besteht mit $x \approx 2,708$ bereits eine starke Annäherung an die Eulersche Zahl.

[46] Die lokalen Bäume können als normale PS-Regeln aufgefaßt werden, und damit ist für Schritt 3 ein polynomialer Aufwand erforderlich. Die Größe der Grammatik ist jedoch ebenfalls zu berücksichtigen (Earley gibt $O(|G|^2 \cdot n^3)$ an), und die ist gleich der Anzahl der zulässigen lokalen Bäume, mithin astronomisch.

mes, was praktisch nicht möglich ist.

Jedes andere Verfahren muß sich notwendigerweise stärker vom Original entfernen und die metasprachlichen Komponenten in gewissem Maße prozedural interpretieren. Ein realistischer Versuch, GPSG in einem natürlichsprachlichen System verfügbar zu machen, muß berücksichtigen, wie die Komponenten des Formalismus angewendet werden sollen und, vor allem, in welcher Reihenfolge. Diese unmittelbare Bindung an zeitliche Abläufe erfordert eine komplementäre Sichtweise von GPSG, die sich nicht unmittelbar aus GKPS ableiten läßt.[47] Diese Sichtweise geht von einem prozeduralen Standpunkt aus und fordert, daß die Gesamtheit der Komponenten von GPSG genau die wohlgeformten Kategorien und Bäume erzeugen sollen und keine weiteren. Insbesondere muß das Aufspannen des gesamten Suchraumes vermieden werden.

Die grundsätzliche Änderung der Sichtweise, die für die Sprachverarbeitung mit GPSG unumgänglich ist, läßt sich bezüglich der Erzeugung eines zulässigen lokalen Baumes wie folgt zusammenfassen:

- Anstatt für eine Vielzahl vollständig spezifizierter Kategorien und Projektionen zu entscheiden, ob sie legal bzw. zulässig sind, beginnt man mit einem einzigen Objekt, nämlich einem stark unterspezifizierten lokalen Baum, der durch eine ID-Regel projiziert ist und reichert ihn schrittweise durch Information an, die aus der Anwendung von FCRs und MIPs resultiert. Am Ende liegt ein vollständig determinierter lokaler Baum vor, der zulässig ist aufgrund seiner Genese.

Ich nenne diese Sichtweise *konstruktiv,* denn sie erlaubt die Konstruktion einer zulässigen syntaktischen Struktur. Im Unterschied hierzu ist der axiomatische Ansatz constraint-basiert; er sieht die Reduktion des Suchraumes auf die zulässigen Strukturen vor. In einer konstruktiven Version von GPSG agieren FCRs und MIPs als MerkmaUranspori-Prinzipien, die einen Informationszuwachs in den Kategorien im lokalen Baum bewirken. Im Unterschied hierzu filtern FCRs und MIPs in der axiomatischen Version nicht-legale bzw. unzulässige Objekte aus.

Der folgende Abschnitt präsentiert eine konstruktive Version der GPSG. Sie wurde im Rahmen der Berliner Projekte der EUROTRA-D-Begleitforschung mit dem Ziel entwickelt, wesentliche Teile des GPSG-Formalismus zu implementieren und damit die linguistischen Fundamente für Parsing und Generierung zu legen [Busemann/Hauenschild 1988, Hauenschild/Busemann 1988]. Der hier beschriebene Stand der konstruktiven Version ist nicht als endgültig zu betrachten (eine Weiterentwicklung, die eine allgemeine Behandlung der Koordination erlaubt, wird in [Preuß 1989] beschrieben).

[47]Shiebers Ansicht, daß aus GKPS eine implizite Abfolge herleitbar sei [Shieber 1986], erweist sich bei genauem Hinsehen als zu optimistisch; Shieber nennt selbst einige der oben diskutierten Hindernisse.

6 Der Berliner GPSG-Formalismus

6.1 Kategorien, Extension und Unifikation

Die Idee, eine Kategorie als ein Objekt zu betrachten, das schrittweise mehr Information aufnehmen kann, wird durch die Definition der Kategorie als n-stelliger Term implementiert. Dabei ist n die Anzahl der Merkmale. Die Werte von Merkmalen können atomar, kategorienwertig oder variabel sein. Das spezielle Atom „$\neg$“ drückt aus, daß ein Merkmal *undefiniert* bleibt (in GKPS wird dies durch die Assertion $\neg[f]$ ausgedrückt). Im Unterschied dazu bedeutet ein variabler Wert, daß das betreffende Merkmal *unspezifiziert* ist und noch ein Atom oder eine Kategorie als Wert bekommen kann.

Definition 20. Sei M eine Menge von Merkmalen, $W \cup \{\neg\}$ eine Menge von atomaren Werten und V eine Menge von Variablen. Seien M_{atom} und M_{cat} die Mengen der atom- bzw. kategorienwertigen Merkmale, wobei $M = M_{atom} \cup M_{cat}$ und $M_{atom} \cap M_{cat} = \emptyset$. Die totale Funktion $D : M_{atom} \rightarrow W$ weist einem Merkmal seinen Wertebereich zu. Eine totale Funktion C heißt *Kategorie* gdw. für alle $\mathsf{m}_i \in M$ $(1 \leq i \leq card(M))$

- $C(\mathsf{m}_i) \in D(\mathsf{m}_i)$ oder
- $C(\mathsf{m}_i) \in V$ oder
- $C(\mathsf{m}_i) = \neg$ oder
- $C(\mathsf{m}_i) = C'$, C' ist eine Kategorie mit $C'(\mathsf{m}_i) = \neg$, und für alle in C' eingebetteten Kategorien C'' gilt ebenfalls $C''(\mathsf{m}_i) = \neg$.[48]

Um eine gegebene Kategorie weiter zu spezifizieren, wird die Unifikation als einzige Basisoperation benutzt. Dies führt dazu, daß jegliche Assertionen über Kategorien, die in GKPS eine wichtige Rolle spielten, aufgegeben werden.

Neben der Notation, daß ein Merkmal undefiniert ist (die ihre Entsprechung durch den Merkmalwert „$\neg$“ erhält), bieten GKPS die Möglichkeit, die Forderung auszudrücken, daß ein Merkmal einen Wert haben muß (vgl. etwa (7) auf Seite 29, hier nochmals als (74) wiederholt). Dies könnte hier durch ein besonderes Variablensymbol realisiert werden, das jeden Wert außer „$\neg$“ annehmen darf. Es stellte sich jedoch nicht als notwendig heraus, dies in dem Formalismus aufzunehmen.

(74) $\langle \mathsf{subcat} \rangle \supset \neg\langle \mathsf{slash} \rangle$

(75) $\langle \mathsf{passive} : - \rangle \supset \neg\langle \mathsf{vform} : \mathsf{pas} \rangle$

Die Forderung, daß ein Merkmal nicht durch einen bestimmten Wert spezifiziert sein darf (vgl. (75)), läßt sich hingegen nicht in einer Kategorie ausdrücken. Obwohl es sehr nützlich wäre, *direkt* darzustellen, daß z.B. manche Verben keine Passivformen bilden, wird auf diese Möglichkeit zugunsten der Einfachheit der Kategoriendarstellung verzichtet.[49] Für die Beschränkungen hinsichtlich des Passivs gibt es andere

[48] Um dies formal zu beschreiben, genügt die Definition einer Funktion analog zu τ in Definition 3 auf S. 23.

[49] Dieser Verzicht ermöglicht die direkte Implementation von Kategorien als Prolog-Terme sowie die Verwendung der Prolog-Unifikation. Dies hat Vor- und Nachteile; siehe hierzu Abschnitt 7.

Ausdrucksmöglichkeiten (vgl. Abschnitt 10).

Wir betrachten jetzt konstruktive Formulierungen von Extension und Unifikation. Die Extension muß im Unterschied zur Version von GKPS die Möglichkeit undefinierter und variabler Spezifikationen umfassen. Variablen erhalten in einer Extension entweder denselben Namen, oder sie sind instantiiert.

Definition 21. Seien C_i, C_j Kategorien. C_j heißt *Extension* von C_i ($C_i \sqsubseteq C_j$), gdw. gilt:

- für alle $\mathsf{m} \in M_{atom}$:
  Falls $C_i(\mathsf{m}) \in D(\mathsf{m}) \cup \{\neg\}$, so $C_j(\mathsf{m}) = C_i(\mathsf{m})$
  Falls $C_i(\mathsf{m}) \in V$, so $C_j(\mathsf{m}) = C_i(\mathsf{m})$ oder $C_j(\mathsf{m}) = \neg$ oder $C_j(\mathsf{m}) \in D(\mathsf{m})$.
- für alle $\mathsf{m} \in M_{cat}$:
  Falls $C_i(\mathsf{m}) \in K$, so $C_i(\mathsf{m}) \sqsubseteq C_j(\mathsf{m})$.
  Falls $C_i(\mathsf{m}) \in V$, so $C_j(\mathsf{m}) = C_i(\mathsf{m})$ oder $C_j(\mathsf{m}) = \neg$ oder $C_j(\mathsf{m}) \in K$.
  Falls $C_i(\mathsf{m}) = \neg$, so $C_j(\mathsf{m}) = \neg$.

Ferner gilt für alle Spezifikationen mit identischem Wert $\mathsf{v} \in V$ auf beliebiger Ebene der Einbettung in C_i, daß ihre Gegenstücke in C_j einen identischen Wert w besitzen.

Die letzte Bedingung stellt sicher, daß die Kospezifikation von variablen Werten bei der Erzeugung einer Extension nicht verloren geht (beispielsweise gilt nicht: S[plu:X, rel:[plu:X]] $\sqsubseteq$ S[+plu, rel:[-plu]]).

Die Unifikation muß die Veränderung ausdrücken, die mit den Kategorien durch den Informationszuwachs vorgegangen ist. Ihre Definition erhält daher eine prozedurale Komponente, so daß Vor- und Nachzustand jeder Kategorie beschreibbar sind.

Definition 22. Sei K eine Multimenge von Kategorien. Die Elemente von K *unifizieren* (geschrieben $\sqcup(K)$) gdw. es eine Kategorie C gibt mit $C_i \sqsubseteq C$ für alle $C_i \in K$ und für alle weiteren Kategorien C' mit $C_i \sqsubseteq C'$ für alle $C_i \in K$ gilt: $C \sqsubseteq C'$.

Es gilt $\forall C_i \in K, \forall \mathsf{m} \in M : \; C_i(\mathsf{m}) \leftarrow C(\mathsf{m})$.

Durch den Linkspfeil wird eine Wertzuweisung ausgedrückt. Dies bewirkt, daß die Unifikation die Kategorien—im Unterschied zu GKPS—effektiv verändert. Wenn für ein Merkmal alle Werte in den C_i variabel sind, bedeutet das die *Kospezifikation* aller Variablen; d.h. wann immer sie einen konstanten Wert erhalten, erhalten sie denselben Wert. Die Unifikation bewirkt einen monoton wachsenden Grad der Spezifikation einer Kategorie. Eine atomare oder kategorienwertige Merkmalspezifikation kann höchstens noch spezifischer gemacht, niemals aber zurückgenommen werden. Diese Veränderung der Werte ist das Entscheidende am konstruktiven GPSG-Formalismus.

Ich verwende im folgenden die Notation „$\sqcup(K)|\mathsf{m}$“, wobei K eine Multimenge von Kategorien ist und $\mathsf{m} \in M$. Damit wird ausgedrückt, daß die Elemente von K *bezüglich* m operational unifizieren.

6.2 FCRs, ID-Regeln und LP-Aussagen

Der rein prädikative Charakter der FCRs wird ebenfalls zugunsten einer funktionalen Sichtweise aufgegeben. Eine FCR hat die Form $C_1 \Rightarrow C_2$ und besagt, daß jede Kategorie, die eine Extension von C_1 ist, mit C_2 unifizieren muß; erst dann ist sie legal. FCRs instantiieren also Merkmale und *erzeugen* somit legale Kategorien in allen Fällen, in denen die Unifikation möglich ist.

ID-Regeln und LP-Aussagen werden unverändert in die konstruktive Version übernommen. Komplexe LP-Aussagen, wie von [Uszkoreit 1986a] vorgeschlagen und in Abschnitt 4.3 diskutiert, werden nicht im Rahmen der Syntax formuliert. Da die GPSG-Komponente als modularer Bestandteil eines komplexeren Gesamtsystems konzipiert ist, kann lineare Präzedenz im Prinzip auf den verschiedenen Repräsentationsebenen dieses Systems beschrieben werden. Auf der syntaktischen Ebene kann somit freie Wortstellung bestehen mit Ausnahme der „harten" syntaktischen Beschränkungen, die die Wortstellung total festlegen (z.B. Det $\prec$ N1).

6.3 Die Head-Feature-Konvention

Im folgenden wird die Rolle der MIPs in einer konstruktiven Version von GPSG beschrieben. Ich beginne mit dem Problem, die freien HEAD-Spezifikationen für die Kategorien im lokalen Baum zu bestimmen, die für die Wirkung der HFC in GKPS wesentlich sind. Die Definition der freien Merkmalspezifikationen in Abschnitt 4.8 erfordert die Aufzählung einer großen Menge von Projektionen für jede ID-Regel, was grundsätzlich ein Problem von derselben Komplexität ist wie die Aufzählung der Kategorien, da HEAD-Merkmale kategorienwertig sein können. Um dies zu vermeiden, wird die Wirkung der HFC prozedural rekonstruiert, indem eine modifizierte Version von HFC einerseits *nach* den anderen MIPs angewendet wird und andererseits keine lokalen Bäume ablehnt.

Die Anwendung nach den anderen MIPs stellt sicher, daß alle Spezifikationen, die durch FCRs oder MIPs an HEAD-Merkmalen erzwungen werden, zum Zeitpunkt der Anwendung von HFC bereits vorliegen. Somit operiert die HFC auf lokalen Bäumen, die dem FFP und dem CAP genügen. Wenn ein HEAD-Merkmal an der Mutter und einem Head verschieden spezifiziert sind, entspricht das der Tatsache, daß es nicht frei an einer dieser Kategorien ist (unterschiedliche Spezifikation durch ID-Regeln oder durch FCRs) oder daß FFP oder CAP dies so verlangen. Daraus folgt, daß keine lokalen Bäume abgelehnt werden dürfen.

HFC identifiziert Heads mithilfe eines binären Merkmal **head**, das jeder GPS-Grammatik eigen ist. Jede Head-Tochter, die mit der Mutter bezüglich eines HEAD-Merkmals unifizieren kann, muß dies tun.

Definition 23. Sei t ein lokaler Baum, der alle FCRs erfüllt und allen anderen MIPs genügt. Sei C_0 die Mutter von t, und seien C_i $(1 \leq i \leq n)$ die Töchter in t mit $C_i(\mathsf{head}) = +$. Der Baum t genügt der HFC gdw. für alle $\mathsf{m} \in$ HEAD einer der folgenden Fälle zutrifft:

- $C_0(\mathsf{m}) \notin V$, und es gibt ein C_i mit $C_0(\mathsf{m}) \neq C_i(\mathsf{m})$

- Es gibt C_i, C_j mit $C_i(\mathsf{m}), C_j(\mathsf{m}) \notin V$, so daß gilt: $C_i(\mathsf{m}) \neq C_j(\mathsf{m})$. Dann gilt $C_0(\mathsf{m}) = \neg$
- $\sqcup(\{C_0, C_1, \ldots, C_n\})|\mathsf{m}$

Dies rekonstruiert die Intuition der HFC als Default-Prinzip, nicht jedoch den unerwünschten Effekt der Definition in GKPS, daß HFC durchaus Spezifikationen fordert, die nicht im Einklang mit den anderen MIPs stehen. Die vorliegende Definition der HFC dürfte daher ebenso allgemein und dabei wesentlich einfacher sein, als es in GKPS möglich war. Dennoch ist sicherlich auch hier das letzte Wort noch nicht gesprochen, denn es könnte sich herausstellen, daß diese Interpretation der HFC zu liberal ist. Ich komme darauf noch verschiedentlich zurück.

Bevor ich mit der Beschreibung der anderen MIPs fortfahre, sind einige Bemerkungen zur Interaktion der HFC mit FCRs am Platze. Nach Anwendung der HFC können FCRs anwendbar werden, die vorher nicht anwendbar waren. Somit müssen FCRs nach Anwendung der HFC auf Anwendbarkeit geprüft werden. Sollte eine FCR aufgrund einer durch HFC instantiierten HEAD-Spezifikation anwendbar werden und eine Kategorie als illegal zurückweisen, liegt eine Situation vor, die in der axiomatischen Version von GPSG nicht auftreten kann. Ein Beispiel: HFC hat (vform : pas) von der Mutter an ein (verbales) Head transportiert, das nicht passivfähig ist (ausgedrückt durch (passive : −)). Die FCR (76) wird erstmalig anwendbar auf das Head. Da sie verlangt, daß eine Konstituente mit einer Passivspezifikation auch passivierbar sein muß, weist sie das Head als illegal zurück.

(76) [pas] ⇒ [+passive]

In GKPS würden FCRs diese Kategorie von Anfang an nicht zugelassen haben, und HFC hätte es infolgedessen mit einer anderen Spezifikation von vform zu tun gehabt (die natürlich nicht frei an der Mutter ist). Situationen dieser Art werden im konstruktiven Formalismus nicht gesondert berücksichtigt.

Es deutet sich ein fundamentales Problem mit dem konstruktiven Formalismus an: wie soll bei unvollständiger Spezifikation von Kategorien zu einem gegebenen Zeitpunkt endgültig festgestellt werden, ob eine Kategorie legal ist? Ich werde das Problem in Abschnitt 6.7 diskutieren und eine Lösung vorstellen.

6.4 Das Agreement-Prinzip

Die Tatsache, daß das CAP vor der HFC angewendet werden muß, stellt uns vor das Problem, daß es lokale Bäume geben kann, die nicht ausreichend spezifiziert sind, um die semantischen Typen zu identifizieren, auf denen das CAP beruht. Wie bereits erwähnt, basieren die Typen teilweise auf HEAD-Spezifikationen. Darüber hinaus besteht neben der Vielzahl linguistischer Kritikpunkte, die in Abschnitt 4.7 diskutiert wurden, die formale Schwierigkeit, daß die Kontrollbeziehung nicht hinreichend allgemein für Funktoren mit mehr als einem Argument definiert ist.

Das CAP wurde daher ersetzt durch einen rein syntaktischen Mechanismus, das *Agreement Principle* (AP), das folgendermaßen definiert ist [Weisweber 1987]: Jede Tochter in einem lokalen Baum, die an Kongruenzbeziehungen teilnimmt, muß mit der Mutter hinsichtlich einer Teilmenge AGR der Merkmale unifizieren. Wenn ein

AGR-Merkmal undefiniert ist, wird es vom AP ignoriert. Jeder lokale Baum, der das AP nicht erfüllt, ist unzulässig. AGR enthält gewöhnlich Merkmale für Kasus, Genus, Numerus und Person. Eine Kategorie wird dafür gekennzeichnet, daß sie an Kongruenzbeziehungen teilnimmt, indem FCRs ein binäres Merkmal agr instantiieren (z.B. sichern die FCRs (77)–(78) die Kongruenz von Subjekt und finitem Verb im Deutschen).

(77) [nom] ⇒ [+agr]

(78) [fin] ⇒ [+agr]

Definition 24. Sei t ein lokaler Baum mit der Mutter C_0 und derjenigen Teilmenge von Töchtern C_i $(1 \leq i \leq n)$, für die gilt: $C_i(\mathbf{agr}) = +$. Dann genügt t dem AP gdw. für alle $\mathsf{m} \in$ AGR einer der folgenden Fälle zutrifft:

- $C_0(\mathsf{m}) = \neg$
- $\sqcup(\{C_0\} \cup \{C_i | C_i(\mathsf{m}) \neq \neg\})|\mathsf{m}$

Diese Art, Kongruenzbeziehungen darzustellen, hat viel gemein mit der des CAP der mittleren GPSG. Sie kommt ohne ein Konzept von Kontrolle aus. Es sind keine semantischen Typen involviert, und welche Konstituenten miteinander kongruieren, muß nicht separat festgestellt werden, sondern ist einfach eine Folge aus dem Zusammenspiel zwischen dem AP und der HFC sowie den einzelsprachlichen Definitionen der FCRs und den Merkmalmengen AGR und HEAD. Die genaue Aufgabenverteilung zwischen HFC und AP hängt in der konstruktiven Version von GPSG u.a. davon ab, in welcher Reihenfolge die beiden Prinzipien operieren (wenn AGR ∩ HEAD $\neq \emptyset$) und wann die FCRs angewendet werden.[50] Die Anwendungsreihenfolge der MIPs und FCRs wird weiter unten diskutiert.

Allerdings kann das AP in dieser Form nicht der Tatsache gerecht werden, daß eine Kategorie an bestimmten Kongruenzrelationen teilnimmt und an anderen nicht (z.B. kommt es in Equi-Konstruktionen vor, daß ein direktes Objekt mit einem Reflexivpronomen kongruiert, aber nicht mit dem finiten Verb). In Kategorien ist dies nicht mit einem binären Merkmal (agr) ausdrückbar. Eine verfeinerte Version des AP wurde in [Busemann/Hauenschild 1988] vorgeschlagen; sie beruht auf verschiedenen Werten von agr, die die Kongruenzrelationen voneinander unterscheiden (z.B. agr1 und agr2 statt +). In besagten Equi-Konstruktionen ist dann ein direktes Objekt (genau wie das Reflexivpronomen) mit ⟨agr : agr2⟩ spezifiziert, während das Subjekt und das finite Verb beide ⟨agr : agr1⟩ enthalten. Das solchermaßen revidierte AP verlangt, daß Kategorien mit derselben agr-Spezifikation hinsichtlich der AGR-Merkmale unifizieren müssen, wie oben beschrieben.

Dieser Ansatz erlaubt es Kategorien, Merkmalspezifikationen zu transportieren, die aus verschiedenen Kongruenzrelationen stammen. Dies funktioniert nur, wenn die Kategorie ererbte kategorienwertige Merkmale enthält, die selbst wiederum für

[50] Außerdem spielt die Strategie eine Rolle, aufgrund derer lokale Bäume durch Kategorienunifikation *vor* Anwendung der MIPS miteinander kombiniert werden. Es kann also in einem lokalen Baum Information vorliegen, die aufgrund der Anwendung der MIPs in einem anderen lokalen Baum instantiiert wurde. Näheres hierzu folgt weiter unten und in Abschnitt 9.

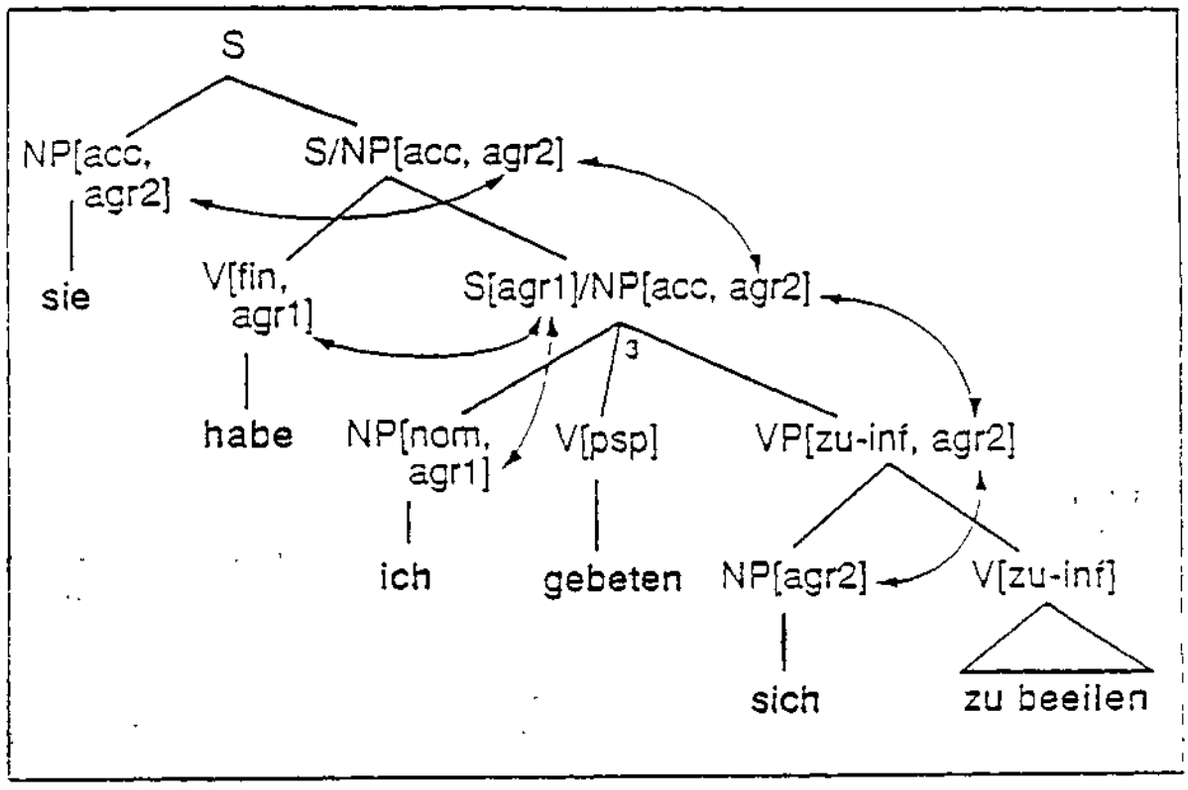

Abbildung 11: Kongruenzbeziehungen mit dem revidierten AP

agr spezifiziert werden können. Diese Spezifikationen werden ebenfalls vom revidierten AP betrachtet, um in einem lokalen Baum festzustellen, welche Kategorien an Kongruenzrelationen teilnehmen. Die Hypothese, die der Revision des AP zugrundeliegt, besagt, daß nicht mehr Kongruenzrelationen zu berücksichtigen sind, als Kategorien vorliegen (einschließlich der eingebetteten). Betrachten wir in Abbildung 11 die Mutter des lokalen Baums (3). Sie enthält einen ererbten slash-Wert[51]. Das revidierte AP benutzt die agr-Spezifikation im slash-Wert, um Kongruenz zwischen dem direkten Objekt (NP[acc]) und dem Reflexivpronomen herzustellen. Die agr-Spezifikation von S wird benutzt, um Subjekt-Verb-Kongruenz zu etablieren.

Das Merkmal agr kann nach wie vor durch FCRs instantiiert werden, obwohl auf einige charakteristische Ausnahmen zu achten ist, bei denen der Wert besser in den ID-Regeln spezifiziert wird. Zu ihnen gehört die VP, die nicht immer (agr : agr2) wie in Abbildung 11 enthalten darf, denn im Falle von Subjekt-Equi-Verben muß sie mit dem Subjekt kongruieren.

Einen noch allgemeineren Vorschlag für ein Kongruenzprinzip im Rahmen einer konstruktiven Version von GPSG enthält [Preuß 1989]. Der Leitgedanke ist, einen Mechanismus anzugeben, der das gesamte Aufgabenspektrum des CAP abdeckt, ohne auf semantische Typen zurückgreifen zu müssen. Das Kongruenzprinzip beruht darauf, daß in den ID-Regeln sogenannte *Spezifikationsmerkmale* an verschiedenen Kategorien (unterschiedlicher Einbettungstiefe) denselben Wert besitzen. Im lokalen Baum verlangt das Kongruenzprinzip an diesen Kategorien die Kospezifikation ei-

[51] Die verwendeten Grammatiken führen Lücken in ID-Regeln an der Mutter ein. Dies wird u.a. auch in [Jacobson 1987] vorgeschlagen; siehe Abschnitt 4.6.

ner Menge von *Kongruenzmerkmalen*, die dem betreffenden Spezifikationsmerkmal zugeordnet sind.

(79) S—>[+top],S/[+top]

Betrachten wir den für GAP so problematischen Fall der Topikalisierungsregel (32), die im Berliner System die Form von (79) annimmt. Dem Spezifikationsmerkmal top sind alle Merkmale der Grammatik zugeordnet. Infolgedessen muß in jeder Projektion von (79), die dem Kongruenzprinzip genügt, die Tochter [+top] mit dem slash-Wert der anderen Tochter übereinstimmen, und zwar unabhängig davon, welchen Typs die topikalisierte Kategorie ist.[52]

6.5 Das Foot-Feature-Prinzip

Ich schließe die Beschreibung der MIPs mit dem FFP ab, dessen Wirkung im Großen und Ganzen von GKPS übernommen wurde. Eine spezielle Behandlung ist notwendig für den Merkmalwert „->". Alle Töchter unifizieren mit der Mutter hinsichtlich der Teilmenge FOOT der Merkmale, vorausgesetzt, daß die Werte nicht ererbt sind. Töchter, die Undefiniert bezüglich eines FOOT-Merkmals sind, werden vom FFP ignoriert, es sei denn, das FOOT-Merkmal ist an allen Töchtern oder der Mutter Undefiniert. In diesem Falle fordert das FFP, daß alle Kategorien Undefiniert bezüglich dieses FOOT-Merkmals sind. Ein lokaler Baum, der das FFP verletzt, ist unzulässig.

Definition 25. Sei *t* ein lokaler Baum mit der Mutter C_0 und einer Menge von Töchtern C_i ($1 \leq i \leq n$). Dann genügt *t* dem FFP gdw. für alle instantiierten Spezifikationen von FOOT-Merkmalen m einer der folgenden Fälle zutrifft:

- $(\forall C_i : C_i(\mathsf{m}) = \neg) \equiv (C_0(\mathsf{m}) = \neg)$
- $\sqcup(\{C_0\} \cup \{C_i | C_i(\mathsf{m}) \neq \neg\})|\mathsf{m}$

Das theoretische Problem der Überlastung des Merkmals slash (vgl. Abbildung 5 auf S. 44) besteht auch hier. In den verwendeten Grammatikfragmenten spielt es jedoch keine Rolle.

6.6 Eine Anwendungsreihenfolge

Nachdem die Komponenten der konstruktiven GPSG soweit beschrieben wurden, soll die Frage der Anwendungsreihenfolge erneut behandelt werden. Aus konstruktiver Sicht soll die Anwendung einer Komponente idealerweise nicht früher erfolgen, als bis alle Informationen, auf denen sie aufbaut, definitiv in den Kategorien spezifiziert sind. Wenn zum Beispiel nicht sichergestellt ist, daß Kategorien nicht noch Extensionen von jenen in LP-Aussagen werden können, sollte nicht versucht werden, LP-Konsistenz zu überprüfen.

[52] Dieser Vorschlag wird im Rahmen des Projekts KIT-FAST implementiert und anhand von GPSG-Fragmenten für Deutsch und Englisch überprüft. In der vorliegenden Arbeit wird noch die Implementation der ersten Version des AP beibehalten.

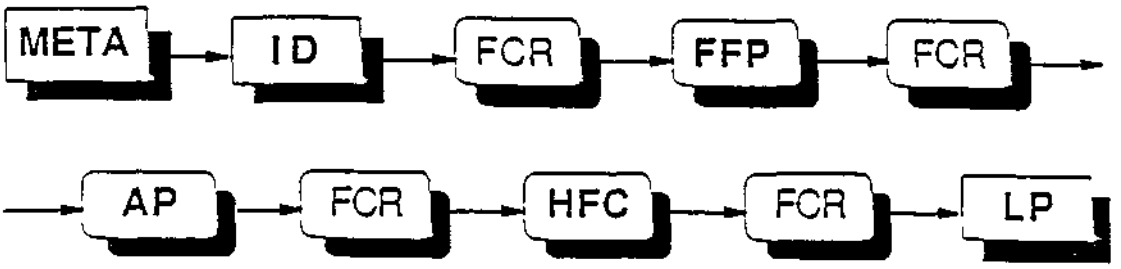

Abbildung 12: Anwendungsreihenfolge in einer konstruktiven Version von GPSG

Das FFP hängt ausschließlich von Spezifikationen in ID-Regeln ab und kann daher zuerst angewendet werden. Das AP setzt die vorherige Anwendung von FCRs voraus. Wie oben dargelegt wurde, operiert das AP vor der HFC. LP-Konsistenz kann nur bei vollständig spezifizierten Kategorien ein sicheres Ergebnis bringen. Daher erfolgt der entsprechende Test ganz am Schluß. Da nach jedem Schritt neue Information in Kategorien auftreten kann, die FCRs anwendbar macht, welche vorher nicht anwendbar waren, werden FCRs nach dem FFP, dem AP und der HFC angewendet. Aus Effizienzgründen ist es sinnvoll, sie auch auf Kategorien in lokalen Bäumen anzuwenden, die nur die Information aufgrund der ID-Regeln enthalten, um fehlerhafte Regelauswahl beim Parsing oder bei der Generierung zum frühestmöglichen Zeitpunkt zu entdecken.

Abbildung 12 stellt eine Anwendungsreihenfolge dar, die durch Aspekte des lokalen Informationszuwachses motiviert wurde. Sie ist ähnlich zu der Reihenfolge, die Shieber auf der Grundlage impliziter Annahmen in GKPS gefunden hat [Shieber 1986].

Eine Komponente in Shiebers Abfolge fehlt jedoch in Abbildung 12, nämlich FSDs. Sie wurden nicht in den Formalismus aufgenommen. Stattdessen findet eine rigidere Komponente Platz, die zur Definition der ID-Regeln benutzt wird: Mit *Ahasnamen* für Kategorien [Evans/Gazdar 1984] werden Kurzformen wie NP[nom] interpretiert als Kategorien mit einer Reihe von Spezifikationen, die der Kurzform nicht ohne weiteres anzusehen sind. Beispielsweise sind alle NPs für vform Undefiniert. Aliasnamen ermöglichen eine ausreichende Beschränkung der freien Merkmalinstantiierung, um auf FSDs ganz verzichten zu können. Abschnitt 7 geht näher auf Aliasnamen ein. Dort wird auch die Rolle von Metaregeln im Berliner GPSG-System beschrieben.

6.7 Zulässige Bäume

Zum Abschluß der Darstellung der konstruktiven Version wird die Definition der Zulässigkeit gegeben.

Definition 26. Ein Baum ist *zulässig* gdw. er terminiert ist und jeder lokale Teilbaum lokal zulässig ist. Ein lokaler Baum ist *lokal zulässig* gdw. er entweder aus dem Lexikon stammt oder durch eine ID-Regel projiziert ist, die FCRs erfüllt, dem FFP, dem AP und der HFC genügt und keiner LP-Aussage widerspricht.

Diese Definition ist, obwohl analog zu der in [GKPS, S. 104], nicht unproblematisch, da die Art und Weise, in der lokale Bäume zu komplexeren Bäumen kombiniert wer-

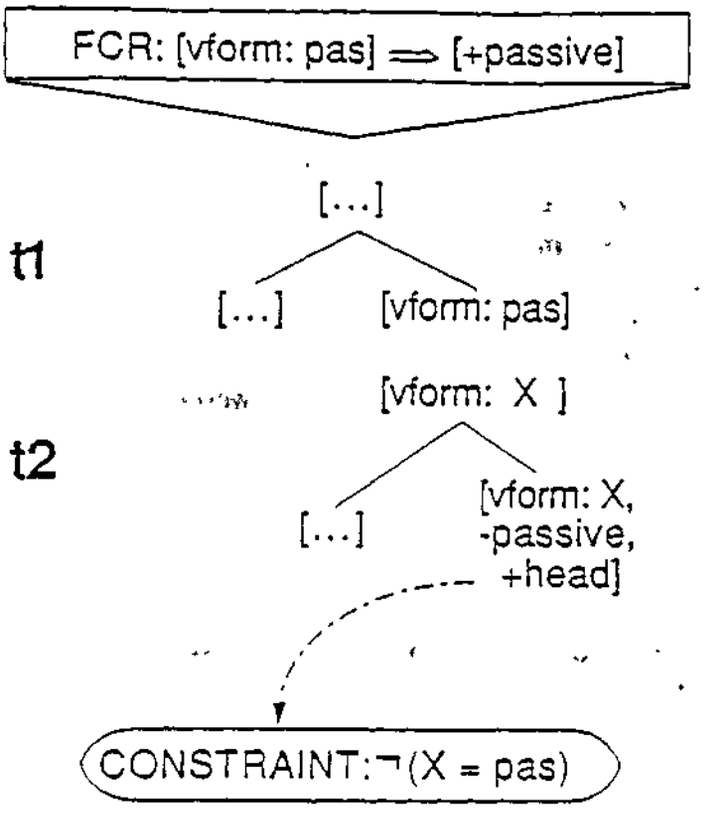

Abbildung 13: Kontrolle der Baumgenerierung durch Constraints

den können, sich von der in GKPS unterscheidet. Wie in Abschnitt 5 erwähnt wurde, beruht die Kombination in GKPS auf der *Identität* der Mutter des einen mit einer Tochter des anderen Baumes. Aus konstruktiver Sicht müssen die beiden Kategonen *unifizieren.* Durch diese *Instantiierung durch Konstruktion* werden Spezifikationen aus einem lokalen Baum in einem anderen lokalen Baum zugänglich. Dies kann die Wirkungsweise der MIPs beeinflussen. Zu welchen Zeitpunkten Konstruktion erfolgen und MIPs angewendet werden sollen, wird in Abschnitt 9 diskutiert.

Betrachten wir jetzt den folgenden Fall (Abbildung 13): Der Baum *t%* ist zulässig; er hat ein unspezifiziertes HEAD-Merkmal h, das mit der Mutter durch HFC *kospezifiziert* ist, d.h. denselben variablen Wert hat. In solchen Situationen kann die Instantiierung durch Konstruktion die Zulässigkeit zunichte machen. Man sehe: Die Mutter von t_2 unifiziert mit einer Tochter von t_1. Diese Tochter enthält eine Spezifikation für h (oder wird schließlich eine erhalten durch MIPs oder FCRs), wodurch kospezifizierte Werte in verschiedenen Kategorien von t_2 instantiiert werden. Doch nun ist t_2 auf einmal kein zulässiger Baum mehr, weil der Zuwachs an lokaler Information eine illegale Kategorie erzeugt hat. Grundsätzlich tritt dasselbe Problem auch bei LP-Aussagen auf, so daß eine gegebene Abfolge der Töchter von einer LP-Aussage abgelehnt würde, die zum Zeitpunkt der Überprüfung von LP-Konsistenz nicht anwendbar war.

Da eine abermalige Überprüfung des gesamten Baumes nach vollständiger Spezifikation aller Kategorien keine Lösung darstellt (was soll geschehen, wenn eine FCR interveniert?), wurde eine Strategie der Constraint-Propagierung für FCRs und LP-

Aussagen entwickelt [Weisweber 1988]. Die Grundidee[53] wird am Beispiel der FCRs erläutert.

Nehmen wir an, ein Merkmal m sei unspezifiziert an einer Kategorie *C (C(m)* € V). Eine FCR des Typs (80)[54] wäre auf *C* erst anwendbar, wenn *C(m)* = w gilt. Wenn C(m') = w', ist es unerheblich, welchen Wert m erhält: nach einer Instantiierung von m ist sie entweder nicht anwendbar oder erfüllt. Wenn jedoch *C(m') ungl.* w', so wird ein aussagenlogisches Constraint erzeugt. Falls w' in *C* spezifiziert ist *(C(m') kein el. V),* hat es die Form von (81). Falls w' € V (was u.a. bedeutet, daß die FCR bei einer Anwendung m' instantiiert), so wird (82) erzeugt. Dies erfolgt für alle FCRs; die erzeugten Constraints werden konjunktiv verknüpft und mit *C* assoziiert.

(80) [m:w] => [m':w']

(81) C(m)ungl.w

(82) (C(m) *ungl.* w) V (C(m') = w')

Jede Tochter im lokalen Baum erhält auf diese Weise eine aussagenlogische Formel in konjunktiver Normalform.

Abbildung 13 zeigt ein einfaches Beispiel. Die Variable X wurde in i_2 durch HFC an der Mutter und dem Head kospezifiziert. Wenn der Baum t_2 mit t_1 kombiniert wird, erfolgt eine Auswertung der Constraints an jeder Tochter. Durch die Kombination können Merkmale instantiiert worden sein (wie im Beispiel durch X <— pas). Im Beispiel wird dadurch das Head illegal, und t_2 ist nicht (mehr) zulässig. Die Kombination von t_1 und t_2 muß rückgängig gemacht werden.

Im allgemeinen können als Ergebnis dieser Auswertung die folgenden Fälle eintreten:

- Ein Teilausdruck in einem Constraint ist wahr und wird entfernt. Bleiben ausschließlich Gleichungen übrig, wird die entsprechende Merkmalinstantiierung in *C* veranlaßt. Die zugrundeliegende FCR ist nun für *C* erfüllt.
- Das Constraint ist falsch. *C* erfüllt nicht alle FCRs und ist nicht legal.
- Das Constraint ist nicht vollständig auswertbar, weil für ein m C(m) nach der Kombination von t_2 und t_1 variabel bleibt; die betreffenden Teilausdrücke des Constraints werden zur Mutter propagiert und im lokalen Baum t_1 weiterverarbeitet.

Die Propagierung zum (topologisch gesehen) höheren lokalen Baum setzt eine Bottom-Up-Auswertung voraus. Wie gezeigt werden wird, ist dies im Hinblick auf die Verarbeitungsstrategien angemessen.

Die Anwendung der FCRs in Form von kategorienspezifischen Constraints bringt, quasi nebenbei, zwei weitere Vorteile mit sich. Zum einen ist zu jedem Zeitpunkt

[53]Eine elegante Implementation wäre mithilfe von *freeze* aus Prolog II [Colmerauer 1982] möglich. Ein weiterer Weg besteht in der Verwendung eines anderen Unifikators, etwa im Sinne der *bedingten Unifikation,* wie von [Hasida 1986] vorgeschlagen.

[54]Ohne Beschränkung der Allgemeinheit nehme ich nur jeweils eine Merkmalspezifikation auf beiden Seiten an.

die für eine Kategorie relevante Teilmenge der FCRs berücksichtigt, so daß jede erfüllte FCR im nächsten Anwendungszyklus nicht mehr beachtet werden muß (infolge der Monotonie-Eigenschaft der Unifikation ist sichergestellt, daß sie auch weiterhin erfüllt ist). Dies bedeutet einen beachtlichen Effizienzgewinn. Zum ändern muß nicht mehr auf die Anwendungsreihenfolge der FCRs untereinander geachtet werden (wenn eine FCR aufgrund von Spezifikationen anwendbar ist, die nur von einer anderen FCR instantiiert werden, darf sie erst nach dieser angewendet werden, da sie sonst niemals zum Zuge käme).

Bei LP-Aussagen ist eine ähnliche Lösung angemessen. Falls eine LP-Aussage einen lokalen Baum ablehnen würde, wenn ein Merkmal m mit dem Wert w instantiiert wird, so wird ein Constraint erzeugt, welches ausdrückt, daß m nicht den Wert w in dieser Kategorie erhalten darf. Diese Constraints werden propagiert und ausgewertet, solange die Werte von m variabel und kospezifiziert sind.

6.8 Die wesentlichen Unterschiede zwischen dem axiomatischen und dem konstruktiven GPSG-Formalismus

In diesem Abschnitt werden die wichtigsten Unterschiede zwischen der konstruktiven und der axiomatischen GPSG anhand einiger Schlüsselbegriffe zusammengefaßt.

- Kategorien können zusätzliche Information aufnehmen. In der axiomatischen Version sind sie unveränderlich.
- Die Unifikation ist die grundlegende Operation für Informationszuwachs in Kategorien. In der axiomatischen Version ist sie eine Wohlgeformtheitsbedingung.
- Die FCRs und MIPs verursachen lokalen Informationszuwachs. In der axiomatischen Version wählen sie unter Kategorien bzw. Projektionen aus.
- Syntaktische Strukturen werden aufgrund der Unifikation von Kategorien erzeugt. In der axiomatischen GPSG existieren sie infolge der Identität von Kategorien.

Es wurde gezeigt, daß die konstruktive Version von GPSG die kombinatorische Explosion vermeidet, die eine originalgetreue Interpretation der Metasprache von GKPS bewirkt hätte. Die veränderte Sichtweise führt zu einer beachtlichen Vereinfachung der HFC, da es nicht mehr notwendig ist, die Mengen der freien HEAD-Spezifikationen zu bestimmen; daher brauchen nicht alle Projektionen einer ID-Regel betrachtet zu werden.

Das Dilemma hinsichtlich der logischen Abhängigkeit von CAP und HFC, das bei jeder Implementation in einen schwer zu behebenden Reihefolgenkonflikt mündet, wurde ebenfalls beseitigt. Der Preis dafür ist allerdings hoch: das AP ist nicht mehr semantisch basiert und kann daher nicht mehr Keenan's Prinzip gerecht werden, daß Kongruenzbeziehungen auf bestimmte Weise von Funktor-Argument-Strukturen abhängen.

7 Die Architektur des Berliner GPSG-Systems

Die konstruktive Version von GPSG bildet die Grundlage zur syntaktischen Sprachverarbeitung im Berliner GPSG-System. Sie ist gleichermaßen geeignet für Parsing und Generierung, denn sprachliches Wissen kann strikt vom Verarbeitungswissen des Parsers bzw. des Generators getrennt werden. Dies ermöglicht die Analyse und Generierung mit ein- und derselben, bidirektional zu verwendenden Grammatik.

Ferner werden einzelsprachspezifisches und generelles syntaktisches Wissen[55] strikt voneinander getrennt. Dies ermöglicht den Austausch von Grammatiken, ohne den Parser bzw. Generator zu berühren. Eine solche Möglichkeit ist besonders vorteilhaft, wenn das System für MÜ benutzt wird (im Rahmen des Projektes KIT-FAST werden Sätze aus Informationstexten der EG von Englisch nach Deutsch und umgekehrt übersetzt).

Als *einzelsprachspezifisch* werden die jeweiligen Mengen von ID-Regeln, LP-Aussagen, FCRs und Lexikoneinträgen betrachtet. Das *generelle* sprachliche Wissen besteht in der Bedingung für zulässige Bäume, die durch die Metasprache der GPSG gegeben ist.[56] Es umfaßt die in Abschnitt 6 beschriebene Metasprache und wird prozedural als Modul zur Anwendung von FCRs, MIPs und LP-Aussagen repräsentiert. Das Modul wird von Parser und Generator in exakt derselben Weise verwendet.

Abbildung 14 zeigt die Komponenten des Berliner GPSG-Systems. Es besteht aus den sprachlichen Wissensquellen, einem Parser, einem Generator und einem Grammatikeditor. Der Parser ist in [Weisweber 1987] beschrieben, und hinsichtlich des Generators (genauer: der beiden Generatoren) wird auf [Busemann 1990] verwiesen. Hier soll ihre Funktion nur kurz skizziert und dann die Aufgaben des Editors dargestellt werden, ohne den die Implementation des Berliner GPSG-Systems linguistisch unbefriedigend wäre.

Der Parser erzeugt aus einer Endkette—einem natürlichsprachlichen Satz—eine GPSG-Struktur. Es handelt sich um einen bottom-up operierenden Chart-Parser, der voraussetzt, daß Wortformen entsprechende lexikalische GPSG-Kategorien zugeordnet sind. Auf der Grundlage der GPSG-Struktur wird eine Funktor-Argument-Struktur (FAS, vgl. [Hauenschild/Umbach 1988]) als satzsemantische Repräsentation der Eingabe erzeugt.

Der Generator geht von einer FAS aus und erzeugt aus ihr—aufgrund einer anderen Verarbeitungsstrategie als der Parser—ebenfalls eine GPSG-Struktur. Auch er setzt voraus, daß den lexikalischen Kategorien Wortformen zugeordnet sind. Dann stellen die Blätter der GPSG-Struktur, von links nach rechts gelesen, die generierte Endkette dar.

Beide Prozesse benutzen die Implementation der MIPs als Unterprogramm, beide greifen auf dieselbe Grammatik zu, und beide gehen von derselben Relation zwischen lexikalischen Kategorien und Wortformen aus. Diese Relation wird mithilfe eines Stammformenlexikons und von Lemmatisierungs- bzw. Flexionsprozessen [Bu-

[55]Ich spreche nicht von *universellem* Wissen, da der Nachweis der Universalität aufgrund der beiden betrachteten Sprachen Deutsch und Englisch natürlich nicht zu führen ist.

[56]Diese Unterscheidung verläuft analog zu der von Programmen und Programmiersprachen. Ich werde den Vergleich in Abschnitt 12 weiter treiben durch die Unterscheidung zwischen interpretierten und compilierten „Programmen".

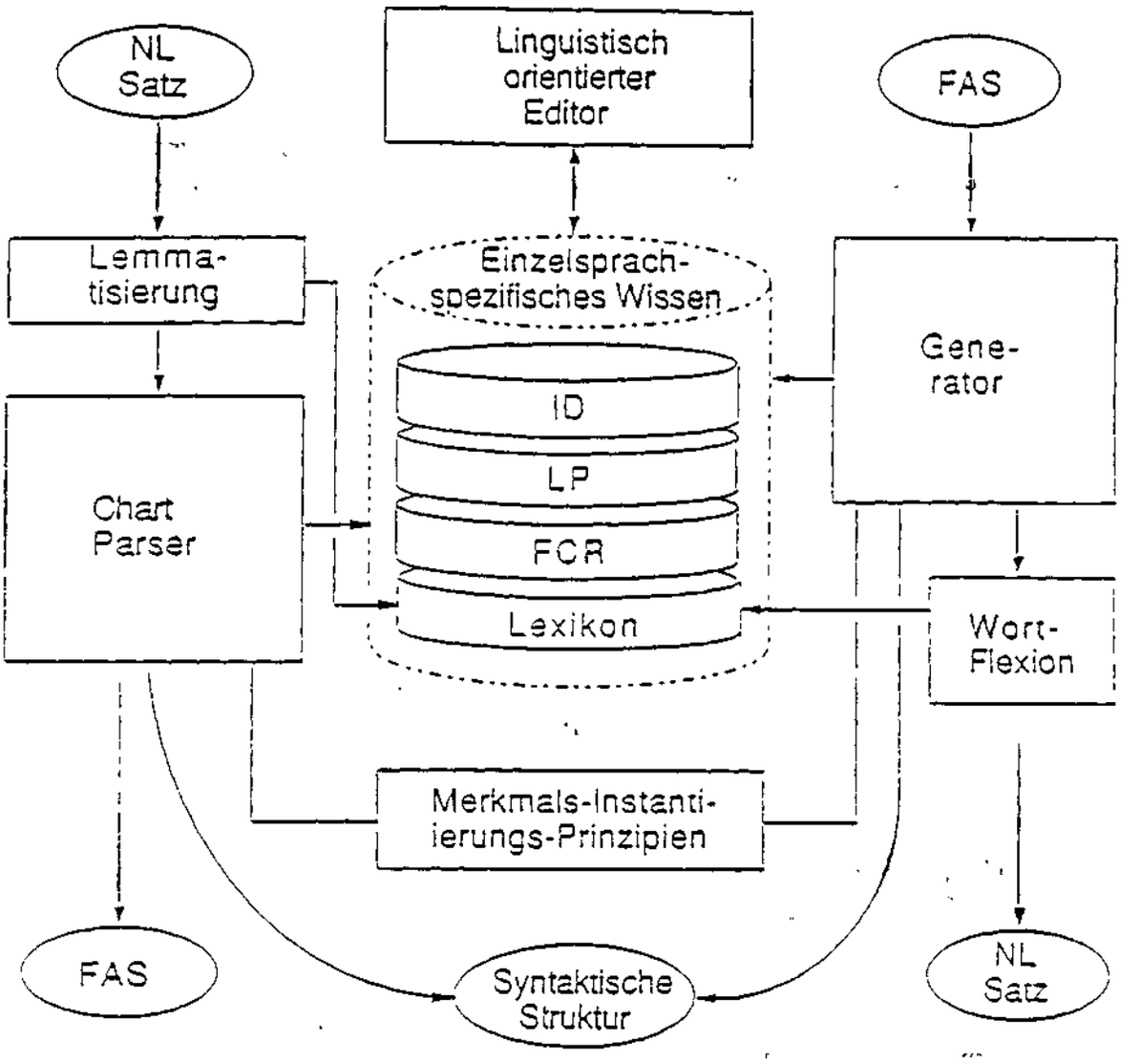

Abbildung 14: Der Aufbau des Berliner GPSG-Systems

semann 1988a] hergestellt, die vollkommen unabhängig von GFSG arbeiten. Die Kopplung zur Syntax wird in Abschnitt 10 diskutiert. Auf diese Weise wird die Auflistung aller Wortformen in einem Vollformenlexikon umgangen, die für stark flektierende Sprachen wie das Deutsche unökonomisch wäre.

Als unabdingbare Voraussetzung für die Erstellung einer GPS-Grammatik ist eine komfortable Entwicklungsumgebung notwendig. Es gab bereits für die mittlere GPSG ein Entwicklungs- und Testsystem, ProGram [Evans/Gazdar 1984]. Für eine konstruktive Variante von GKPS wurde im Rahmen des britischen Alvey-Projektes die portable und integrierte Entwicklungsumgebung GDE entwickelt [Briscoe *et al.* 1987]. Zum Testen von Regeln werden gewöhnlich Parser verwendet. Im vorliegenden Fall kann das GPSG-System als Testumgebung herangezogen werden (insbesondere stellt der Generator eine wichtige Testfunktion dar: er ermöglicht die Prüfung, ob eine Grammatik übergeneriert).

Der .Linguistisch orientierte Editor (LED) bietet die Unterstützung für die Formulierung verschiedener Regeltypen [Kindermann/Quantz 1987]. LED akzeptiert

Eingaben in einer linguistisch geeigneten externen Notation, die dann—nach einer Korrektheits- bzw. Plausibilitätsprüfung—in die interne Notation transformiert wird, die die Komponenten des GPSG-Systems verarbeiten.

LED verarbeitet Definitionen von Merkmalnamen, ihrer Wertebereiche und von Teilmengen der Merkmale (HEAD, FOOT, AGR). Die Aufgabe, eine Darstellung für Kategorien mit fünfundzwanzig bis dreißig Merkmalen anzubieten, die übersichtlich und linguistisch aussagekräftig ist, wird anhand von *Aliasnamen* gelöst, wie sie erstmalig in ProGram definiert wurden. Aliasnamen setzen die Definition von Kategorien (d.h. Namen und Wertebereiche der Merkmale) voraus. Sie bestehen aus drei Teilen:

- einem Namen (z.B. np),
- einer Anzahl definierender Merkmalspezifikationen (z.B. (n : +), (v : —}, (bar: 2)), und
- einer Menge instantiierbarer Merkmale (z.B. per, plu, gen, cas, usw.).

Ein Alias der Form np(PER,PLU,GEN,CAS,FLEX) steht für eine Kategorie, die an den definierenden Merkmalen des Alias entsprechend spezifiziert, an den instantiierbaren Merkmalen variabel und für sämtliche übrigen Merkmale Undefiniert ist. Die Grammatikschreiberin kann den instantiierbaren Merkmalen Werte zuweisen, und z.B. Kategorien wie np(PER,PLU,GEN,nom,FLEX) in ID-Regeln definieren.

Bei der Definition von ID-Regeln kann die Notation H (für Head) als Bestandteil der externen Notation verwendet werden. In der internen Notation wird die Spezifikation (head : +) benutzt.

Der Übergang von der externen in die interne Grammatiknotation schließt eine Anzahl von Vorverarbeitungsschritten ein. Der wichtigste stellt die Anwendung von Metaregeln dar, die nicht während des Parsings oder der Generierung erfolgt, sondern für die gesamte Grammatik im Vorhinein (vgl. [Thompson 1981]). Dies bringt beträchtliche Effizienzvorteile während der Verarbeitung, der dann die erweiterte Menge von ID-Regeln zugrundeliegt.

Bei der Erzeugung der internen Notation wird berücksichtigt, daß als Basis-Operation des GPSG-Systems die Prolog-Unifikation verwendet wird. Dies erwies sich aus Effizienzgründen als zweckmäßig, nachdem eine versuchsweise Implementation der PATR-Unifikation [Shieber *et al.* 1983] in Waterloo-Core-Prolog die verbrauchte CPU-Zeit erheblich erhöhte und lange Antwortzeiten infolge des Timesharing-Betriebs auf der benutzten IBM 4381-Anlage die Folge waren.

Die aus linguistischen Gründen wünschenswerten direkten Möglichkeiten, Disjunktion und Negation von Merkmalspezifikationen in Kategorien auszudrücken, fehlen bei der Kodierung als Prolog-Terme. Sie sind aber theoretisch auf der Ebene des Editors, in der externen Notation, gegeben. Die Übersetzung einer Negation (z.B. [-pas]) in die interne Notation erfordert die Aufzählung des Wertebereichs von vform ohne pas:

```
cat(..., VFORM, ...):-
   member(VFORM, [fin.inf.psp.zu]).
```

Die Integration von Editor und GPSG-System erfüllt somit beide Desiderata zugleich, indem sie linguistisch angemessene Ausdrucksmöglichkeiten (im Editor) und effiziente Verarbeitung (im Parser bzw. Generator) bietet.

LED ist in Prolog und Pascal auf einer IBM 4381-Anlage implementiert und nicht ohne weiteres portierbar. Eine Weiterentwicklung entsteht zur Zeit mit dem Editor für GPS-Grammatiken (EGG), der in Arity-Prolog 5.0 auf PC AT implementiert wird. In seiner endgültigen Ausbaustufe soll EGG Metaregeln und Lexikoneinträge erfassen können und in das GPSG-System integriert sein.

Abschließend folgen einige Daten zur Implementation des GPSG-Systems. Das GPSG-System wurde ursprünglich in Waterloo-Core-Prolog auf einer IBM 4381-Anlage unter dem VM/SP-Betriebssystem implementiert. Da nur die im KIT-Core-Standard [Bittkau *et al.* 1987] definierten Prädikate verwendet wurden, ist der Code kompatibel mit verschiedenen anderen Prolog-Dialekten, für die dieser Standard implementiert ist (z.B. Symbolics-Prolog, M-Prolog, Arity-Prolog). Unterschiedliche Versionen des Systems sind auf IBM 4381, auf Symbolies-Lispmaschinen 3640 auf SUN 340[57] und auf PC AT lauffähig.

8 Die Grammatikfragmente für Deutsch und Englisch

Die in dieser Arbeit verwendeten Beispiele aus dem Berliner GPSG-System basieren auf GPS-Grammatiken für Fragmente des Deutschen und des Englischen, wie sie Ende 1988 implementiert waren. In diesem Abschnitt werden die wesentlichen Charakteristika dieser Grammatiken beschrieben.

Beide Grammatiken decken einen ähnlichen Sprachumfang ab. Sie umfassen typische Wortstellungsphänomene und behandeln, teilweise in Anlehnung an [Uszkoreit 1984],

- subkategorisierte nominale und infinitivische Verbargumente,
- Verberst-, -zweit- und -endstellung,
- Topikalisierung der meisten Konstituenten,
- die üblichen Hilfs- und Modalverbkonstruktionen sowie
- Nominalgruppen mit subkategorisierten Argumenten und (eingeschränkt) Relativsätzen.

Eine erste Version der deutschen Grammatik beschreibt [Preuß 1987].- Ich gehe der Reihe nach auf die genannten Punkte ein.

Das Deutsche erlaubt eine stärkere Variation in der Wortstellung als das Englische. Daher wurde eine flache Struktur des Mittelfeldes angesetzt (z.B. (83)), in der auch das Subjekt auftreten kann. In Deklarativsätzen kann jede dieser Konstituenten

[57]Eine Adaption an Quintus-Prolog wurde von John Phillips an der Universität Tübingen im Juni 1988 durchgeführt.

topikalisiert werden (topikalisierte Konstituenten sind Extensionen von [+top]). Daher wird grundsätzlich die Topikalisierungsregel (84), verwendet.[58] Darüber hinaus können infinite Verben zusammen mit einem Objekt im Vorfeld stehen. Dies erfordert zusätzliche VP-Analysen, die ausschließlich für die Topikalisierung benötigt werden (85). Der Zusammenhang zwischen (85) und (83) ist systematisch und wird durch eine Metaregel erfaßt.

(83) S —> V[6], NP[nom], NP[acc]

(84) S —•+ X[+top], S[fm, +ac] / X[+top]

(85) VP—* V[6],NP[acc]

(86) S / VP —> NP[nom]

Die Einführung der Lücken erfolgt im Falle der VP durch die Regel (86) und bei anderen Konstituenten nach dem Vorschlag von [Jacobson 1987] (vgl. Abschnitt 4.4) durch folgende Metaregel:

(87) S —>W, Y

$$S/Y \xrightarrow{V} W$$

Es ergeben sich damit für (83) die zusätzlichen ID-Regeln (88) bis (90). Die Entscheidung für den Verzicht auf leere Ketten ist durch Schwierigkeiten mit bottom-up gesteuertem Parsing begründet, das vor jedem Wort der Eingabekette mit einer beliebigen Zahl leerer Ketten rechnen müßte. Als Preis für die Vermeidung dieses Problems müssen, wenn das Head topikalisiert wird, wie in (90), die Effekte der HFC in der ID-Regel nachgespielt werden.[59] .

(88) S / NP[nom] —> V[6],NP[acc]

(89) S / NP[acc] —-> V[6],NP[nom]

(90) S / V[6] —> NP[nom],NP[acc]

Verberst- und Verbletztsätze werden mithilfe der „flachen" Struktur (vgl. (83)) dargestellt. Die unterschiedliche Stellung des Verbs hängt von den Spezifikationen der HEAD-Merkmale ac (assertive clause) und mc (main clause) ab (vgl. [Uszkoreit 1984]). [+ac] erzwingt die Verwendung der Topikalisierungsregel, während [+mc] die Präzedenz des finiten Verbs vor allen nichtlexikalischen Schwestern fordert. [-mc] bewirkt die Endstellung des Verbs (vgl. die FCRs und LP-Aussagen (91)-(96)). Somit ergeben sich die folgenden Beschreibungen:

[+ac, -f-rnc] Deklarativsatz: *Hans liest das Buch.*

[+ac, —mc] Relativsatz: … *der das Buch liest.*

[—ac, -fmc] Frage-/Befehlssatz: *Liest Hans das Buch? Lies das Buch!*

[58]Da die Übereinstimmung zwischen slash-Wert und topikalisierter Konstituente nicht durch den Formalismus sichergestellt ist, müssen die entsprechenden Variablen in der ID-Regel kospezifiziert werden.

[59]Dies ist möglich, weil HEAD n FOOT = 0 und HEAD D AGR.

[-ac, -mc] Nebensatz: … *Hans das Buch liest,*

(91) [+ac, -rel][60] => [+mc]

(92) [+ac] => [fin]

(93) [+mc] => [fin]

(94) [bar:0, X] => [-ac, -mc] (X G (pas, prp, inf, zu})

(95) V[+mc] -< [--subcat]

(96) [-.subcat] -< V[-top, -mc]

Im Englischen wird die übliche Analyse von Deklarativsätzen mithilfe der ID-Regel (3) verwendet, die hier noch einmal als (97) erscheint.

(97) S ⟶ NP, VP

Somit sind die lexikalischen ID-Regeln für Verben VP-Expansionen. Die Topikalisierungsregel wird in der englischen Grammatik nur verwendet, wenn andere Konstituenten als das Subjekt im Vorfeld stehen sollen. Die Einführung von Lücken erfolgt mit einer ähnlichen Metaregel wie (87). Die Verbstellung kann einfacher geregelt werden als in der deutschen Grammatik, indem man Verberstsätze durch eine ID-Regel wie (56) (hier: (98)) beschreibt.

(98) S[+inv] —> V[+aux],NP,VP[-aux,bse]

Regeln dieses Typs gehen aus der Anwendung einer Metaregel („Subject-Aux Inversion") hervor.

Die in den Grammatiken zugrundegelegten X-Bar-Syntaxen sind für N, A und P gleich. Die maximale Projektionsebene ist zwei (NP, AP, PP). Die Ebene [bar:l] ist lediglich für N ausgeprägt und dient zur Adjunktion von APs oder (in der englischen Grammatik) Relativsätzen. Die lexikalische Ebene [bar:0] ist für alle vier Hauptkategorien vorhanden. Bezüglich V unterscheiden sich die X-Bar-Syntaxen. Die maximale Projektionsebene in der englischen Grammatik ist auch hier zwei. VP und S unterscheiden sich, analog zu GKPS, durch unterschiedliche Spezifikationen von subj: VP ist V[bar:2, -subj], während S als V[bar:2, +subj] kodiert ist. In der deutschen Grammatik wurde in Anlehnung an [Uszkoreit 1984] als maximale Projektion drei gewählt (für S); das Merkmal subj gibt es nicht. Eine VP wird einfach als V[bar:2] kodiert.

Das Konzept der Subkategorisierung ist in den beiden Grammatiken unterschiedlich. Während in der englischen Grammatik das von GKPS Verwendung findet (vgl. Abschnitte 2.2 und 4.4), wird in der deutschen Grammatik die Subkategorisierung als die syntaktische Umgebung von Lexemen angesehen, die durch eine lexikalische ID-Regel beschrieben ist und nicht ausschließlich als Eigenschaft von Lexemklassen. Infolgedessen subkategorisieren Verben auch ihre Subjekte. Bei Passivsätzen ändern Verben ihre Subkategorisierung, z.B. besitzt das transitive Verb *bitten* mit (subcat : 6) bei agenslosem Passiv die Subkategorisierung von intransitiven Verben

[60]Das Merkmal rel wird für die Beziehung zwischen einem Relativpronomen und seinem Bezugsnomen benötigt (s.u.); die vorliegende Spezifikation schließt also Relativsätze aus. Bei ihnen wird die Topikalisierungsregel verwendet, wobei Verbletztstellung gefordert ist.

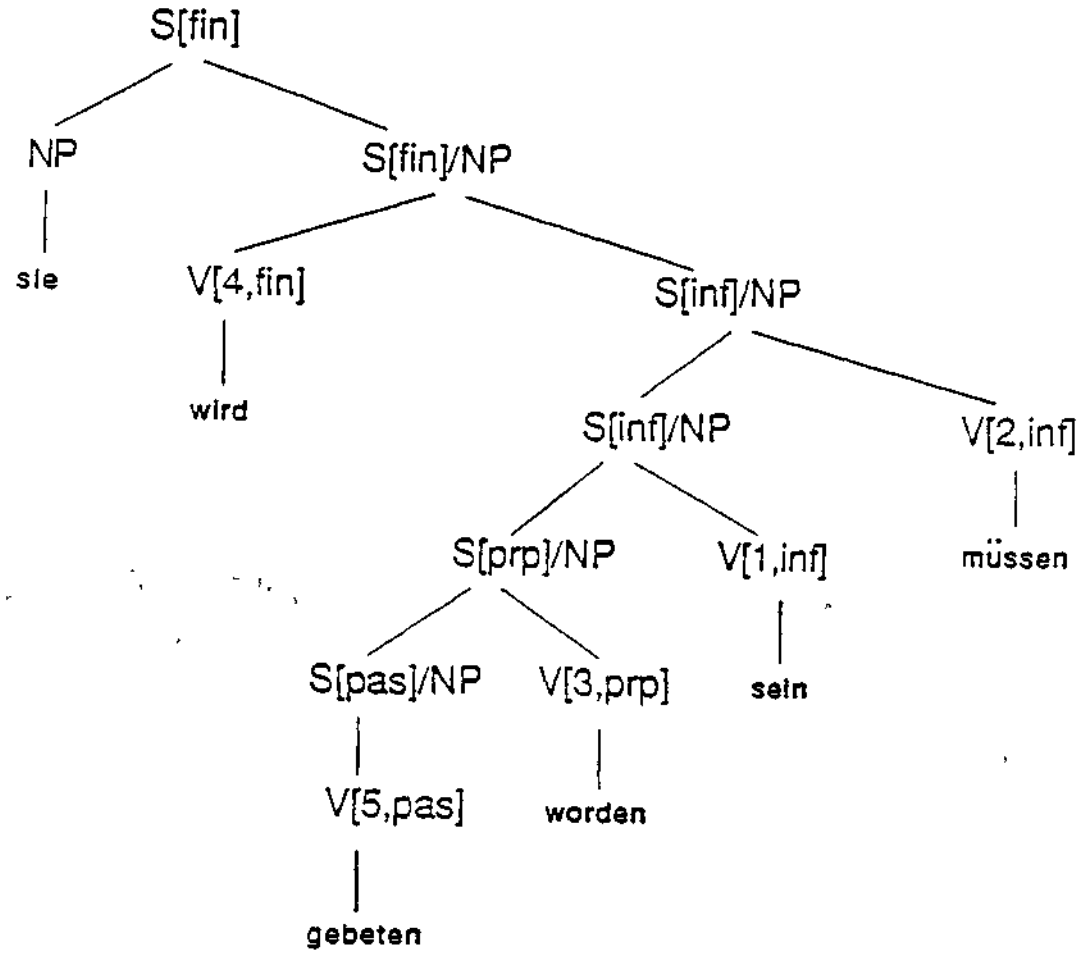

Abbildung 15: Der Hilfsverbkomplex in der deutschen Grammatik

({subcat : 5».

Hilfs- und Modalverben werden in beiden Grammatiken nach ähnlichen Prinzipien kodiert, die [Uszkoreit 1984] beschreibt und die auf [Gazdar *et al.* 1982] zurückgehen. Eine Anzahl binär verzweigender ID-Regeln führen das Hilfsverb (als Head) ein und eine entsprechend für vform spezifizierte Kategorie, die sonst im wesentlichen mit der Mutter übereinstimmt (vgl (99)-(102)).

(99) 3 —> V[2], S[inf] (Modalverb)

(100) S—>V[l],S[psp] (Perfekt-Hilfsverb) ' *'

(101) S—>V[3],S[pas] (Passiv-Hilfsverb)

(102) S —> V[4, fin], S[inf] (Futur-Hilfsverb)

Für das Deutsche gilt (und ist in der Grammatik kodiert), daß höchstens je eine Futur- und Passivspezifikation in der syntaktischen Struktur eines Verbkomplexes auftreten darf, wobei das Futur am obersten und das Passiv am tiefsten Verb stehen muß (Genaueres siehe Abschnitt 10). Abbildung 15 zeigt ein Beispiel mit den obigen Regeln.

Abtrennbare Verbpräfixe sind nicht im deutschen Fragment eingeschlossen. Ein neuerer Vorschlag findet sich in [Volk 1988].

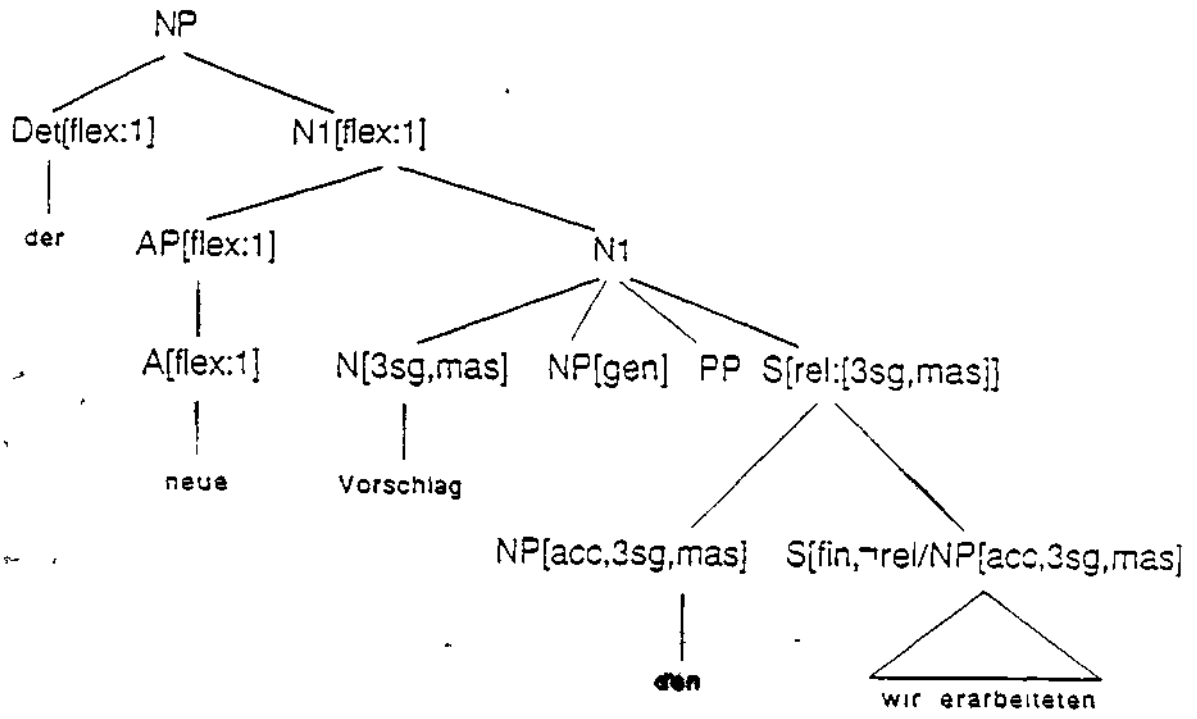

Abbildung 16: Der NP-Komplex in der deutschen Grammatik

Die Nominalgruppen-Syntax ist wenig ausgearbeitet (vgl. Abbildung 16). Es gibt NP-Expansionsregeln mit und ohne Det-Tochter. Eine Expansionsregel für Nl führt eine AP ein. Der Abhängigkeit der Form attributiv gebrauchter Adjektive von der Präsenz und der Art des Determiners wird durch das Merkmal flex Rechnung getragen. Nl-Expansionen können auch lexikalische Regeln der Form (103) sein.

(103) Nl —* N[5], (NPfeen]), (PP[fuer]), (S[finj)

Nur der Head ist obligatorisch; Genitivattribut, Präpositionalattribut und Relativsatz sind optional. LP-Aussagen sorgen dafür, daß die Abfolge „N-NP-PP-S" strikt eingehalten wird. In der Abbildung wird die Kongruenz zwischen Relativpronomen und Bezugsnomen dargestellt. Sie erfolgt mithilfe eines kategorienwertigen FOOT-Merkmals rel, das die Spezifikationen für per, plu und gen transportiert. In den lexikalischen ID-Regeln wie (103) sind diese Merkmale an N und im rel-Wert von S kospezifiziert.

Wie viele Beschreibungen des NP-Komplexes im Deutschen vermeidet dieser Ansatz nicht gewisse strukturelle Mehrdeutigkeiten bei Genitivattributen. Sätze wie *wenn Sängerinnen Kollegen Melodien vorsingen* erhalten unzulässige Genitiv-Analysen infolge unterschiedlicher (intransitiver, transitiver und ditransitiver) Subkategorisierungen des Verbs *vorsingen.* [Schachtl 1988] gibt eine Lösung im Rahmen von LFG an.

Die Nominalgruppen-Syntax in der englischen Grammatik unterscheidet sich nur marginal von der der deutschen. Relativsätze werden analog zur AP durch N1 eingeführt. Das Merkmal rel ist nicht kategorienwertig, sondern binär. Die Grammatik regelt nicht, ob als Relativpronomen *who(m)* oder *which* bzw. *that* zu verwenden ist.

In beiden Grammatiken wird das binäre Merkmal agr ausschließlich durch FCRs

spezifiziert. Extensionen von [+agr] sind in der deutschen Grammatik folgende Kategorien: S[-rel, comp:nil][61], VP, V[fin], NP[nom], Nl, N, Det[-rel], AP, A. In der englischen Grammatik sind es folgende Kategorien: VP, V[fin], NPfnom], Nl, N, Det, [+rel].

In beiden Grammatiken wurden LP-Aussagen nur zur Beschränkung der Wortstellung aufgrund „harter" syntaktischer Kriterien verwendet. Sie regeln also nicht die Wortstellung im Mittelfeld, die in beiden Grammatiken völlig frei gelassen ist. Dies ist zwar zu liberal, aber wie in Abschnitt 4.3 gezeigt wurde, ist eine linguistisch fundierte Behandlung des Problems nicht im Rahmen der Syntax allein möglich. Die Implementation von komplexen LP-Aussagen in einem Mehrebenen-Modell mit GPSG ist eine computerlinguistisch interessante Aufgabe, die jedoch nicht Gegenstand dieser Arbeit ist.

Zum Abschluß folgen einige Angaben über die Größe der Grammatiken. Die deutsche Grammatik umfaßt 27 Merkmale (davon 16 binär und 2 kategorienwertig), 55 ID-Regeln (9 mit einer Tochter, 31 mit zwei, 9 mit drei und 6 mit vier Töchtern), davon 21 aufgrund von 3 Metaregeln erzeugt, 12 LP-Aussagen und 17 FCRs. Das Lexikon umfaßt etwa 250 Stammformen.

Die englische Grammatik umfaßt 25 Merkmale (davon 16 binär und eines kategorienwertig), 76 ID-Regeln (18 mit einer Tochter, 41 mit zwei und 17 mit drei Töchtern), davon 40 aufgrund von 3 Metaregeln erzeugt, 7 LP-Aussagen und 13 FCRs. Das Lexikon umfaßt etwa 150 Stammformen.

9 Prozedurale Aspekte der Strukturbildung

Jedes Verfahren zur maschinellen Sprachverarbeitung auf der Grundlage der konstruktiven Version von GPSG ruht auf zwei Säulen, die die in den vorigen Abschnitten formulierte Modularisierung von Formalismus, Grammatik und Verarbeitungsstrategie widerspiegeln:

- Konstruktion der syntaktischen Struktur nach Maßgabe der ID-Regeln
- Spezifizierung der Struktur aufgrund der FCRs, MIPs und LP-Aussagen

Da ID-Regeln in ihren Kategorien gewöhnlich stark unterspezifiziert sind, kann die aufgebaute Struktur als Skelett betrachtet werden, zu dem das Fleisch noch hinzugefügt werden muß. Dies entspricht der Aufgabe der MIPs, die die Kategorien weitgehend spezifizieren. Sie spezifizieren sie aber nicht immer vollständig; typischerweise wird bei Kospezifikation variabler Merkmalwerte ein endgültiger Wert zu einem späteren Zeitpunkt durch Konstruktion instantiiert.

In diesem Abschnitt soll die Frage aufgegriffen werden, wie die Instantiierung durch Konstruktion mit der Instantiierung durch FCRs und MIPs interagiert. Diese Frage resultiert direkt aus dem konstruktiven Ansatz, in welchem ein Parser oder ein Generator entscheiden muß, welche ID-Regeln für die Strukturbildung herangezogen werden sollen. Diese Entscheidung hängt ab von dem noch zu verarbeitenden Teil der Eingabekette bzw. der Ausgangsstruktur und von der bisher erzeugten syntaktischen

[61] Dies beschreibt solche Sätze, die weder Relativsätze noch Komplementsätze sind.

Teilstruktur (die Erzeugung isolierter, zulässiger lokaler Bäume bedeutet drastische Übergenerierung und ist daher aus Effizienzgründen nicht sinnvoll).

9.1 Verarbeitungsstrategien

Eine syntaktische Struktur wird durch sukzessive Expansions- bzw. Reduktionsschritte erzeugt, wobei jeweils genau ein lokaler Baum eingeführt wird, um den die Struktur wächst. Ich nenne diese Struktur *Gesamtstruktur* und die sie erweiternden Operationen *strukturbildende Aktionen* (SBAs).

Expansionsschritte sind Top-Down-Operationen: Die Mutter eines neu einzuführenden lokalen Baumes unifiziert mit einer nichtterminalen Blattkategorie aus der Gesamtstruktur. Reduktionsschritte sind Bottom-Up-Operationen: Alle Töchter eines neu einzuführenden lokalen Baumes unifizieren mit Wurzelkategorien aus der Gesamtstruktur.

Die erzeugten Strukturen müssen durch die MIPs und LP-Aussagen weiter spezifiziert werden, um zulässig im Sinne von GPSG zu werden. Dies kann entweder unmittelbar nach Beendigung einer SBA geschehen, oder nachdem alle Kategorien des neu eingeführten lokalen Graphen an SBAs beteiligt waren. Hieraus ergeben sich vier unterschiedliche Strategien:

1. Struktur wird durch Expansion, also top-down, konstruiert. Die MIPs werden unmittelbar nach jeder SBA angewendet, also ebenfalls top-down.

2. Struktur wird durch Reduktion, also bottom-up, konstruiert. Die MIPs werden unmittelbar nach jeder SBA angewendet, also ebenfalls bottom-up.

3. Struktur wird durch Expansion, also top-down, konstruiert. Die MIPs werden angewendet, nachdem alle Töchter zulässige Bäume dominieren, also bottom-up.

4. Struktur wird durch Reduktion, also bottom-up, konstruiert. Die MIPs werden angewendet, nachdem die Mutter Blattkategorie eines zulässigen Baumes geworden ist, also top-down.

Diese Strategien lassen sich anhand von Abbildung 17 veranschaulichen. Abbildung 17a zeigt lokale Bäume, die von ID-Regeln bzw. von Lexikoneinträgen projiziert werden und genau die ererbten Spezifikationen enthalten (wie bisher sind aus Platzgründen nicht alle dargestellt). Die lokalen Bäume können durch SBAs miteinander kombiniert werden und ergeben so die Gesamtstruktur in Abbildung 17b, wobei Merkmale durch Konstruktion instantiiert werden. Das Resultat der Anwendung von FCRs, MIPs und LP-Aussagen auf die lokalen Bäume der Gesamtstruktur zeigt Abbildung 17c.

Jede SBA erzeugt einen Teil von Abbildung 17b, während jede Anwendung der FCRs, MIPs und LP-Aussagen einen Teil von Abbildung 17c generiert. Dabei werden variable Merkmalspezifikationen kospezifiziert.

Nicht alle vier Strategien sind effizient einsetzbar. Zum einen liegt das an der Verarbeitungsweise der Ausgangsstrukturen (bei der Generierung; vgl. [Busemann

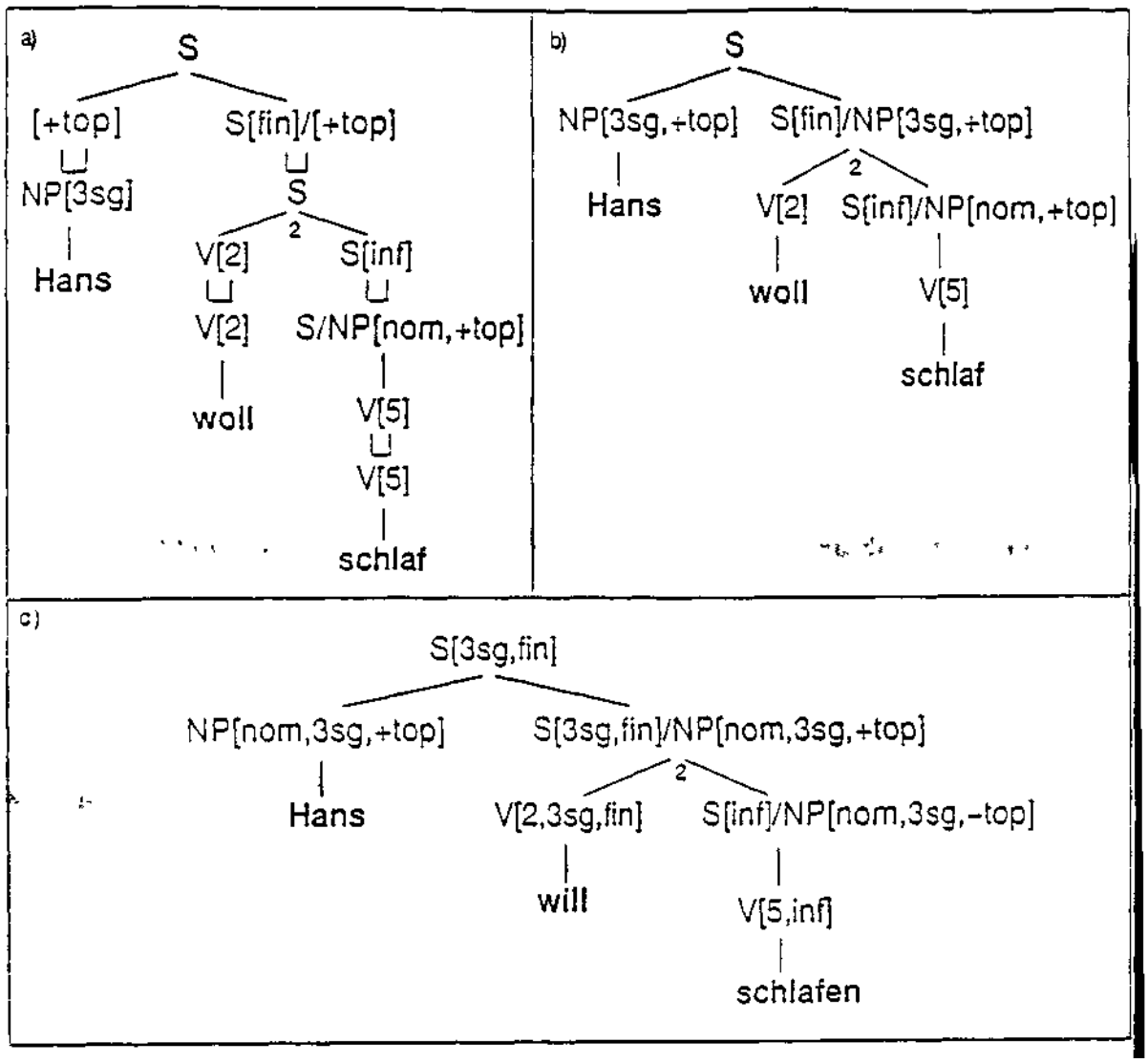

Abbildung 17: Strukturbildung und Merkmalsinstantiierung

1990, Kap. 6]) bzw. der Endkette (beim Parsing). Zum ändern liegt es am Formalismus der GPSG, genauer: am AP und am FFP. Dies soll im folgenden erläutert werden.

Das AP setzt angemessene Spezifikationen von agr an den Töchtern eines lokalen Baumes voraus. Bei einer top-down vorgehenden Merkmalsinstantiierung kann dies nicht sichergestellt werden: Baum l kann als Füller eine NP[nom] erhalten, die durch eine FCR mit [+agr] zu spezifizieren ist oder eine NP[acc], bei der dies nicht der Fall ist. Welcher Art der Füller ist, muß sich erst noch herausstellen. Bei einer bottom-up vorgehenden Merkmalsinstantiierung wird zuerst der slash-Wert für die Lücke aufgesucht und an topologisch höhere lokale Bäume transportiert. In Baum l ist der slash-Wert (und damit die Tochter [-t-top]) durch Konstruktion ausreichend spezifiziert, damit FCRs *vor* der Anwendung des AP das Merkmal agr entsprechend instantiieren können.

Hinsichtlich des FFP ist folgende Überlegung wesentlich. FOOT-Merkmale können an verschiedenen Töchtern und der Mutter unifizieren. Anhand der ID-Regeln läßt sich nicht bestimmen, an welchen Töchtern ein FOOT-Merkmal gemäß

FFP im Einzelfall instantiiert werden soll. Diese Unterbestimmtheit der ID-Regeln führt zu nichtdeterministischem Verhalten bei einer top-down vorgehenden Merkmalsinstantiierung. Welche der Töchter für ein FOOT-Merkmal zu instantiieren sind, muß geraten werden; für eine fundierte Entscheidung enthält der lokale Baum zu wenig Information.

In Abbildung 17 tritt dieses Problem bei Baum 2 auf. S und V[2] könnten beide für slash spezifiziert werden.[62] Allerdings ließe sich entgegnen, daß Modalverben niemals eine Lücke einführen und V[2] daher in der ID-Regel mit (slash : -,) zu spezifizieren seien. Überzeugendere Beispiele bestehen in Analysen für parasitäre Lücken (vgl. Abschnitt 4.6) oder in Strukturen mit Projektionen der ID-Regel (59) (vgl. Abbildung 10 auf Seite 57), die hier als (104) wiederholt wird.

(104) VP^ VP,PP . . _ . _ . . .

Bei einer Bottom-Up-Strategie der Merkmalsinstantiierung liegt genügend Information in einem lokalen Baum vor, um eindeutig zu entscheiden, an welchen Konstituenten ein FOOT-Merkmal zu instantiieren ist, denn es steht fest, welche Töchter für ein FOOT-Merkmal den Wert „-," besitzen, weil alle von den Töchtern dominierten Teilbäume bereits zulässig gemäß dem FFP sind.

Somit sind die Strategien l und 4 inhärent ineffizient und werden nicht verwendet. Die beiden anderen Strategien wurden im Berliner GPSG-System eingesetzt. Strategie 2 wurde im Parser implementiert. Sie beginnt bei Kategorien, die durch das Lexikon (bzw. die Lemmatisierungskomponente) zugelassen werden. Jedem Reduktionsschritt folgt die Anwendung der MIPs und der LP-Aussagen auf den neu in die Gesamtstruktur eingefügten lokalen Baum. So wird Information aus den lexikalischen Kategorien „den Baum hinauf gereicht" und kann die Menge der weiteren möglichen Reduktionsschritte, die durch die Grammatik zugelassen werden, beschränken. Beispielsweise kann ein lokaler Baum, der das Futur-Hilfsverb aufgrund der ID-Regel (105) einführt, nicht mit einer Extension von S[psp] in der Gesamtstruktur kombiniert werden, da die Mutter durch HFC eine Extension von S[fin] geworden ist.

(105) S—>V[4,fin],S[inf]

Strategie 3 wird im Generator benutzt. Sie beginnt mit einem lokalen Baum (mit Mutter S etwa), dessen Kategorien nur aufgrund einer ID-Regel spezifiziert sind. Die Expansionsschritte werden durch lexikalische Insertion abgeschlossen. Nachdem alle Töchter eines lokalen Baumes entweder Terminalsymbole sind oder zulässige lokale Bäume dominieren, werden die FCRs und MIPs auf den lokalen Baum angewendet. Die LP-Aussagen operieren als generative Komponente, die alle LP-zulässigen Permutationen der Töchter (und damit eine Anzahl lokaler Bäume) erzeugt. Für die Konstruktion der Struktur steht bei diesem Verfahren keine Information aus der Anwendung der MIPs zur Verfügung. Die Expansionsschritte werden maßgeblich durch die Ausgangsstruktur des Generators gesteuert.[63]

[62]Zwar sind lexikalische Kategorien Extensionen von [-slash], doch dies ergibt sich aus dem *Lexikon* und nicht aus den ID-Regeln.

[63]Ausführliche Beispiele für den Einsatz von Strategie 3 bei der Generierung enthält [Busemann 1990].

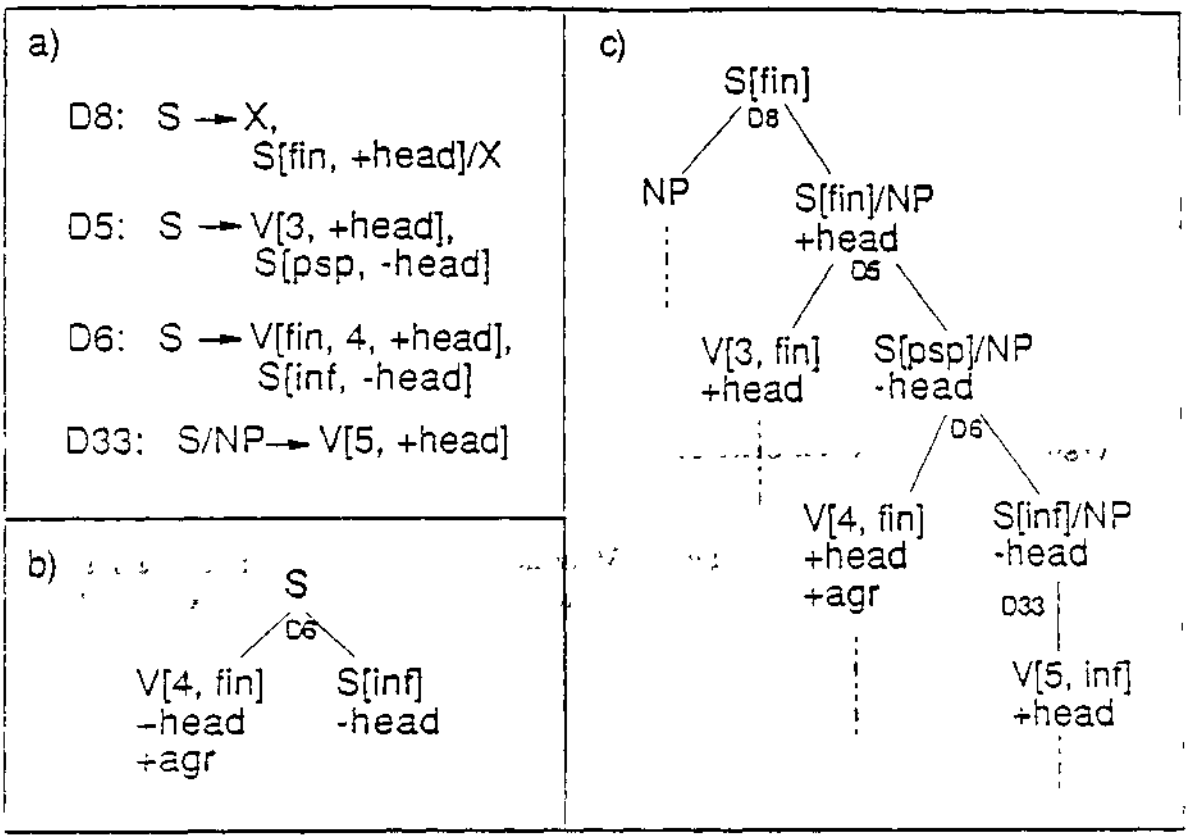

Abbildung 18: Strukturbildung bei der Generierung

In der beschriebenen Interaktion von Verarbeitungsstrategien, GPSG-Formalismus und Grammatiken liegt ein Problem verborgen, das im folgenden Abschnitt diskutiert wird.

9.2 Instantiierung durch Konstruktion und HFC

Bei den beiden Strategien 2 und 3 erfolgt die Instantiierung durch Konstruktion vor der Instantiierung durch die MIPs, doch während bei Strategie 2 die Töchter in einem lokalen Baum durch Konstruktion spezifischer werden, ist es bei Strategie 3 die Mutter. Insofern treffen die MIPs unterschiedliche Situationen an, wie an dem obigen Beispiel leicht ersichtlich ist. Abbildung 18 zeigt (a) vier ID-Regeln und (b) einen lokalen Baum, der von (105) (D6) projiziert ist. Diese Regel führt das Futurhilfsverb ein, das immer finit ist. Sie verlangt aber nicht, daß in einem Verbalkomplex nur ein finites Verb vorkommen darf. Der Generator erzeugt mithin Strukturen wie in Abbildung 18c mit einem zweiten finiten Verb. Eines wird durch (105) eingeführt (lokaler Baum Do), und das andere wird durch HFC in Baum D5 instantiiert.

Natürlich müssen solche Strukturen ausgeschlossen werden, und erneut ist die Frage zu beantworten, welcher von mehreren Lösungswegen einzuschlagen ist. Man kann die Grammatik ergänzen, die Verarbeitungsstrategie ändern oder den Formalismus modifizieren:

- Ist die Instantiierung durch Konstruktion von den MIPs zu ignorieren?

- Ist die Grammatik zu liberal; sollte sie genau ein finites Verb pro Verbalkomplex zulassen?
- Ist HFC zu liberal; sollte sie einen lokalen Baum wie D6 ablehnen?

Der erste Weg verzichtet auf den Vorteil, den die Strategien 2 und 3 gegenüber den Strategien l und 4 besitzen. Das effiziente Funktionieren von AP und FFP basiert ja gerade auf Instantiierung durch Konstruktion. Allein für HFC ist ein Ausschluß der durch Konstruktion erzeugten Spezifikationen erwägenswert. HFC würde dann nur auf den ererbten und den durch FCRs, AP und FFP erzeugten HEAD-Spezifikationen operieren. Das Kontrollregime wird damit aber in hohem Maße kompliziert, denn HFC kann nun die Rücknahme einer SBA erzwingen (wie etwa bei Baum D6 im obigen Beispiel).

Die stärkere Spezifikation der Kategorien in den ID-Regeln ist immer ein Weg, Probleme mit Expansionsstellen zu lösen. Er wird in zahlreichen computerlinguistischen Varianten von GPSG gewählt, ohne daß klar genug gesagt wird, daß man damit u.U. gegen die Theorie verstößt, deren oberstes Ziel es ist, möglichst wenig mutwillige Festlegungen machen zu müssen, aber möglichst viel aus allgemeinen Prinzipien folgen zu lassen. Im vorliegenden Fall genügt es, die Mutter von (105) ebenfalls mit [fin] zu spezifizieren (nicht aber, diese Spezifikation statt an die Head-Tochter an die Mutter zu schreiben, weil dann nämlich der *Parser* beliebige Formen des Heads akzeptieren würde). Dies ist die gewählte Lösung, aber sie kann aus dem genannten Grund nicht völlig befriedigen.

Der dritte Weg schließlich verlangt eine Revision der MIPs, und wahrscheinlich ist dies erfolgversprechend. Wie in Abschnitt 4.8 erwähnt wurde, zeigt [Preuß 1989] einen Weg auf, die strikte Definiton der HFC aus der mittleren GPSG zu benutzen und dennoch dieselben sprachlichen Phänomene wie GKPS abzudecken. Damit würde die fehlende Information nicht in die ID-Regeln gesteckt werden sondern aus den MIPs hervorgehen, ganz wie es GKPS entspräche. HFC verlöre den Charakter eines Defaultprinzips und wäre unabhängig von vorher erfolgten Merkmalinstantiierungen. Insbesondere würde dann die verarbeitungsstrategisch bedingte Instantiierung durch Konstruktion die Arbeit der MIPs nicht mehr beeinflussen, denn HFC könnte in der Vorverarbeitungsphase auf die ID-Regeln angewendet werden. Die Unabhängigkeit von Verarbeitungsstrategien, Formalismus und Grammatiken wäre mithin klarer verwirklicht.[64]

10 Der Lexikonzugriff bei der Generierung

Vor dem Hintergrund der Trennung von Generierungsverfahren, generellem sprachlichen und einzelsprachspezifischem Wissen werden zwei Arten von Abhängigkeiten zwischen diesen Komponenten bei der Einfügung lexikalischer Elemente in die syntaktische Struktur aufgezeigt [Busemann 1988b]. Eine Gruppe von Phänomenen,

[64]Man beachte, daß dieses Problem dem Anspruch an eine von Verarbeitungsgesichtspunkten unabhängige Formulierung von GPS-Grammatiken entsprang. Es verschwindet sofort, wenn bei allen Verfahren (Parsing und Generierung) nur *eine* bestimmte Verarbeitungsstrategie (entweder 2 oder 3) verwendet würde.

paradigmatische Lücken, wird kritisch bei der Verwendung eines Stammformenlexikons und einer separaten Flexionskomponente zur Erzeugung von Vollformen (Abschnitt 10.2). Eine zweites Problem stellt die Auswahl von Perfekt-Hilfsverben im Deutschen dar (Abschnitt 10.3). Zunächst wird jedoch begründet, warum die Verwendung eines Stammformenlexikons für die Generierung notwendig ist.

10.1 Generierung mit Stammformenlexika

In GKPS wird wenig über das Lexikon ausgesagt. Für jede Wortform gibt es mindestens einen Eintrag bestehend aus phonologischer Form, syntaktischer Kategorie, einer eventuellen Darstellung der phonologischen Form von morphologisch irregulär gebildeten Wortformen und einer Bedeutungsrepräsentation (GKPS, S. 34). Nicht behandelt wird die Beziehung zwischen flektierten Wortformen und syntaktischer Kategorie.

Im Berliner GPSG-System[65] wird diese Beziehung durch eine nachgeschaltete Flexionskomponente hergestellt, die mit einem Stammformenlexikon (SFL) zusammenarbeitet. Das lexikalische Wissen umfaßt folgende Informationen:

- ein Wortstamm (i.a. die längste Zeichenkette, die allen flektierten Formen gemeinsam ist),
- eine unterspezifizierte Kategorie, die die Kategorien aller flektierten Formen als Extensionen hat,
- morphologische Charakterisierungen (Umlautfähigkeit, abgeläutete Formen, Flexionsklassen usw.)
- Markierungen und Bedeutungsrepräsentationeu, auf die hier nicht eingegangen wird.

Die beiden ersten Punkte lassen sich zu Regeln des Typs Kategorie —>• Stamm zusammenfassen. Diese Regeln werden von der Syntax zur Terminierung der Bäume benötigt und wie ID-Regeln behandelt. Die morphologischen Charakterisierungen werden ausschließlich von der nachgeschalteten Flexionskomponente benutzt. Das Verb *bitten* besitzt z.B. folgende lexikalische Information:

Regel:	V[6, paux:h] —>	*bitt*
Morpholog. Char.:	Grundform C:	*bat*
	Grundform D:	*baete* , . s
	Passiv/Perfekt:	*gebeten*
	Konjugation:	Typ4

Der Stamm wird für die Flexion der regulär gebildeten Formen benutzt; alle anderen flektieren aufgrund der abgeläuteten Grundformen C und D (vgl. [Busemann 1988a]). Dabei bestimmt der Konjugationstyp, welche Formen mit welchem Stamm

[65] Ich konzentriere die Diskussion auf die Generierung des Deutschen, ohne sie damit grundlegend einzuschränken.

gebildet werden. Die syntaktische Kategorie beschreibt ein transitives Verb, dessen perfektive Formen mit *haben* gebildet werden. Sie ist stark unterspezifiziert und subsumiert Kategorien für alle Wortformen, die mit dem Stamm gebildet werden.

Eine grundsätzliche Alternative zum SFL ist das Vollformenlexikon (VFL), das die zulässigen Lexikoneinträge auflistet, wobei die Kategorien vollständig spezifiziert sind. Wie bereits erwähnt, sprechen Ökonomie-Aspekte gegen die Verwendung eines VFL im Deutschen.

Ein weiterer Grund für die Verwendung eines SFL ergibt sich aus der folgenden Argumentation. Bei der Konstruktion der Struktur während der Generierung müssen im Zuge der lexikalischen Insertion präterminale Kategorien mit den Kategorien im Lexikon unifizieren. Die durch diese Unifikation entstehenden Kategorien sind noch immer erst teilweise spezifiziert, denn die MIPs haben noch nicht ihr Werk getan. So stehen z.B. Person und Numerus in der Kategorie eines zu verbalisierenden Verbs noch nicht fest.

Erst wenn ein Baum zulässig ist, steht fest, daß an den präterminalen Kategorien alle Informationen angesammelt sind, die für die Flexion des Wortstammes durch die nachgeschaltete Morphologiekomponente erforderlich sind. Diese benutzt dann die Stammform und die morphologischen Charakterisierungen im Lexikon zur Erzeugung einer flektierten Wortform.

Ein solcher Aufbau von vollständig spezifizierten Kategorien kann erst durch die Verwendung eines SFL effizient realisiert werden; aus einem VFL kann der Generator nicht eindeutig einen Eintrag auswählen, da zum Zeitpunkt der lexikalischen Insertion die präterminale Kategorie gewöhnlich noch zu unspezifisch ist. Solange der Generierungsprozeß nicht abgeschlossen ist, könnten jederzeit Merkmalspezifikationen an die präterminale Kategorie gelangen und die Auswahl eines anderen Eintrags des VFL erzwingen.

Durch die Verwendung eines SFL kann das sonst notwendige Backtracking vermieden werden. Dies wird in den folgenden Abschnitten gezeigt.

10.2 Paradigmatische Lücken

Die naheliegende Gleichung „VFL = SFL + Flexionsprozesse" ist nicht ohne weiteres richtig. Ein SFL erlaubt es, alle möglichen Extensionen der lexikalischen Kategorien zu generieren; d.h. die Lexikonregeln in einem SFL umfassen das komplette durch die Definition der möglichen Kombinationen von Merkmalspezifikationen definierte Paradigma. Hingegen kann ein VFL die Verwendung bestimmter Kategorien ausschließen, indem die entsprechenden Einträge einfach fehlen. Damit können die zahllosen *paradigmatischen Lücken,* d.h. das Fehlen von bestimmten Formen einzelner Wörter, auf einfache, aber implizite Weise erfaßt werden (z.B. bildet das Futur-Hilfsverb *werden* kein Imperfekt, das transitive Verb *bekommen* kein Passiv usw.). Diese restriktive Eigenschaft des VFL muß bei einem SFL nachgespielt werden. Doch zunächst ein Beispiel.

In Abbildung 19 erlauben die Syntaxregeln (l)-(4) die Generierung der Bäume s_1 und s_2.[66] Der Merkmaltransport erzeugt die Spezifikationen des Merkmals vform

[66]Zur Erinnerung: V[l] stellt ein Perfekt-Hilfsverb, V[3] das Passiv-Hilfsverb und V[5] ein intransitiv subkategorisiertes Verb dar.

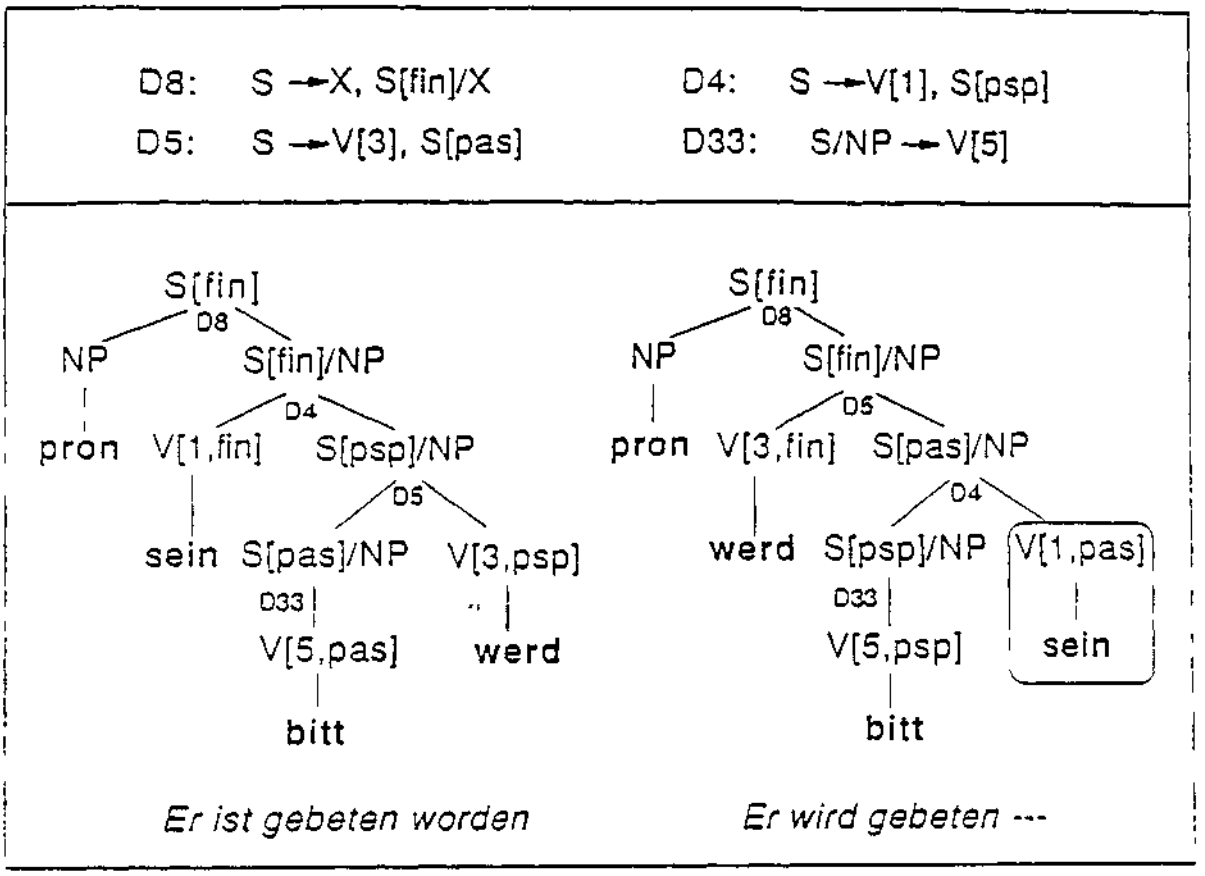

Abbildung 19: Zur lexikalischen Beschränkung syntaktischer Strukturen

an der präterminalen Verbkategorie. Wie ersichtlich, verlangt die markierte Kategorie V[l, pas] in *s^*, ein Passiv-Partizip des Perfekt-Hilfsverbs zu bilden, was die lexikalische Kategorie von sein im SFL nicht verbietet (sie ist ja unspezifiziert in Bezug auf vform).

In einem VFL würde ein Eintrag für V[l, pas] einfach fehlen und der markierte Teilbaum im Verlauf der syntaktischen Generierung als unzulässig erkannt werden. Bei einem SFL hingegen ist der Ausschluß von 52 auf der syntaktischen Ebene nicht möglich. Dann allerdings muß die Morphologiekomponente feststellen, daß V[l] kein Passiv-Partizip besitzt: im morphologischen Lexikoneintrag fehlt das Merkmal passiv/perfekt; stattdessen gibt es ein Merkmal perfekt mit dem Partizip als Wert. Somit resultiert als Ergebnis der gewählten Modularisierung der Generierung mit *3-2* eine nach dem GPSG-Formalismus zulässige Struktur ohne Endkette. Eine zusätzliche Aufgabe der Morphologiekomponente ist hierbei, lexikalisch unzulässige Merkmalkombinationen zu erkennen und die Generierung der Endkette abzubrechen. .

Neben dieser im Berliner GPSG-System realisierten Methode kann man das Problem auch innerhalb der Syntaxkomponente lösen, indem man ein zusätzliches binäres Merkmal passive verwendet, dessen Wert für alle Verben lexikalisch angibt, ob sie ein Passiv bilden. Die im Verlauf der Generierung erforderliche Relation zwischen vform- und passive-Spezifikationen wird mittels der FCR (76) hergestellt, die hier als (106) wiederholt wird. Man erzwingt somit den Ausschluß von $2, da ih[r] die Kategorie V[l, pas, -passive] widerspricht (vgl. Abbildung 13).

(106) [pas] => [+passive]

Ein möglicher Nachteil bei dieser Vorgehensweise ist, daß für jedes analoge Phänomen ein neues Merkmal eingeführt werden muß, was die Komplexität der Kategorien in der Grammatik erhöht.

10.3 Die Generierung von Perfekt-Hilfsverben im Deutschen

Der Generierungsalgorithmus geht davon aus, daß die Verweise auf Terminalsymbole in der Ausgangsstruktur nur jeweils einen Eintrag des SFL zugänglich machen. Diese Forderung scheint zunächst unbequem, denn bei der Generierung von Perfektkomplexen im Deutschen spricht vieles dafür, dem Generator nur den Auftrag „Generiere Perfekt!" zu geben, ohne z.B. auf semantischer Ebene zu spezifizieren, mit welchem Wortstamm dies geschehen soll.[67] Dieser Auftrag wird einfach durch einen Verweis in der Ausgangsstruktur auf die Lexikoneinträge für Perfekt-Hilfsverben repräsentiert.

Bei einem SFL liegt es dennoch auf den ersten Blick nahe, für *sein* und *haben* V[l]-Einträge vorzusehen, auf die gemeinsam verwiesen wird. Die MIPs erfüllen die Aufgabe, den Wert des lexikalisch für Verben spezifizierten Merkmals paux vom Hauptverb (z.B. V[5] in *Si* (vgl. Abbildung 19) zur präterminalen Kategorie V[l] zu transportieren. Diese ist dann ausreichend spezifiziert für die eindeutige Entscheidung zwischen *haben* oder sein, denn beim Lexikonzugriff müssen u.a. die paux-Spezifikationen miteinander unifizieren (hierbei wird ausgenutzt, daß die perfektiven Formen von *sein* mit *sein* und die von *haben* mit *haben* gebildet werden; wäre das anders, müßte man ein zusätzliches Merkmal einführen).

Da jedoch das Generierungsverfahren den V[l]-Stamm in die syntaktische Struktur einfügt, *bevor* die MIPs den Transport durchführen können, ist keine fundierte Entscheidung möglich. Wird der „falsche" Stamm genommen, entsteht beim Merkmaltransport ein Konflikt, da die paux-Spezifikationen des Vollverbs und an V[l] nicht übereinstimmen. Dieser Konflikt führt infolge der „liberalen" Definition der HFC nicht zu einer Zurückweisung der Struktur, sondern der Transportversuch wird erfolglos abgebrochen und der falsche V[l]-Stamm generiert. Der skizzierte Ansatz ist daher untauglich für die verwendete Definition der MIPs. Auch eine „strikte" Fassung der HFC führt nicht zu einer effizienten Lösung, da sie eben das Backtracking erzwänge, das vermieden werden sollte.

Die Entscheidung kann stattdessen ebenfalls in die Morphologiekomponente hinein verschoben werden. Im SFL sieht man nur einen V[l]-Eintrag vor, der einen Pseudo-Wortstamm hat und für paux unspezifiziert ist. So kann in der syntaktischen Struktur die paux-Spezifikation des Vollverbs ungehindert an die präterminale Kategorie V[l] gelangen. Der Wortstamm des Perfekt-Hilfsverbs wird erst im Zuge der nachgeschalteten Flexion anhand von paux bestimmt.

Die Wahl des Perfekt-Hilfsverbs verläuft dann völlig analog zur Wahl von Supple-

[67] Daß z.B. Verben der Bewegung je *nach* lokaler oder temporaler Sehweise verschiedene Perfekt-Hilfsverben fordern, spricht nicht dagegen; dies ist ein semantischer Unterschied und stellt ein Wortwahlproblem dar, das im gegebenen Zusammenhang nicht behandelt wird.

tivstämmen, wie sie **z.B.** in der morphologischen Charakterisierung deutscher Verben ohnehin angelegt ist. Die Verlagerung des Problems in die Morphologiekomponente stellt somit eine harmonische Lösung dar.

11 Weitere Implementationen von GPSG

In diesem Abschnitt sollen eine Reihe anderer Ansätze vorgestellt werden. Es gibt nur relativ wenige Implementationen, die einen signifikanten Teil von GPSG rekonstruiert haben. Häufig werden nur einzelne Komponenten der Metasprache in ganz andere theoretische Zusammenhänge integriert. Das ID/LP-Format beispielsweise wird in funktionalen syntaktischen Ausgangsstrukturen zur Generierung von Sätzen verwendet [Emele 1987]. Solche eher eklektischen Ansätze sind nicht das Thema dieser Arbeit.

Es zeigt sich, daß die Implementationen der mittleren GPSG den Formalismus getreuer widerspiegeln als Systeme, die auf der späten GPSG beruhen. Die Abkehr von der linguistischen Vorlage scheint aus .zwei voneinander unabhängigen Motiven heraus zu erfolgen. Zum einen liegt das an der geringen Durchschaubarkeit der Definitionen in GKPS, die es der Regelschreiberin nicht gerade einfach machen, größere Grammatiken zu formulieren. Zum ändern wurden erst Mitte der Achtziger Jahre die Komplexitätsuntersuchungen publiziert, die nachweisen, daß weder die vollständig kompilierte kontextfreie Grammatik (wegen ihrer Größe) noch die Metasprache (wegen ihrer Komplexität) effiziente Verfahren zur Sprachverarbeitung zulassen. Viele Implementatorinnen verfolgen daher die Maxime, „möglichst viel" in einer Vorverarbeitungsphase zu erledigen.

Hierin unterscheiden sich die meisten anderen Ansätze vom Berliner System, in dem bis auf die Metaregel-Expansion alle implementierten Komponenten der Metasprache zur Laufzeit angewendet werden. Das Berliner GPSG-System unterscheidet zwischen Verarbeitungswissen (Kontrollstruktur des Parsers oder Generators) und sprachlichem Wissen; es unterteilt letzteres abermals in generelles und einzelsprachspezifisches Wissen (Formalismus bzw. Grammatik).

Die Entscheidung, welche Modularisierungen in eine Systemkonzeption eingehen sollen, fällt verschieden aus. Soll das ID/LP-Format direkt verarbeitet werden oder eine Menge von PS-Regeln erzeugt werden? Sollen die MIPs getrennt vom Parser operieren oder integriert sein? Ich werde anhand der verschiedenen Ansätze darstellen, wie diese Fragen jeweils beantwortet werden.

Die grundsätzliche Frage lautet: Welche GPSG-Komponenten sollen in einer Vorverarbeitungsphase der Grammatik operieren und welche während der Analyse oder Generierung einer Endkette. Sie wird in Abschnitt 12 ausführlich diskutiert.

In diesem Abschnitt möchte ich eine andere Klassifikation heranziehen, die wenigstens einige grobe Anhaltspunkte bietet, was die jeweils zugrundeliegende Motivation für die Implementation war. Die meisten Implementationen umfassen den Grammatikformalismus, eine Grammatik und einen Bottom-Up-Parser (für Englisch). Häufig kommen eine Entwicklungsumgebung sowie ein Präprozessor für die Grammatik hinzu.

Bei einer Implementation können unterschiedliche Verwendungsweisen des Rech-

ners intendiert sein. Zum einen kann die Überprüfung theoretisch-linguistischer Hypothesen im Vordergrund stehen. Im diesem Fall dient der Computer primär als *Testbett* für Grammatiken oder den Formalismus. Der Parser wird in diesem Zusammenhang als linguistisches Werkzeug zur Grammatikentwicklung oder zur Demonstration oder gar Evaluation der Korrektheit von Formalismen gebraucht. Die Implementation fordert ein erhebliches Maß an Formalisierung und eine ausgewogene Ausarbeitung der einzelnen Teilbereiche. Beispielsweise genügt ein Grammatikfragment wie das in GKPS (S. 245-249) nicht zur Verarbeitung von Relativsätzen infolge seiner mangelhaften Ausarbeitung der NP-Syntax.[68]

Zum ändern ermöglichen Implementationen von GPSG einen wichtigen Teil der notwendigen linguistischen Fundierung natürlichsprachlicher Systeme. Es gilt, eine linguistische Theorie im Hinblick auf praktische Anwendungen zu interpretieren (Boguraev nennt das „linguistic engineering" [Boguraev 1988, S. 126]). Ein Parser stellt dabei eine unter mehreren Systemkomponenten dar und erfüllt als solche eine bestimmte Funktion in einem *Gesamtsystem,* sei es ein Datenbankzugangssystem oder eine linguistische Werkbank. Diese Funktion ist durch die zugrundeliegende Theorie linguistisch fundiert.

Neben der Verwendungsweise des Rechners ist die Verwendung von Techniken zur effizienten Repräsentation und Verarbeitung ein unterscheidendes Kriterium. Die *strikte*[69] Implementation der Metasprache, wie sie z.B. im Berliner System vorliegt, ist nicht zwingend. Bei Verwendung effizienter Softwaretechniken ist interessant, wie sich die Komponenten der Metasprache aus den gewählten Repräsentationen rekonstruieren lassen.

Dieses Kriterium verläuft orthogonal zu dem erstgenannten, so daß sich vier Gruppen ergeben, denen die Systeme zugeordnet werden können.

1. Testbett; strikte Implementation der Metasprache: Beispiele sind [Evans 1985, Evans/Gazdar 1984, Kilbury 1987, Kilbury 1988, Naumann 1988, Phillips/Thompson 1985, Ramsay 1985]. Die Möglichkeiten einer Rekonstruktion von GPSG im PATR-Formalismus analysiert [Shieber 1986]. Diese Arbeit beschrieb als erste eine implizite Anwendungsreihenfolge der metasprachlichen Komponenten von GPSG, die ja als simultan wirkende Constraints über der Menge der lokalen Bäume definiert sind. Der Vorschlag ist zwar klar und einleuchtend begründet, wurde aber nie implementiert.[70]

   Ich werde auf das ProGram-System von Evans eingehen (Abschnitt 11.1), da viele Konzepte daraus in das Berliner GPSG-System eingeflossen sind und den Formalismus von Naumann skizzieren, der Aspekte der mittleren und der späten GPSG in sich vereinigt (Abschnitt 11.2). Die CCRs von Kilbury wurden bereits in Abschnitt 4.4 diskutiert.

[68] Es handelt sich dabei offensichtlich um die Zusammenstellung der verwendeten Beispielregeln und nicht um eine ausgewogene Grammatik. Genau dies zeigt ganz deutlich, wo eine Implementation Lücken schließen muß.

[69] Strikt in dem Sinne, daß die Komponenten der Metasprache die zu implementierenden Objekte darstellen.

[70] Daß im Berliner GPSG-System praktisch dieselbe Anwendungsreihenfolge verwendet wird, dürfte die unabhängig voneinander erarbeiteten Argumente in beiden Ansätzen stützen.

2. **Testbett; Verwendung besonderer Techniken:** Beispiele sind [Fisher 1989, Nakazawa/Neher 1987, Nakazawa *et al.* 1988].

   Die Arbeit von Fisher wird im Zusammenhang mit dem Ansatz des Alvey-Projekts diskutiert (s.u.). In Abschnitt 11.3 stelle ich den Vorschlag von Nakazawa und Neher zur Repräsentation von Merkmalspezifikationen mit Bitvektoren vor.

3. **Gesamtsytem; strikte Implementation der Metasprache:** Ein Beispiel hierfür ist das natürlichsprachliche System von Hewlett Packard [Gawron *et al.* 1982], das auf der mittleren GPSG basiert. Der Berliner GPSG-Formalismus, der eine der Grundlagen für Parsing und Generierung in einem multilingualen MÜ-System darstellt, entstand aus der Testbett-Sichtweise heraus und wurde in ein komplexes Gesamtsystem eingebettet.

4. **Gesamtsystem; Verwendung besonderer Techniken:** Beispiele hierfür sind der GPSG-basierte Formalismus des britischen Alvey-Projektes [Boguraev 1988, Boguraev *et al.* 1988, Briscoe *et al.* 1987, Phillips 1987, Phillips/Thompson 1987], bei dem u.a. auf flexible Regelformate und auf Laufzeiteffizienz beim Parsing geachtet wurde, und das HPSG-basierte Datenbankzugangssystem von Hewlett Packard [Nerbonne/Proudian 1987].

   In Abschnitt 11.4 stelle ich das im Alvey-Projekt verwendete Konzept der Propagierungsregeln vor, mit dem die MIPs beschrieben werden und gehe auf den hierin verwandten Ansatz von Fisher ein.

11.1 Das ProGram-System von Evans

ProGram [Evans 1985, Evans/Gazdar 1984] ist ein interaktives Werkzeug zur Entwicklung und zum Testen von GPS-Grammatiken gemäß der mittleren GPSG-Theorie. Es ist in Prolog implementiert und an der Universität Sussex erhältlich. Zahlreiche Ideen und Konzepte der Implementation sind in die Entwicklung des Berliner GPSG-Systems eingeflossen (vgl. [Keller 1987]).

Die Notwendigkeit, selbst kleinere GPS-Grammatiken mithilfe von Computerwerkzeugen zu entwickeln, wurde bereits in Abschnitt 7 betont. In ProGram ist ein Parser als zentrales diagnostisches Werkzeug integriert, um die weitgehende Korrektheit von Grammatiken zu prüfen und festzustellen, ob den Eingabeketten die intendierten Strukturen zugeordnet werden. Da ProGram keinen Generator umfaßt, kann nicht ohne weiteres geprüft werden, ob eine Grammatik übergeneriert, d.h. allgemein als ungrammatisch bezeichneten Eingabeketten Strukturen zuordnet.

ProGram sieht vor, daß nach der Eingabe der verschiedenen Komponenten (Merkmalsyntax, ID-Regeln, Metaregeln, LP-Regeln, FCRs, FCDs, Aliases, RACs[71] und ein Lexikon) eine Normalisierungs- und eine Vorverarbeitungsphase erfolgt, in der die Grammatik in eine für den Parser geeignete Notation konvertiert wird.

In der Normalisierungsphase wird eine erste Fehlersuche durchgeführt und die Merkmale jeder Kategorie in eine kanonische Ordnung gebracht. Nicht vorhandene

[71] RACs, *Root Admissibthty Condittons*, beschranken Merkmalspezifikationen der Wurzel einer Struktur, mit denen „uninteressante" Ergebnisse des Parsers ausgeblendet werden können.

Spezifikationen werden mithilfe von Prolog-Variablen kodiert. Somit ist die Merkmalsunifikation in GPSG einfach durch Prolog-Unifikation implementiert. Außerdem werden eine Reihe weiterer Formatänderungen bei den anderen Komponenten durchgeführt.

In der Vorverarbeitungsphase werden die Metaregeln angewendet sowie danach die HFC und das CAP ausgeführt und ihre Effekte in die ID-Regeln kompiliert. Durch die Einführung identischer Prolog-Variablen bei gleich zu spezifizierenden Merkmalen wird sichergestellt, daß diese Merkmale denselben Wert erhalten. Alle anderen Komponenten der Theorie und der Grammatik sind in den Parser integriert und werden zur Laufzeit aktiviert.

Im Unterschied zur mittleren GPSG wird in der Vorverarbeitungsphase keine kontextfreie Grammatik erzeugt, sondern die GPS-Grammatik wird direkt zum Parsen verwendet. Dies stellt einen Schritt hin zur direkten Interpretation der GPSG dar.

11.2 Der GPSG-Parser von Naumann

In [Naumann 1988] wird die LISP-Implementation eines Chart-Parsers beschrieben, der zwar nicht indirekt arbeitet (d.h. vermittels einer explizit zugrundeliegenden kontextfreien Grammatik), jedoch auch nicht direkt, wie im Geiste von GKPS:

> Ihm liegt ein Organisationsmodell zugrunde, das zwischen dem der Synopsis [der mittleren GPSG; S.B.] und dem aus GKPS einzuordnen ist: einerseits werden wie in der Synopsis zunächst kontextfreie Regeln erzeugt, mit denen dann der Parser arbeitet; andererseits aber bilden die Instantiierungsprinzipien einen integralen Bestandteil des Parsers, d.h., sie wirken bei der Generierung von Strukturen als spezielle Filter […], die die Erzeugung von [unzulässigen; S.B.] Strukturen ausschließen […]. [Naumann 1988, S. 66]

Aus den Metaregeln, den ID-Regeln und den LP-Aussagen werden kontextfreie Regeln kompiliert. Mit diesen erfolgt das Parsing (bottom-up, Breite-zuerst), wobei die MIPs angewendet werden. Im Anschluß daran instantiiert ein *Post-Filter* weitere Merkmale aufgrund von FCRs und FSDs und filtert Strukturen mit nicht legalen Kategorien aus.

Die Aufgabe des Parsers besteht aus Strukturgenerierung, Merkmalinstantiierung durch direkte Anwendung von FFP, CAP und HFC sowie Aktualisierung von Merkmalspezifikationen in bereits verarbeiteten Kategorien. Der letzte Punkt dient m.E. ausschließlich dazu, die Unifikation von Variablen korrekt nachzuspielen, denn Naumann verzichtet auf die Verwendung von Unifikation als Basisoperation (vgl. [Naumann 1988, S. 80]).

Die MIPs sind als Vorbedingungen für die bottom-up erfolgende Strukturgenerierung definiert, d.h. eine Teilstruktur S_2 wird nur dann in eine Teilstruktur S_1 eingefügt, wenn die MIPs dies in S_1 zulassen. Im Falle der Einfügung müssen ggf. diejenigen Tochterknoten in S_1 gemäß den MIPs aktualisiert werden, die bereits Teilstrukturen dominieren und zusätzlich ggf. weitere Kategorien in diesen Teilstrukturen.

Inhaltlich unterscheiden sich die MIPs ähnlich stark von GKPS wie die des Berliner GPSG-Systems. Das FFP fordert im Unterschied zu GKPS, daß ererbte FOOT-Merkmale an der Mutter auch an einer Tochter ererbt sein müssen (zur Begründung vgl. [Naumann 1988, S. 91ff]). Das CAP wurde ebenfalls in ein rein syntaktisches Prinzip abgeändert, insofern als Controller und Controllee(s) durch entsprechende Merkmalspezifikationen in den ID-Regeln zu kennzeichnen sind. Die HFC orientiert sich an der Definition von [Gazdar/Pullum 1982]. Die MIPs werden in der Reihenfolge FFP-CAP-HFC abgearbeitet; offenbar geschieht dies in Analogie zu einer argumentativ begründbaren Anordnung der entsprechenden Prinzipien in GKPS (vgl. [Naumann 1988, S. 65, Fn. 14]).

Die MIPs sind als integrale Bestandteile des Parsers definiert. Das Parsing erfolgt in zwei Durchläufen, und es können unzulässige Strukturen entstehen, die wieder ausgefiltert werden müssen.

Naumanns Grammatik deckt (im Unterschied zur englischen Grammatik des Berliner Systems) die Beispiele aus GKPS ab und liefert dieselben Analysen wie GKPS.

11.3 Bitvektor-Repräsentationen: Nakazawa und Neher

Ausgehend von der Annahme, daß die Anwendung der metasprachlichen Komponenten zur Laufzeit die Beschränkung der zu generierenden Kategorien und Strukturen ermöglicht, während eine Vorverarbeitung der Grammatik zu einer kombinatorischen Explosion der Anzahl der Kategorien und Strukturen führt, schlagen Nakazawa und Neher eine effiziente Repräsentation der Merkmalspezifikationen und Regelschemata mithilfe von Bitvektoren vor [Nakazawa/Neher 1987, Nakazawa *et al.* 1988]. Sie ist als Transformation aus der üblichen Schreibweise in die interne Darstellung implementiert.

Die wesentliche Idee ist, die Basisoperationen Extension und Unifikation durch Boole'sche Operationen über Bits durchführen zu können. Bestimmte Folgen solcher Operationen erlauben dann die Rekonstruktion der Komponenten der GPSG-Metasprache.

Auf diese Weise wird die Expansion von Regelschemata zur Laufzeit implementiert und in einem modifizierten Earley-Parser für die Analyse englischer und japanischer Sätze eingesetzt. Der zentrale Aspekt der Arbeit ist die Repräsentation mit Bitvektoren, nicht die Vollständigkeit der Metasprache. So wird in diesem Ansatz ohne Metaregeln und ID/LP-Format gearbeitet. Ebenso wurde auf eine Implementation des CAP verzichtet.

Die interne Repräsentation von Merkmalspezifikationen beruht auf einem Bitvektor, dessen Länge für eine gegebene Grammatik konstant ist und von der Anzahl der möglichen Merkmal-Wert-Paare abhängt. Für kategorienwertige Merkmale ist ein einziges Bit vorgesehen, das das Vorhandensein einer kategorienwertigen Spezifikation darstellt.[72] Der Wert wird durch einen eigenen Vektor repräsentiert. Der Vektor in (107) (vgl. [Nakazawa/Neher 1987, S. 116]) ist die „Ausgangsstellung" (er

[72]Die fünfundzwanzig atomwertigen und vier kategorienwertigen Merkmale aus GKPS ergäben eine Vektorlänge von $l \geq 149$.

stellt die in allen Merkmalen Undefinierte Kategorie dar).[73]

(107)

v	n	bar	subcat	past	agr
+ -	+ -	0 1 2	1 2 3	+ -	
1 1	1 1	1 1 1	1 1 1	1 1	1

Partiell spezifizierte Kategorien werden repräsentiert durch „Abschalten" der Bits für die Spezifikationen, die nicht in der Kategorie enthalten sind. Ist für ein kategorienwertiges Merkmal m eine Spezifikation vorhanden, wird das entsprechende Bit abgeschaltet und der Wert von m auf externe Weise mit der Kategorie assoziiert.

Sei *bit(C)* ein Bitvektor der Länge n für eine Kategorie *C'.* Dann können Extension und Unifikation wie folgt definiert werden (für kategorienwertige Merkmale wird ein entsprechender Rekursionsschritt erforderlich):

- $C \sqsubseteq C'$ gdw. $\forall i\ (1 \leq i \leq n) : bit_i(C') = bit_i(C) \vee bit_i(C')$
- $C \sqcup C'$ gdw. $\forall i\ (1 \leq i \leq n) : bit_i(C') = bit_i(C) \wedge bit_i(C')$

Im folgenden skizziere ich, wie HFC, FFP und FCRs mithilfe dieser Operationen implementiert werden. HFC erfolgt nur für instantiierte HEAD-Spezifikationen, vermutlich um die Probleme mit freien Spezifikationen zu umgehen (vgl. Abschnitt 4.8).[74] Mit jeder Kategorie ist ein zweiter Vektor *bit2* assoziiert, der für HFC die ererbten Spezifikationen der Mutter ausschließt (alle instantiierbaren Spezifikationen sind abgeschaltet). Die Menge HEAD wird durch einen weiteren Bitvektor, *head-feature,* dargestellt, in dem die Bits für mögliche HEAD-Spezifikationen abgeschaltet sind. [Nakazawa/Neher 1987] geben folgendes Beispiel, das die Wirkung der HFC der mittleren GPSG reflektiert. Nehmen wir an, in der Regel (108) unifiziere das Head mit der HEAD-Spezifikation [+past]. Gemäß HFC soll sie ebenfalls an der Mutter auftreten. (109) zeigt die Ergebnisse einer Reihe von Operationen, die mit (7) zur Repräsentation von VP[+past] führen. In (5) sind alle irrelevanten Spezifikationen ausgeschlossen, und (6) enthält die instantiierten HEAD-Spezifikationen. Diese werden mit der Mutter unifiziert.

(108) VP —> V NP

[73][Nakazawa *et al.* 1988] sehen jeweils einen zusätzlichen Wert für „Undefiniert" vor.

[74]Dies ist möglich, wenn man diejenigen HEAD-Spezifikationen, die nicht frei sind, in den Regeln angibt. Man verzichtet auf die Metanotation H für Heads; außerdem ist subcat kein HEAD-Merkmal.

(109)

		v	n	bar	subcat	past	agr
		+ −	+ −	0 1 2	1 2 3	+ −	
1	*bit*(V[+past])	1 0	0 1	1 0 0	0 0 0	1 0	1
2	*bit*(VP)	1 0	0 1	0 0 1	0 0 0	1 1	1
3	*bit*2(VP)	1 1	1 1	1 1 1	0 0 0	0 0	0
4	*head-feature*	0 0	0 0	0 0 0	1 1 1	0 0	0
5	3 ∨ 4	1 1	1 1	1 1 1	1 1 1	0 0	0
6	5 ∨ 1	1 1	1 1	1 1 1	1 1 1	1 0	1
7	6 ∧ 2	1 0	0 1	0 0 1	0 0 0	1 0	1

Bei dem FFP werden auf analoge Weise die instantiierten FOOT-Spezifikationen an jeder Tochter berechnet. Die Unifikation *U* dieser Vektoren wird mit der Mutter unifiziert, um die Bits für die an ihr zu instantiierenden FOOT-Merkmale abzuschalten (es wird vereinfachend davon ausgegangen, daß alle FOOT-Merkmale kategorienwertig sind). Ferner wird zu jedem Merkmal aus *U* die den Töchtern assoziierte Kategorie der Mutter assoziiert.

FCRs werden als „illegale" Bitmuster kodiert, die bei erfolgreicher Unifikation mit einer Kategorie zu deren Ausschluß führen.[75] Es wird zunächst eine Termumformung durchgeführt (vgl. (111)-(112)). Dann lassen sich die illegalen Bitvektoren direkt aufschreiben (113).

(110) [+past] ⇒ [+v]

(111) ¬([+past] ∧ ¬[+v])

(112) ¬([+past] ∧ [−v]) ∨ ¬([+past] ∧ [v]) ¬([+past] ∧ [¬v])

(113) ¬(01 11 111 111 10 1) ∨ ¬(00 11 111 111 10 1) ∨ ¬(11 11 111 111 10 1)

11.4 Propagierungsregeln: Phillips und Thompson

In diesem Abschnitt wird (vorwiegend am Beispiel des im Alvey-Projekt verfolgten Ansatzes) eine Rekonstruktion der MIPs anhand eines generelleren Regelformats skizziert.

Einen Überblick über Ziele und Forschungsstand des dreiteiligen britischen Großprojekts gibt [Boguraev 1988]. Das Alvey-Projekt verfolgt das Ziel, eine Reihe von Werkzeugen für Forschung im Bereich der Verarbeitung natürlicher Sprache zu

[75] Eine ähnliche Idee erwies sich auch bei den FCR-Constraints im Berliner GPSG-System als tragfähig (vgl. Abbildung 13).

erstellen. Das schließt einen effizienten Parser ein, der eine große Grammatik des Englischen analysieren kann.

Phillips und Thompson [Phillips 1987, Phillips/Thompson 1987] stellen ebenfalls fest, daß der Formalismus von GKPS nicht für effiziente Implementation geeignet ist. Sie schlagen eine Abbildung der MIPs auf einen einfacheren, „low-level"-Formalismus vor, der maschinelle Sprachverarbeitung unterstützt. Die MIPs und FSDs werden von sogenannten *Propagierungsregeln* (kurz: P-Regeln) subsummiert, die eine wohldefinierte Syntax und Semantik besitzen. FCRs werden nicht verwendet; vereinfacht gesagt, können ihre Effekte mithilfe von P-Regeln nachgespielt werden (vgl. [Phillips/Thompson 1987, S. 130fj). Der Formalismus besteht dann aus vier Komponenten: den ID-Regeln, den LP-Aussagen, den Metaregeln und den P-Regeln.

P-Regeln werden auf ID-Regeln angewendet. Sie dienen dazu, Merkmale an Kategorien zu instantiieren oder zu kospezifizieren, existierende Spezifikationen zu überschreiben sowie Töchter aus ID-Regeln zu entfernen oder hinzuzufügen. Eine P-Regel ist eine Liste von vier bis sechs Elementen:

1. eine Liste von Merkmalnamen (in geschweiften Klammern), die bestimmt, welche Merkmale durch die P-Regel betroffen sind,

2. eine Kategorie, die als *Pattern* der Mutter der ID-Regel fungiert,

3. eine Kategorie, die als *Modifikator* der Mutter der ID-Regel fungiert (optional),

4. eine der Operationen aus {+, -, :} mit folgender Bedeutung:

   „+" Das einzige Argument ist eine Kategorie, die der ID-Regel als zusätzliche Tochter hinzugefügt wird.

   „—" Das einzige Argument ist eine Kategorie. Alle Töchter einer ID-Regel, die Extensionen dieser Kategorie sind, werden aus der Regel entfernt.

   „:" Das erste Argument ist eine Kategorie. Alle Töchter einer ID-Regel, die Extensionen dieser Kategorie sind, werden durch das zweite Argument, ebenfalls eine Kategorie, modifiziert.

5. Weitere Elemente sind Argumente der Operation.

Betrachten wir einige Beispiele (nach [Phillips 1987, S. 11]). Durch (114) wird ein FSD ausgedrückt, indem jede NP-Tochter, die nicht für Kasus spezifiziert ist, den Wert acc erhält (Instantiierung). In (115) wird gefordert, daß in Regeln, deren Mutter eine Extension von [+subj] ist, NP-Töchter nominativen Kasus erhalten (Überschreiben). Eine Andeutung, wie HFC rekonstruiert werden kann, vermittelt (116). Es stellt sicher, daß Spezifikationen für n und v an der Mutter ebenfalls am Head auftreten. Das Präfix „@" bedeutet variable Spezifikationen (z.B. [@n, @v]). Der Buchstabe „F" ist eine interne Variable, an die jeweils der zu verarbeitende Merkmalname gebunden wird. Als Resultat der Anwendung von (116) können ID-Regeln mit kospezifizierten (variablen) Merkmalwerten entstehen.

(114) {} X : NPhcas] [acc]

(115) {} [+subj] : NP [nom]

(116) {n v} @F : H ©F

Einen ähnlichen Ansatz verfolgt [Fisher 1989] mit Propagierungsregeln, die Merkmalspezifikationen entweder von der Mutter zur Tochter transportieren oder umgekehrt. Das Merkmaltransport-Regime legt für jedes Merkmal fest, in welche Richtung(en) es seinen Wert weitergeben darf. Eine Reihe zusätzlicher Bedingungen erlauben die Rekonstruktion *ähnlicher* Prinzipien wie HFC, FFP und CAP. Beispielsweise wird für HFC der Merkmaltransport dahingehend eingeschränkt, daß HEAD-Merkmale nur von der Mutter zum Head transportiert werden dürfen. Wie Fisher selbst betont, ist auf diese Weise eine exakte Rekonstruktion der MIPs aus GKPS nicht möglich (z.B. kann die Durchschnittsbildung von HEAD-Spezifikationen verschiedener Heads nicht modelliert werden oder die Übereinstimmung einer Tochter mit dem kategorialen Merkmalwert einer Schwester aufgrund des CAP). Aber dies wird inkauf genommen, denn Fishers Hauptziel ist ein Parser mit polynomialem Zeitaufwand für linguistisch plausible Grammatiken.

Betrachten wir nun am Beispiel des FFP, wie bei Phillips und Thompson die MIPs durch P-Regeln ausgedrückt werden.

(117) {wh re} ¬F @F : ¬F @F
{slash} ¬F @F : [¬F, ¬subcat] @F

Wenn die Mutter für wh, re oder slash nicht spezifiziert ist, erhält sie dieselben Werte für diese Merkmale wie Töchter, die nicht für sie spezifiziert sind. Für slash wird zusätzlich gefordert, daß nur nichtlexikalische Töchter involviert sind (dies wird in GKPS durch eine FCR sichergestellt). Die Forderung, daß die Kategorien für die angeführten Merkmale nicht spezifiziert sein dürfen, stellt sicher, daß die ererbten FOOT-Spezifikationen vom FFP ausgeschlossen werden.

Das Vorhandensein einer Variablen an mehreren Töchtern impliziert nicht, daß *alle* diese Töchter den gleichen Wert haben müssen. Die Verwendung von Variablen ist hier völlig anders als z.B. in ProGram oder im Berliner System. Dies läßt sich am besten am Beispiel verdeutlichen. Betrachten wir das Merkmal slash. Beim Abgleich einer ID-Regel, auf die (117) angewendet wurde, mit einem lokalen Baum erfolgt keine automatische simultane Instantiierung der Vorkommen von @slash. Der lokale Baum ist auch dann zulässig, wenn keine Tochter einen Wert für slash besitzt. @slash wird dann einfach nicht gebunden. Wenn irgendeine Tochter im lokalen Baum einen instantiierten slash-Wert besitzt, dann wird @slash an ihn gebunden, und die Mutter muß denselben Wert besitzen. Andere Töchter haben denselben slash-Wert oder sind für slash Undefiniert.

Der Formalismus ist so aufgebaut, daß eine Kompilation der Grammatik möglich wird. Zuerst werden Metaregeln auf ID-Regeln angewendet (unter Einhaltung des endlichen Abschlusses) und auf die resultierende Menge von ID-Regeln danach die P-Regeln. Abschließend werden die LP-Aussagen benutzt, um eine Menge von PS-Regeln zu erzeugen.

Eine Unterscheidung von einzelsprachlichem und generellem sprachlichen Wissen ist in diesem Formalismus unnatürlich. Ein Hinweis darauf ist das Fehlen von (einzelsprachlichen) FCRs, deren Effekte von P-Regeln nachgespielt werden. Andererseits werden die MIPs, die zum generellen sprachlichen Wissen gehören, ebenfalls von

P-Regeln dargestellt (teilweise von denselben, wie etwa (117)). Sie sind damit gleichermaßen der Bearbeitung durch die Regelschreiberin zugänglich wie Metaregeln, ID-Regeln, LP-Aussagen und andere P-Regeln.

Aufgrund des Projektziels, einen *Parser* zu erstellen, liegt die Verbindung von generellem sprachlichen Wissen und Kontrollwissen nahe. Infolgedessen ist es nicht verwunderlich, daß der Formalismus nicht deklarativ definiert ist (vgl. [Phillips 1987, S. 24]):

- Top-Down-Verarbeitung ist aufwendiger als Bottom-Up-Verarbeitung. Dies hängt mit dem anhand von (117) diskutierten Abgleich zwischen ID-Regeln und lokalen Bäumen zusammen; ein Argument das im Zusammenhang mit dem FFP bereits in Abschnitt 9 auftrat.

- Die P-Regeln sind extrinsisch geordnet, d.h. es wird ‚von außen" für eine kumulative Wirkung auf ID-Regeln gesorgt.

Phillips hält es für wichtiger, daß der Formalismus wohldefiniert, einfach handhabbar und effizient sei.

Selbst wenn diese Forderungen mit dem geschilderten Ansatz eingelöst werden, erscheint es aufgrund der geringen Modularisierung fraglich, ob er zur Beschreibung anderer Sprachen als des Englischen geeignet ist und ob P-Regeln formulierbar sind, mit denen andere Kontrollstrukturen effizient eingesetzt werden können.

12 Indirekte und direkte Interpretationen von GPSG

Zu jeder Phase der GPSG-Entwicklung hat man sich Gedanken gemacht, wieviel Vorverarbeitung erfolgen soll. Das Dilemma stellt sich heute so dar (vgl. Abschnitt 5): Für eine direkte Interpretation der GPSG gibt es keine prinzipiell effizienten Algorithmen und bei einer indirekten Interpretation entsteht eine extrem große Anzahl von Regehi, die als Faktor in die Verarbeitungszeit eingeht.

Die meisten Ansätze (so etwa alle in dieser Arbeit beschriebenen Implementationen) stellen sich diesem Dilemma, denn es gibt einige gute Gründe für die Existenz von in der Praxis effizient arbeitenden Algorithmen. Die komplexitätstheoretischen Ergebnisse zeigen, wo ein Hebel angesetzt werden kann: Alle Verfahren kennen irgendein Konzept von variablen Merkmalwerten, das es gestattet, Kategorien zusammenzufassen und Werte in verschiedenen Kategorien zu kospezifizieren. Dies ermöglicht—im Falle einer direkten Interpretation—die Vermeidung der kombinatorische Explosion der Kategorienzahl und—im Falle einer indirekten Interpretation—die Beschränkung der Anzahl der PS-Regeln.

Bevor ich in Abschnitt 12.2 die unterschiedlichen Entscheidungen, die den Implementationen zugrundeliegen, zusammenfasse und diskutiere, möchte ich auf [Evans 1987] eingehen, wo das genannte Dilemma vermieden wird.

12.1 Die stark direkte Interpretation von GPSG

Evans untersucht in [Evans 1987] die Beziehung zwischen GPSG als formaler Theorie der Syntax natürlicher Sprachen und der Anwendung von GPSG zum Parsing natürlichsprachlicher Endketten. Sein Interesse liegt an der wechselseitigen Beeinflussung von den im Formalismus ausgedrückten linguistischen Generalisierungen und den Möglichkeiten, sie in einer Implementation auszudrücken.

Gewöhnlich sehen Computerlinguisten GPSG als eine abstrakte Beschreibung von zugrundeliegenden kontextfreien Grammatiken an. Nur wenig wurde auf die algorithmischen Eigenschaften der in GPSG ausgedrückten Generalisierungen geachtet. Umgekehrt beruhen Implementationen auf einem (wenn auch implizit) zugrundeliegenden, kontextfreien Formalismus und spiegeln somit wenig von GPSG wider.

Evans unternimmt Schritte, um den algorithmischen Ansatz vom kontextfreien Substratum zu befreien und GPSG *direkt* zu interpretieren als einen Formalismus, der natürlichsprachliche Syntaxen beschreibt. Er unterscheidet den indirekten Ansatz (mit Verwendung einer expliziten kontextfreien Grammatik, wie in der mittleren GPSG vorgeschlagen) vom direkten Ansatz, den er wiederum unterteilt in eine *schwach direkte* Interpretation, bei der eine kontextfreie Grammatik implizit eine Rolle spielt und eine *stark direkte* Interpretation ohne kontextfreies Substratum. GKPS ist in diesem Sinne als schwach direkt zu bezeichnen, denn der zentrale Begriff des zulässigen lokalen Baums entspricht dem der PS-Regel. Wohl sämtliche Implementationen von GPSG sind entweder indirekt oder schwach direkt konzipiert.

Evans beschreibt anhand dreier Beispiele (ID/LP-Format, ID-Regeln und X-Bar-Theorie, MIPs) bisher unentdeckte Möglichkeiten, von der Beschränkung auf lokale Bäume abzugehen und die Komponenten der GPSG-Theorie algorithmisch stark direkt zu nutzen. Auf diese Weise wird die algorithmische Verwendung von GPSG auf der Ebene der Theorie selbst diskutierbar. Eine Reihe von algorithmisch motivierten Änderungen am Formalismus lassen sich damit unmittelbar auch theoretisch begründen.

Wo man das ID/LP-Format verwendet, wird es schwach direkt interpretiert, denn LP-Aussagen werden nur im Kontext von ID-Regeln oder deren Projektionen ausgewertet. Evans schlägt ein Konzept von *sisterhood* vor, das es erlaubt, im Verlauf des Parsings beliebige benachbarte Knoten in einer Chart durch die LP-Aussagen zu überprüfen. Damit wird die in GPSG angelegte, eigenständige Bedeutung der LP-Aussagen in der Theorie algorithmisch (stark) direkt umgesetzt.

Das zweite Beispiel befaßt sich mit ID-Regeln. Im Verlauf der Entwicklung von GPSG wurde immer mehr Information aus PS-Regeln extrahiert und in generelle Prinzipien kodiert. Das, was letztlich übrig blieb, sind die ID-Regeln. Man sollte sie nicht als eigens *konzipierte* Komponente der Metasprache betrachten, denn ihre Bedeutung kann nur im Zusammenhang mit dem gesamten Formalismus beschrieben werden. Evans versucht, den Anteil der ID-Regeln an GPSG-Strukturen durch *Rollen* zu beschreiben, die die ID-Regeln spielen. Er unterscheidet (anhand der impliziten X-Bar-Theorie) verschiedene Regeltypen: *Entwicklungsregeln* haben ein einziges Head mit geringerem bar-Wert als die Mutter; mit ihnen werden Phrasen auf der Basis des Heads aufgebaut. *Ergänzungsregeln* haben ein einziges Head mit demselben

bar-Wert wie die Mutter; sie unterbrechen die Arbeit der Entwicklungsregeln, um Modifikatoren o.a. hinzuzufügen. In *Koordinationsregeln* sind alle Töchter Heads. Anhand solcher Regeltypen kann sich die Verarbeitungsstrategie des Parsers orientieren. Entwicklungsregeln können bottom-up und bidirektional verarbeitet werden, wobei die Regel durch die Analyse des Heads getriggert wird. Ähnlich verhält es sich mit Erweiterungsregeln, die jedoch durch den Modifikator getriggert werden.

Da^ 'ritte Beispiel behandelt die MIPs. Insbesondere die nichtlokalen Abhängigkeiten, die durch HFC und FFP lokal (und damit schwach direkt) beschrieben werden, versucht Evans unabhängig von lokalen Bäumen (und damit stark direkt) zu rekonstruieren. Er definiert dazu *reguläre Merkmale* als solche, deren Spezifikationen durch eine Grammatik nur an bestimmten Stellen der Struktur beeinflußt werden können; an allen anderen Stellen bestimmen die MIPs die Spezifikation. Eine Ausprägung für das FFP bildet Abbildung 4: die Grammatik spezifiziert slash nur an Füller und Lücke, nicht aber in der Mitte; dort ist das FFP zuständig. Evans' stark direkte Interpretation des FFP im Parsing setzt voraus, daß slash regulär ist. Die Verarbeitung in einem Bottom-Up-Parser könnte so aussehen: Wenn der Parser eine Lücke einführen soll, „merkt er sich" die Kategorie (in einer zweiten Agenda) und erzeugt erst dann entsprechende Kanten in der Chart, wenn er auf einen entsprechenden Füller trifft. Die Regularität von slash erübrigt die „Buchführung" in jedem lokalen Baum des mittleren Bereichs.

Diese Vorschläge zu einer stark direkten Interpretation stellen einen ersten Schritt auf dem Weg zu einer neuartigen Implementation eines signifikanten Teils von GPSG dar, mehr nicht. Dies wird von Evans ausdrücklich betont:

> In fact, it is not my intention to put up this formalism äs a complete entity at all. It is simply the sum total of the various disjoint proposals - I have discussed above. It constitutes a reworking of neither the whole of GPSG, nor any integrated consistent fragment of it. [Evans 1987, S. 125]

12.2 Indirekte und schwach direkte Interpretationen von GPSG

Die verschiedenen Versionen des GPSG-Formalismus unterscheiden sich u.a. im Hinblick auf den Stellenwert einer zugrundeliegenden kontextfreien Grammatik. Im allgemeinen kann man die Grammatik im Eingabeformat (z.B. ID-Regeln, LP-Aussagen, Metaregeln, FCRs, FSDs) als *Quellgrammatik* bezeichnen. Zur Sprachverarbeitung wird eine *Zielgrammatik* benutzt, die aus einem Vorverarbeitungsschritt resultiert. Die Teile der Quellgrammatik, die nach der Vorverarbeitung nicht in die Zielgrammatik hineinkompiliert sind, werden zur Laufzeit (d.h. während des Parsings bzw. der Generierung) verwendet. Für die betrachteten Systeme ist die Aufteilung sehr unterschiedlich, wie aus der Abbildung 20 hervorgeht.[76]

Metaregeln werden generell in der Vorverarbeitungsphase angewendet.

[76]Man erinnere sich, daß alle Systeme einen Parser für Englisch enthalten; das von Nakazawa und Neher analysiert auch Japanisch, und das Berliner GPSG-System analysiert *und generiert* sowohl Englisch als auch Deutsch.

Autorinnen	siehe	Quellgrammatik	Vorverarbeitung	Zielgrammatik	zur Laufzeit
ProGram: Evans, Gazdar	11.1	KK, ID/LP, Meta, FCR, FCD, RAC	Meta, HFC, CAP	KK, ID/LP mit Kospezifikation	FFP, ID/LP, FCD, FCR, RAC
Naumann	11.2	KK, ID/LP, Meta, FCR, FSD	Meta, ID/LP	KK, kfG mit Kospezifikation	a) FFP, CAP, HFC b) FCR, FSD
Nakazawa, Neher, Hinrichs	11.3	KK, kfG	Merkmalvektor, Kategorienvektoren	Bitvektoren, kfG	FFP, HFC, FCR, FSD
Alvey: Phillips, Thompson	11.4	KK, ID/LP, Meta, P-Regeln	Meta, ID/LP, P-Regeln	KK, kf Schemata	—
Berliner GPSG-System	6, 7	KK, ID/LP, Meta, FCR	Meta	KK, ID/LP	FFP, AP, HFC, ID/LP, FCR

Abkürzungen: KK := komplexe Kategorien ID/LP := ID-Regeln und LP-Aussagen
Meta := Metaregeln kfG := kontextfreie Grammatik

Abbildung 20: GPSG-Systeme im Überblick

Das FFP hingegen kann nicht ohne weiteres in den Regeln kodiert werden, denn es steht erst zur Laufzeit fest, welche Töchter FOOT-Spezifikationen tragen. In der Vorverarbeitung müßten alle Möglichkeiten in Betracht gezogen werden und entsprechend viele Regeln erzeugt werden. Daher wird das FFP gewöhnlich zur Laufzeit angewendet. Nur das Alvey-System bildet eine Ausnahme infolge seiner speziellen Behandlung von Variablen.

HFC kann dann in der Vorverarbeitung operieren, wenn sie im Sinne der mittleren GPSG definiert ist und nicht als Default-Prinzip wirkt. Im letzteren Fall hängt ihr Effekt von der Wirkung anderer Prinzipien ab, die aus unabhängigen Gründen zur Laufzeit angewendet werden können (FFP).

Die Kompilation des ID/LP-Formats im Rahmen einer Vorverarbeitung wie bei Naumann und im Alvey-System legen die zusätzliche Annahme über LP-Aussagen nahe, daß sie keine instantiierten Spezifikationen einbeziehen. Die Autoren geben jeweils an, daß dies keine nennenswerte Beschränkung für die Analyse des Englischen bedeute. Für das Deutsche sieht dies jedoch anders aus. Im Berliner GPSG-System etwa wird die Stellung des finiten Verbs in Abhängigkeit von der Satzart durch LP-Aussagen bestimmt, die auf Spezifikationen von vform und mc zugreifen ((95) und (96)). Beide Merkmale sind nicht in allen Kategorien ererbt. Das bedeutet zunächst noch nicht, daß das ID/LP-Format nicht im Verlauf einer Vorverarbeitung aufgelöst werden könne. Dies ist sehr wohl berechenbar, indem vor Anwendung der LP-Aussagen eine Menge zusätzlicher ID-Regeln generiert wird, in denen jeweils unterschiedliche Spezifikationen der betroffenen Merkmale vorkommen. Diese wären dann ererbt und könnten von LP-Aussagen erfaßt werden.

Zwei Argumente sprechen gegen eine solche Vorgehensweise. Zum einen könnten

dabei PS-Regeln erzeugt werden, die niemals zulässige Bäume projizieren, da die nachträglich eingeführten Spezifikationen gegen die HFC verstoßen. Zum ändern vervielfacht sich aufgrund jeder betroffenen LP-Aussage die Größe der Grammatik, so daß der Vorteil, keine LP-Aussagen zur Laufzeit behandeln zu müssen, bei weitem durch den Nachteil einer großen Regelmenge überwogen wird.

Die vorangegangenen Überlegungen zeigen, das effizienzsteigernde Verlagerungen von Operationen in eine Vorverarbeitungsphase meistens einen Preis haben, der in Abstrichen an der linguistischen Ausdruckskraft der Grammatik zu bezahlen ist. Ob diese Abstriche inkauf genommen werden können, hängt von der intendierten Verwendung des Systems ab. Die meisten der hier betrachteten Systeme sind erklärtermaßen nur für die Analyse des Englischen ausgelegt. Mit den daraufhin inkauf genommenen Abstrichen werden allerdings die interessanten Eigenschaften von GPSG nicht ausgenutzt, die sich für die im Berliner GPSG-System gewählte Modularisierung als entscheidend erwiesen: die Eignung des Formalismus für die Verarbeitung verschiedener Grammatiken und seine Neutralität bezüglich der Kontrollstrategie (Parsing oder Generierung).

Literatur

[Aho 1968] Alfred V. Aho (1968), 'Indexed Grammars', in *Journal of the Association of Computing Machinery* 15, 647-671.

[Bach/Partee 1980] Emmon Bach und Barbara Partee (1980), 'Anaphora and Semantic Structure', in J. Kreiman und A. E. Ojeda (Hg.), *Papers from the Parasession on Pronouns and Anaphora,* Chicago Linguistics Society, 1980, 1-28.

[Barton *et al.* 1987] G. Edward Barton, Robert C. Berwick und Eric Sven Ristad (1987), *Computational Complexity and Natural Language,* Cambridge, London: MIT Press.

[Bittkau *et al.* 1987] Oliver Bittkau, Christian Haider und Jorg-Uwe Kietz (1987), *KIT-CORE PROLOG Description (Version 1.0),* Technische Universitat Berlin, KIT Interner Arbeitsbericht Nr. 17.

[Boguraev 1988] Branimir Boguraev (1988), 'A Natural Language Toolkit: Reconciling Theory with Practice', in [Reyle/Rohrer 1988], 95-130.

[Boguraev *et al.* 1988] Bran Boguraev, John Carroll, Ted Briscoe und Claire Grover (1988), 'Software Support for Practical Grammar Development', in *Proc. 12. COLING-88,* Budapest, 54-58.

[Brachman/Schmolze 1985] Ronald J. Brachman und Jaime G. Schmolze (1985), 'An Overview of the KL-ONE Knowledge Representation System', in *Cognitive Science* 9, Nr. 2, 171-216.

[Briscoe *et al.* 1987] Ted Briscoe, Claire Grover, Bran Boguraev und John Carroll (1987), 'A Formalism and Environment for the Development of a Large Grammar of English', in *Proc. 10. IJCAI-87,* Mailand, 703-708.

[Busemann 1988a] Stephan Busemann (1988), 'Surface Transformations During the Generation of Written German Sentences', in D. D. McDonald und L. Bole (Hg.), *Natural Language Generation Systems,* Berlin, New York: Springer (Symbolic Computation), 898-165.

[Busemann 1988b] Stephan Busemann (1988), 'Zum Lexikonzugriff bei der Generierung mit GPSG', in H. Trost (Hg.), *Proc. 4- Osterreichische Artificial-Intelligence-Tagung. Wiener Workshop Wissensbasierte Sprachverarbeitung,* Berlin, New York: Springer (IFB Bd. 176), 1988, 164-170.

[Busemann 1990] Stephan Busemann (1990), 'Generierung mit Generalisierten Phrasenstruktur-Grammatiken', Dissertation, Universitat des Saarlandes.

[Busemann/Hauenschild 1988] Stephan Busemann und Christa Hauenschild (1988), 'A Constructive View of GPSG or How to Make it Work', in *Proc. 12. COLING-88,* Budapest, 77-82. , ,

[BuCmann 1983] Hadumod Bufimann (1983), 'Lexikon der Sprachwissenschaft', Stuttgart: Kroner.

[Chomsky 1970] Noam Chomsky (1970), 'Remarks on Nominalization', in R. A. Jacobs und P. S. Rosenbaum (Hg.), *Readings in English Transformational Grammar,* Waltham, MA.: Ginn, 184-221.

[Chomsky/Halle 1968] Noam Chomsky und Morris Halle (1968), *The Sound Pattern of English,* New York: Harper & Row.

[Colmerauer 1982] A. Colmerauer (1982), *Prolog II Reference Manual and Theoretical Model,* Groupe d'Intelligence Artificielle, Universite de Marseille, ERA CNRS 363.

[Emele 1987] Martin Emele (1987), 'FREGE-Ein objektorientierter FRont-End-GEnerator', in K. Morik (Hg.), *GWAI-87. 11th German Workshop on Artificial Intelligence,* Berlin, New York: Springer (IFB Bd. 152), 1987, 64-73.

[Engdahl 1980] Elisabet Engdahl (1980), *The Syntax and Semantics of Questions in Swedish,* Ph.D. Dissertation, University of Massachusetts, Amherst, MA.

[Evans 1985] Roger Evans (1985), 'Program—a Development Tool for GPSG Grammars', in *Linguistics* 23, 213-243.

[Evans 1987] Roger Evans (1987), *Theoretical and Computational Interpretations of Generalised Phrase Structure Grammar,* University of Sussex, Cognitive Science Research Paper CSRP 085.

[Evans/Gazdar 1984] Roger Evans und Gerald Gazdar (1984), *The Program Manual,* University of Sussex, Cognitive Science Research Paper CSRP 035.

[Fisher 1989] Anthony J. Fisher, 'Practical Parsing of Generalized Phrase Structure Grammars', in *Computational Linguistics* 15, Nr. 3, 139-148.

[Gawron *et al.* 1982] J. Maxk Gawron, Jonathan King, John Lamping, Egon Loebner, E. Anne Paulson, Geoffrey K. Pullum, Ivan A. Sag und Thomas Wasow (1982), 'Processing English with a Generalized Phrase Structure Grammar', in *Proc. Conf. of the 20th Meeting of the ACL,* Toronto, 74-81.

[Gazdar 1980] Gerald Gazdar (1980), 'A Phrase Structure Syntax for Comparative Clauses', in T. Hoekstra, H. v. d. Hulst, und M. Moortgat (Hg.), *Lexical Grammar,* Dordrecht: Foris, 1980, 165-179.

[Gazdar 1981a] Gerald Gazdar (1981), 'On Syntactic Categories', in *Philosophical Transactions (Series B) of the Royal Society,* 295, 267-283.

[Gazdar 1981b] Gerald Gazdar (1981), 'Unbounded Dependencies and Coordinate Structure', in *Linguistic Inquiry* 12, 155-184.

[Gazdar 1982] Gerald Gazdar (1982), 'Phrase Structure Grammar', in P. Jacobson und G. K. Pullum (Hg.), *The Nature of Syntactic Representation,* Dordrecht: Reidel, 131-186.

[Gazdar 1985] Gerald Gazdar (1985), *Applicability of Indexed Grammars to Natural Languages,* in [Reyle/Rohrer 1988], 69-94.

[Gazdar/Pullum 1982] Gerald Gazdar und Geoffrey K. Pullum (1982), *Generalized Phrase Structure Grammar: A Theoretical Synopsis,* Indiana University Linguistics Club, Bloomington.

[Gazdar/Pullum 1985] Gerald Gazdar und Geoffrey K. Pullum (1985), 'Computationally Relevant Properties of Natural Languages and Their Grammars', in *New Generation Computing* 3, Nr. 3, 273-306.

[Gazdar/Sag 1981] Gerald Gazdar und Ivan Sag (1981), 'Passive and Reflexives in Phrase Structure Grammar', in J. Groenendijk, T. Janssen und M. Stokhof (Hg.), *Formal Methods in the Study of Language,* Mathematical Centre Tracts, Amsterdam, 1981, 131-152.

[Gazdar *et al.* 1982] Gerald Gazdar, Geoffrey Pullum und Ivan Sag, (1982), 'Auxiliaries and Related Phenomena in a Restricted Theory of Grammar', in *Language* 58, 591-638.

[Gazdar *et al.* 1985] Gerald Gazdar, Ewan Klein, Geoffrey Pullum und Ivan Sag, (1985), *Generalized Phrase Structure Grammar,* Oxford: Blackwell.

[Gazdar *et al.* 1986] Gerald Gazdar, Geoffrey Pullum, Robert Carpenter, Ewan Klein, Thomas Hukari und Robert Levine, *Category Structures,* University of Sussex, Cognitive Science Research Paper CSRP 071.

[Gunji 1987] Takao Gunji (1987), *Japanese Phrase Structure Grammar. A Unification-Based Approach,* Dordrecht: Reidel.

[Haddock *et al.* 1987] Nicholas Haddock, Ewan Klein und Glyn Morrill (Hg.), *Working Papers in Cognitive Science, Volume I. Categorial Grammar, Unification Grammar and Parsing,* Centre for Cognitive Science, University of Edinburgh.

[Hasida 1986] Koiti Hasida (1986), 'Conditioned Unification for Natural Language Processing', in *Proc. 11. COLING-86,* Bonn, 85-87.

[Hauenschild 1988] Christa Hauenschild (1988), 'GPSG and German Word Order', in [Reyle/Rohrer 1988], 411-431.

[Hauenschild/Busemann 1988] Christa Hauenschild und Stephan Busemann (1988), 'A Constructive Version of GPSG for Machine Translation', in E. Steiner, P. Schmidt und C. Zelinsky-Wibbelt (Hg.), *From Syntax to Semantics—Insights From Machine Translation,* London: Frances Pinter, 1988, 216-238.

[Hauenschild/Umbach 1988] Christa Hauenschild und Carla Umbach (1988), 'Funktor-Argument-Struktur. Die satzsemantische Representations- und Transferebene im Projekt KIT-FAST', in J. Schiitz (Hg.), *Workshop 'Semantik und Transfer',* IAI Working Papers Nr. 6, Saarbrücken, 16-35.

[Hendriks 1986] Herman Hendriks (1986), *Foundations of GPSG Syntax,* Doctoraalscriptie Wijsbegeerte, Universitat Amsterdam.

[Hukari/Levine 1986] Thomas E. Hukari und Robert D. Levine (1986), 'Generalized Phrase Structure Grammar: A Review Article', in *Linguistic Analysis* 16, Nr. 3-4, 135-260.

[Jackendoff 1977] Ray S. Jackendoff (1977), *X-bar Syntax. A Study of Phrase Structure,* Cambridge, MA.: MIT Press.

[Jacobson 1987] Pauline Jacobson (1987), 'Review of Gazdar *et al.* 1985', in *Linguistics and Philosophy* 10, 389-426.

[Joshi *et al.* 1975] Aravind K. Joshi, Leon S. Levy, and M. Takahashi (1975), 'Tree Adjunct Grammars', *Journal of the Computer and System Sciences* 10, Nr. 1, 136-163.

[Joshi 1985] Aravind K. Joshi (1985), 'Tree Adjoining Grammars: How Much Context-Senitivity is Required to Provide Reasonable Structural Descriptions?', in D. Dowty, L. Karttunen und A. Zwicky (Hg.) *Natural Language Parsing—Psychological, Computational and Theoretical Perspectives,* Cambridge, MA.: Cambridge University Press, 206-250.

[Joshi 1986] Aravind K. Joshi (1986), 'The Convergence of Mildly Contextsensitive Grammar Formalisms', in T. Wasow und P. Sells (Hg.), *The Processing of Linguistic Structures,* Cambridge, MA.: MIT-Press.

[Kaplan/Bresnan 1982] Ronald Kaplan und Joan Bresnan (1982), 'Lexical Functional Grammar—A Formal System for Grammatical Representation', in J. Bresnan (Hg.), *The Mental Representation of Grammatical Relations,* Cambridge, MA.: MIT Press, 173-281.

[Kay 1979] Martin Kay (1979), 'Functional Grammar', in *Proc. 5th Annual Meeting of the Berkeley Linguistics Society,* 142-158.

[Keenan 1974] Edward A. Keenan (1974), 'The Functional Principle: Generalizing the Notion of *subject of* in N. W. La Galy, R. A. Fox und A. Bruck (Hg.), *Papers of the Tenth Regional Meeting of the Chicago Linguistics Society,* 1974, 298-309.

[Keller 1987] William R. Keller (1987), *An Overview of the Project NASEV Parser,* University of Sussex, Cognitive Science Research Paper CSRP-86.

[Kilbury 1986] James Kilbury (1986), 'Category Co-occurrence Restrictions and the Elimination of Metarules', in *Proc. 11. COLING-86,* Bonn, 50-55.

[Kilbury 1987] James Kilbury (1987), 'A Proposal for Modifications in the Formalism of GPSG', in *Proc. 3rd Conf. of the European Chapter of the ACL,* Kopenhagen, 156-159.

[Kilbury 1988] James Kilbury (1988), 'Parsing with Category Co-occurrence Restrictions', in *Proc. 12. COLING-88,* Budapest, 324-327.

[Kindermann/Quantz 1987] Carsten Kindermann und Joachim Quantz (1987), *Ein Editor mit integriertem Prdprozessor fur das Berliner GPSG-System,* Technische Universitat Berlin, KIT Interner Arbeitsbericht Nr. 18.

[Lenerz 1977] Jurgen Lenerz (1977), *Zur Abfolge nominaler Satzglieder im Deutschen,* Tübingen: Narr.

[Nakazawa/Neher 1987] Tsuneko Nakazawa und Laura Neher (1987), 'Rule Expansion on the Fly—a GPSG Parser for Japanese/English Using a Bit Vector Representation of Features and Rule Schemas', in *Studies in Linguistic Sciences* 17, Nr. 2, Dept. of Linguistics, University of Illinois, Urbana-Champaign, Illinois, 115-124.

[Nakazawa *et al.* 1988] Tsuneko Nakazawa, Laura Neher und Erhard W. Hinrichs (1988), 'Unification with Disjunctive and Negative Values for GPSG Grammars', in *Proc. 8. ECAI-88,* Munchen, 467-472.

[Naumann 1988] Sven Naumann (1988), *Generalisierte Phrasenstrukturgrammatik: Parsingstrategien, Regelorganisation und Unifikation,* Niemeyer: Tubingen.

[Nebel/Smolka 1989] Bernhard Nebel und Gert Smolka (1989), *Representation and Reasoning with Attributive Descriptions,* IBM Deutschland, Institut fur wissensbasierte Systeme, Stuttgart, IWBS Report Nr. 81.

[Nerbonne 1982] John A. Nerbonne (1982), "Phantoms' and German Fronting: Poltergeist constituents?', in *Linguistics* 24, 857-870.

[Nerbonne/Proudian 1987] John Nerbonne und Derek Proudian (1987), *The HP-NL System,* Hewlett-Packard Laboratories, Technical Report.

[Peters/Uszkoreit 1982] Stanley Peters und Hans Uszkoreit (1982), 'Essential Variables in Metarules', Vortrag auf der Tagung *Annual Meeting of the Linguistic Society of America,* San Diego, CA.

[Phillips 1987] John D. Phillips (1987), *A Computational Representation for Generalised Phrase-Structure Grammars,* Ms.

[Phillips/Thompson 1985] John D. Phillips und Henry S. Thompson (1985), 'GPSGP—a Parser for Generalized Phrase Structure Grammars', in *Linguistics* 23, Nr. 2, 245-261.

[Phillips/Thompson 1987] John D. Phillips und Henry S. Thompson (1987), 'A Parser for Generalised Phrase Structure Grammars', in [Haddock *et al.* 1987], 115-136.

[Pollard 1984] Carl J. Pollard (1984), *Generalized Phrase Structure Grammars, Head Grammars, and Natural Languages,* Ph.D. Thesis, Stanford University, Stanford CA.

[Pollard/Sag 1983] Carl Pollard und Ivan A. Sag (1983), 'Reflexives and Reciprocals in English: An Alternative to the Binding Theory', in M. Barlow, D. Flickinger und M. Wescoat (Hg.), *Proc. 2nd West Coast Conference on Formal Linguistics,* Stanford Linguistics Department, Stanford CA., 189-203.

[Pollard/Sag 1987] Carl J. Pollard und Ivan A. Sag (1987), *Information-Based Syntax and Semantics. Volume I,* Center for the Study of Language and Information, CSLI Lecture Notes Nr. 13, Chicago: University of Chicago Press.

[Preufi 1987] Susanne Preufi (1987), *GPSG-Syntax fur tin Fragment des Deutschen,* Technische Universitat Berlin, KIT Interner Arbeitsbericht Nr. 20.

[Preufi 1989] Susanne Preufi (1989), *Koordination und Kongruenz in einer Verallgemeinerten Phrasenstrukturgrammatik,* Magisterarbeit, Technische Universitat Berlin.

[Pullum/Gazdar 1982] Geoffrey K. Pullum und Gerald Gazdar (1982), 'Natural-Languages and Context-Free Languages', in *Linguistics and Philosophy* 4, 471-504.

[Ramsay 1985] Allan Ramsay (1985), 'Effective Parsing with Generalised Phrase Structure Grammar', in *Proc. 2nd Conf. of the European Chapter of the ACL,* Genf, 57-61.

[Reyle/Rohrer 1988] Uwe Reyle und Christian Rohrer (Hg.), *Natural Language Parsing and Linguistic Theory,* Dordrecht: Reidel, 1988.

[Ristad 1986] Eric Sven Ristad (1986), 'Computational Complexity of Current GPSG Theory', in *Proc. Conf. of the 24th Annual Meeting of the ACL,* New York, 30-39.

[Ritchie 1987] Graeme Richie (1987), 'The Computational Complexity of GPSG Parsing', in [Haddock *et al.* 1987], 137-142.

[Russell 1985] Graham Russell (1985), 'A GPS-Grammar for German Word Order', in U. Klenk (Hg.), *Kontextfreie Syntaxen und verwandte Systeme. Vortrdge eines Kolloquiums in Ventron (Vogesen) im Oktober 1984,* Tubingen: Niemeyer, 1985, 19-32.

[Sag 1986] Ivan A. Sag (1986), *Grammatical Hierarchy and Linear Precedence,* Center for the Study of Language and Information, Stanford CA, Report Nr. CSLI-86-60.

[Schachtl 1988] Stefanie Schachtl (1988), 'The Problem of Overgeneration in Parsing Processes and the Aid of Linguistic Generalizations', in H. Trost (Hg.), *Proc. 4- Osterreichische Artificial-Intelligence-Tagung. Wiener Workshop Wissensbasierte Sprachverarbeitung,* Berlin, New York: Springer (IFB Bd. 176), 33-42.

[Shieber 1984] Stuart M. Shieber (1984), 'Direct Parsing of ID/LP Grammars', in *Linguistics and Philosophy* 7, 135-154.

[Shieber 1985] Stuart M. Shieber (1985), 'Evidence Against the Context-Freeness of Natural Language', in *Linguistics and Philosophy* 8.

[Shieber 1986] Stuart M. Shieber (1986), 'A Simple Reconstruction of GPSG' in *Proc. 11. COLING-86,* Bonn, 211-215.

[Shieber 1988] Stuart M. Shieber (1988), 'Separating Linguistic Analyses Froms Linguistic Theories', in [Reyle/Rohrer 1988], 33-68.

[Shieber *et al.* 1983] Stuart M. Shieber, Hans Uszkoreit, Fernando C. N. Pereira, Jane J. Robinson und Mabry Tyson (1983), 'The Formalism and Implementation of PATR-II', in B. J. Grosz und M. E. Stickel (Hg.), *Research on Interactive Acquisition and Use of Knowledge. Final Report,* SRI International, 1983, 39-79.

[Thompson 1981] Henry S. Thompson (1981), 'Handling Metarules in a Parser for GPSG', in M. Barlow, D. Flickinger und I. A. Sag (Hg.), *Developments in Generalised Phrase Structure Grammar: Stanford Working Papers in Grammatical Theory, Volume 2,* Indiana Linguistics Club, Bloomington, 1981,26-37.

[Uszkoreit 1984] Hans Uszkoreit (1984), *Word Order and Constituent Structure in German,* Ph.D. Dissertation, University of Texas, Austin.

[Uszkoreit 1986a] Hans Uszkoreit (1986), 'Constraints on Order', in *Linguistics* 24, 883-906.

[Uszkoreit 1986b] Hans Uszkoreit (1986), 'Syntaktische und semantische Generalisierungen im strukturierten Lexikon', in C.-R. Rollinger und W. Horn (Hg.), *Proc. GWAI-86 und 2. Osterreichische Artificial-Intelligence-Tagung*, Berlin, New York: Springer (IFB Bd. 124), 1986, 87-100.

[Uszkoreit 1986c] Hans Uszkoreit (1986), 'Categorial Unification Grammars', in *Proc. 11. COLING-86*, Bonn, 187-194.

[Volk 1988] Martin Volk (1988), *Parsing German with GPSG: The Problem of Separable-Prefix Verbs*, University of Georgia, Athens GA, Advanced Computational Methods Center, ACMC Research Report 01-0026.

[Weisweber 1987] Wilhelm Weisweber (1987), *Bin Dominanz-Chart-Parser fur generalisierte Phrasenstrukturgrammatiken*, Technische Universitat Berlin, KIT-Report Nr. 45.

[Weisweber 1988] Wilhelm Weisweber (1988), 'Using Constraints in a Constructive Version of GPSG', in *Proc. 12. COLING-88*, Budapest, 738-743.