

Seminar
“Syntaxtheorien und computerlinguistische Praxis”

EINFÜHRUNG IN GPSG
ANHAND DER
ALVEY NATURAL LANGUAGE TOOLS

Filippo Svalduz
Bergstrasse 7
6004 Luzern
Tel. 041/420 96 13
svalduz@centralnet.ch

Charlotte Merz
Zederstrasse 12
8032 Zürich
Tel. 01/251 92 37
chmerz@access.unizh.ch

April 2000

Inhaltsverzeichnis

1. Theorie.....	3
1.1 Geschichte/ Begriff.....	3
1.2 Die Struktur von GPSG.....	4
1.2.1 Dominanz- und Präzedenzregeln.....	4
1.2.2 Syntaktische Kategorien.....	5
1.2.3 Generative Kapazität von ID/LP.....	6
1.2.4 Das Head Feature Principle (HFP).....	7
1.2.5 Das Foot Feature Prinzip (FFP).....	8
1.2.6 Das Control Agreement Principle (CAP).....	9
1.2.7 Darstellung von Merkmalen (Features) in GPSG.....	9
1.2.7.1 Feature Co-occurrence Restrictions (FCRs).....	9
1.2.7.2 Feature Specification Defaults (FSDs).....	10
1.3 GPSG und das Englische.....	10
1.3.1 Nominalphrasen.....	10
1.3.2 Infinitivkonstruktionen.....	11
1.3.3 Hilfsverbkonstruktionen.....	11
1.3.4 Adjektive.....	12
1.3.5 Adverbien.....	13
1.3.6 Weitere Konstruktionen.....	13
2. Der Alvey-Parser.....	14
2.1 Die Alvey Tools.....	14
2.1.1 Der Parser.....	14
2.1.2 Metagrammatische Kompilierung.....	15
2.1.3 Deklarationen.....	16
2.1.3.1 Merkmalsdeklaration.....	16
2.1.3.2 Set Declarations.....	17
2.1.3.3 Alias Declarations.....	17
2.1.3.4 Extension Declarations.....	17
2.1.3.5 Top Declarations.....	17
2.1.3.6 ID-Regel Deklaration.....	18
2.1.3.7 PS Regeln.....	18
2.1.3.8 Vererbungsregeldeklaration.....	18
2.1.3.9 Defaultregeldeklaration.....	18
2.1.3.10 Metaregeldeklaration.....	18
2.1.3.11 LP-Regeldeklaration.....	19
2.1.3.12 Wortdefinition.....	19
2.2 Der Generator.....	19
2.3 Der Morphologieanalysator.....	19
2.4 Benchmarks.....	20
3. Beispielsätze.....	21
3.1 Einleitung.....	21
3.2 Frageesätze [y/n, wh], Aussagesätze, Nebensätze).....	22
3.2.1 Do girls smile?.....	22
3.2.2 Who hits sheep?.....	24
3.2.3 Who do dogs eat?.....	27
3.2.4 Who does this dog belong to?.....	31
3.2.5 Dogs that bark eat.....	33
3.2.6 If a dog barks it eats.....	34
3.3 Unterscheidung Komplement / Adjunkt.....	35
3.3.1 The student of biology with long hair sees the girl.....	35
3.3.2 The student gives a bone to the dog every day.....	39
3.4 Raising-Konstruktionen, Infinitive, Hilfsverben.....	39
3.4.1 The student seems to eat.....	39
3.4.2 The dog wants the student to give him a bone.....	41
3.4.3 The student promises the dog to bring a bone.....	41
3.4.4 The dog has already eaten the bone.....	41
3.4.5 The dog must have eaten the bone.....	42

4. Konklusion	44
5. Bibliographie	45
6. Anhang.....	46

1. Theorie

1.1 Geschichte/ Begriff

Noam Chomsky war in den 50er- Jahren der Meinung, dass kontextfreie Grammatiken für die Beschreibung der natürlichen Sprache nicht genügten. Er entwickelte deshalb die Transformationsgrammatik, welche mit den flexibleren Transformationsregeln arbeitet. Diese Regeln erlauben eine Unterteilung der Syntaxstruktur eines Satzes in eine Tiefen- und eine Oberflächenstruktur, wobei die Tiefenstruktur mit kontextfreien Regeln und die Oberflächenstruktur mit Transformationsregeln aus der Tiefenstruktur generiert wird.

In den nachfolgenden drei Jahrzehnten realisierte die linguistische Forschung, dass mit den Transformationen ein zu mächtiges Instrument für die Beschreibung der natürlichen Sprachen entstanden war, denn sie erlauben Operationen, die für die natürlichen Sprachen nie gebraucht werden können. In diesem Zusammenhang entstand um 1980 die Generalized Phrase Structure Grammar (nachfolgend GPSG genannt) als neue generative Grammatiktheorie. GPSG hebt die Unterteilung in zwei Strukturlevel ganz auf (eine sog. monostratale Theorie) und arbeitet dadurch nicht mehr mit Transformationen, sondern mit Unifikation von Merkmalstrukturen. Neu in der GPSG ist auch die strikte Unterteilung in Dominanz- und Präzedenzregeln, welche Generalisierungen erlauben, die mit kontextfreien Regeln nicht möglich sind. Als Standardwerk wird im Allgemeinen das Buch vom Gazdar et al. (1985) genommen, und auch wir werden uns in unserer Arbeit auf diese Theorie abstützen.

Allerdings haben sich auch in GPSG einige Mängel entpuppt. Ein Nachteil ist, dass GPSG eine grosse Anzahl Dominanzregeln benötigt, welche weiter generalisiert werden könnten und so die Zahl der Regeln verringert würden. Eine Weiterentwicklung von GPSG ist die Head-driven Phrase Structure Grammar (nachfolgend HPSG genannt), welche genau das tut und deshalb mit einer weitaus kleineren Anzahl Regeln auskommt. HPSG und andere Weiterentwicklungen davon sind in vielen Details sehr unterschiedlich von GPSG; der formale Grundaufbau von GPSG, der wahrscheinlich wichtigste Beitrag von GPSG zur Syntaxanalyse, ist aber ohne Zweifel auch in den weiteren Generationen sichtbar.

1.2 Die Struktur von GPSG

1.2.1 Dominanz- und Präzedenzregeln

Die bekannten Phrasenstrukturregeln (nachfolgend PS-Regeln genannt) enthalten oft implizite Generalisierungen. Um dies zu vermeiden, verwendet GPSG eine Kombination von Dominanz- und Präzedenzregeln, welche die PS-Regeln ersetzt. Die Dominanzregeln (nachfolgend ID-Regeln, aus dem Englischen *immediate dominance*, genannt) regeln das Verhältnis zwischen einem Mutter- und einem Tochterknoten; die Reihenfolge der Geschwisterknoten wird dabei ausser acht gelassen. Die Abfolge der Geschwisterknoten wird erst durch die Präzedenzregeln (nachfolgend LP-Regeln, für engl. *linear precedence*, genannt) klargemacht. Diese Zweiteilung des Regelsatzes eignet sich, wie erwähnt, um Generalisierungen auszudrücken, die in PS-Regeln nur implizit enthalten sind. Ausserdem lässt sich eine variable Konstituentenreihenfolge mit einer minimalen Anzahl von ID-Regeln beschreiben. ID-Regeln werden ähnlich dargestellt wie PS-Regeln, die Elemente der rechten Regelseite werden allerdings durch Kommas getrennt.

$$(1) \quad A \rightarrow B_1, B_2, \dots, B_n$$

Dabei spielt die Reihenfolge der Komponenten der rechten Regelseite keine Rolle.

$$(2a) \quad A \rightarrow B_1, B_2$$

$$(2b) \quad A \rightarrow B_2, B_1$$

Die beiden Regeln (2a) und (2b) sind äquivalent. Bei den ID-Regeln wird zwischen lexikalischen und nicht-lexikalischen Regeln unterschieden. Die lexikalischen Regeln haben einen Kopf, der das Merkmal *Subcat*, welches die Subkategorisierung regelt, umfasst. Nähere Informationen zum Merkmal *Subcat* stehen im Abschnitt 1.2.2.

LP-Regeln werden folgendermassen anotiert:

$$(3) \quad A < B$$

In Beispiel (3) wird ausgesagt, dass A vor B stehen muss, wenn die beiden Knoten als Geschwister auftreten. LP-Regeln besitzen ausserdem ein weiteres Merkmal. Sie gelten immer für alle ID-Regeln. Es gibt also keine Möglichkeit, eine LP-Regel nur auf eine bestimmte ID-Regel zu münzen. Ein lokaler Baum gilt also als wohlgeformt, wenn er mindestens einer ID-Regel und allen relevanten LP-Regeln genügt. Damit ergibt sich eine Schwierigkeit des Einsatzes von ID/LP-Regeln. Die Regeln können nur dann

angewendet werden, wenn die Geschwisterreihenfolge unabhängig vom Mutterknoten ist. Wie auch die PS-Regeln sind die ID/LP- Regeln nur auf lokale Bäume anwendbar.

1.2.2 Syntaktische Kategorien

- (4a) Stefan scratched himself.
- (4b) Who do you think the man who knows Max saw?

Um die Sätze mit diskontinuierlichen Elementen wie in Beispiel (4a) und (4b) ausdrücken zu können, werden komplexe syntaktische Kategorien eingeführt. Syntaktische Kategorien beinhalten Merkmal-Wert Paare, auch Merkmalstrukturen genannt. Diese setzen sich aus Werten zusammen, die eine Zeichenkette, eine Koreferenz (ein Verweis auf einen anderen Wert) oder eine Merkmalsstruktur enthalten. In den Beispielen (4a) und (4b) wird der Einsatz von ID/LP-Regeln schwierig, da einige syntaktische Einheiten nicht vom Prinzip der unmittelbaren Dominanz erfasst werden können. GPSG löst dieses Problem mit Hilfe syntaktischer Kategorien, welche den Transport syntaktischer Merkmale im Syntaxbaum regeln¹. Bei diesen syntaktischen Merkmalen handelt es sich dabei nicht um Atome, sondern um komplexe Strukturen, die aus kleineren Elementen zusammengesetzt sind.

- (5a) 'Haus' → N[Kasus=Nom, Numerus=Sg, Genus=Neut,...]
- (5b) 'fahren' → V[Pers=1, Numerus=plur, Temp=Präs,...]

Die Konstituenten in GPSG bestehen aus Merkmalsstrukturen. Es wird dabei zwischen den drei Typen von Merkmalen, den allgemeinen, Head²- und Foot-Merkmalen³ (engl. *features*) unterschieden. GPSG verwendet dabei folgende Notation:

- (6a) [CASE=ACC]
- (6b) [CASE ACC]
- (7) [ACC]
- (8) [+ PLU]
- (9a) [VFORM]
- (9b) ~[VFORM]
- (10) V = V0 = V⁰ = {[N -], [V +], [BAR 0]}
- (11a) V2
- (11b) V[2]

Beispiel (7) zeigt dabei ein eindeutiges Merkmal, während Beispiel (8) ein binäres Merkmal darstellt. Man beachte, dass in (8) kein Merkmal vom Typ 'Numerus' definiert

¹ Die Prinzipien zum Transport von diesen Merkmalen sind in den Abschnitten 1.2.4 Das Head Feature Principle (HFP), 1.2.5 Das Foot Feature Prinzip (FFP) und 1.2.6 Das Control Agreement Principle (CAP) beschrieben.

² Headfeatures sind die Merkmale, welche am Kopf einer Phrase instantiiert sind und mit dem HFP nach oben gegeben werden.

ist. Vielmehr wird der Numerus über das Merkmal PLU mit dem Wertebereich {+,-} definiert. Das Beispiel (9b) schliesslich repräsentiert ein Merkmal, das nicht auftreten darf. Wie in (10) gezeigt, wird der Bar-Level entweder nicht geschrieben (besonders bei Bar 0) oder als einfache Ziffer bzw. Exponent notiert um nicht mit der Angabe der Subcat-Klasse verwechselt zu werden ((11a) zeigt den Bar-Level, (11b) zeigt die Subcat-Klasse). Dies ist möglich, weil Subcat-Merkmale nur in Bar-Level 0 auftreten.

Aufgrund der Ähnlichkeiten zwischen VPs ohne Subjekt und Infinitivkonstruktionen (S) mit Subjekt schlossfolgerte man, dass zwischen VP und S kein grundlegender Unterschied besteht. Ein Nebensatz wird vom Hauptsatz durch das Merkmal 'comp' unterschieden. Beispiel (13) zeigt den ersten Versuch einer generellen Regel zur Einführung eines Nebensatzes. Es gilt:

- (12a) S = V2[Subj +] = {[N-], [V+], [Bar 2], Subj +}
 (12b) VP = V2[Subj -] = {[N-], [V+], [Bar 2], Subj -}
 (13) V2[Subj +, Comp] → Comp, V2[Subj +, ~Comp]
 (14a) V2 → V[Subcat 1] (die)
 (14b) V2 → V[Subcat 2], NP (love)
 (14c) V2 → V[Subcat 3], NP, PP[Pform to] (give)

Subkategorisierung wird über das Merkmal 'subcat' geregelt. Wenn man also z.B. für das Verb *to love* [SUBCAT 2] bestimmt, dann wird mit ID-Regel (14b) gesagt, dass transitive Verben mit genau einem NP-Geschwisterknoten auftreten. Dem Merkmal (im Falle von *to love* Transitiv) wird dabei ein expliziter Zahlenwert zugeordnet. Wir stützen uns dabei auf die Subcat-Liste von Gazdar et al. (1985), die im 6. Anhang zu finden ist. Subcat-Werte werden auch an Adjektive, Präpositionen und Nomina vergeben.

Lexikalische Heads werden durch das Subcat-Merkmal von den anderen Kategorien unterschieden. Folgende LP-Regel bewirkt, dass ein lexikalischer Head vor allen Geschwisterkategorien steht:

- (15) [Subcat] < ~[Subcat]

1.2.3 Generative Kapazität von ID/LP

ID/LP-Regelsätze können die meisten PS-Grammatiken beschreiben. Gewisse PS-Grammatiken können jedoch nur über Umwege dargestellt werden.

³ Footfeatures sind z.B. [+Q] für Question oder [+R] für Relativsätze, also Merkmale, die in der Gegenrichtung zu den Kopfmerkmalen laufen.

- (G1) S → NP VP
 NP → Art NG
 VP → V (NP) (PP) (S)
 PP → P NP
 NG → N (S) (PP)

- (G2) S → NP, VP
 NP → Art NG
 VP → V, (NP), (PP), (Z)
 PP → P, NP
 NG → N, (S), (PP)
 Z → S
- } ID-Regeln

- NP < VP
 Art < NG
 V < X
 N < X
 NP < PP
 PP < Z
 S < PP
- } LP-Regeln

Die PS-Grammatik in (G1) kann nicht in eine gleichwertige ID/LP-Grammatik übersetzt werden. Weil Präzedenzregeln für alle ID-Regeln gelten und in (G1) die Reihenfolge von (PP) und (S) je nach Mutterknoten verschieden ist, lässt sich keine allgemeine LP-Regel generieren. Grammatik (G2) umgeht das Problem durch die Einführung des Hilfssymbols Z. Das Problem ist allerdings, dass durch dieses Hilfssymbol manchen Sätzen eine andere Struktur zugewiesen wird als in (G1). Aus Sicht der Linguistin ist die Einführung eines Hilfssymbols aber nicht erstrebenswert.

1.2.4 Das Head Feature Principle (HFP)

Ein finiter Satz enthält eine finite Verbalphrase, welche ihrerseits ein finites Verb beinhaltet. Durch diese Regelmässigkeit wird Unifikation von Merkmalen überhaupt möglich. Die Übereinstimmung von Merkmalen, oft auch Vererbung von Merkmalen genannt, macht sich GPSG zunutze. Den Transport von Merkmalen innerhalb der Satzstruktur regelt das Head Feature Principle (HFP), auch Head Feature Convention

genannt. Das Prinzip postuliert, dass Mutterknoten und Headtochter in allen Headmerkmalen übereinstimmen müssen.

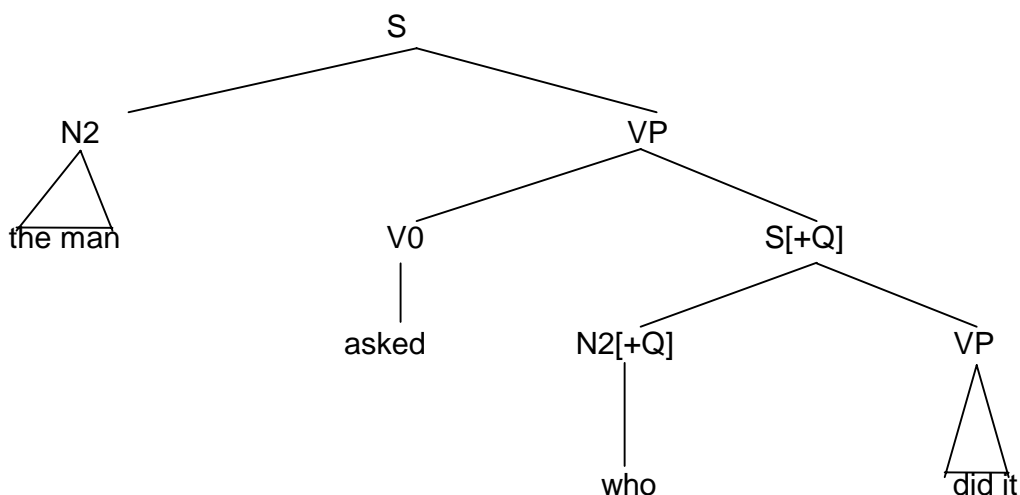
Jede ID-Regel enthält einen Head, der mit 'H' markiert wird. Durch Unifikation wird die Information der Merkmale *N*, *V*, *Bar* auf den Head übertragen.

(16) $VP \rightarrow H_0[\text{Subcat } 2], NP$

1.2.5 Das Foot Feature Prinzip (FFP)

Das FFP ist das zweite von GPSG verwendete Prinzip zur Beschreibung des Merkmalstransportes innerhalb der Syntaxstruktur. Das Prinzip postuliert, dass jedes Foot-Feature, das an irgend einem Tochterknoten auftritt, auch bei einem Mutterknoten auftreten muss. Es wird für die Behandlung von Relativpronomen und Interrogativpronomen verwendet und nur auf instanziierte Merkmale, d.h. Merkmale, die einer Kategorie zugefügt werden und nicht vererbt wurden, angewendet. Beispiel (T1) zeigt die Auswirkungen des FFP. In diesem Baum müssen $S[+Q]$ und $N2[+Q]$ im Merkmal $+Q$ übereinstimmen. Da $N2$ aber nicht der Kopf von S ist, kann nicht das HFP nicht angewendet werden. Die Unifikation von $[+Q]$ auf den zweiten Tochterknoten, VP , wird durch eine spezielle FCR (Siehe 1.2.7.1) verhindert ($V2 \Rightarrow \sim[Q]$)

(T1)



1.2.6 Das Control Agreement Principle (CAP)

Das CAP regelt die Kongruenzbeziehungen zwischen finitem Verb und Subjekt. Ein Head-Merkmal *Agr* wird zum Zweck der Angleichung von Numerus und Person zwischen Verb und Subjekt eingeführt. Dieses Merkmal steht beim Verb und trägt eine Kopie des Subjekts. Dadurch erreicht man, dass das Subjekt die Form des Verbs bestimmt. Die Kernaussage des CAP ist, dass die Kontroll-Merkmale einer Target-Kategorie (Verb) mit den Merkmalen des Controllers (Subjekt) übereinstimmen müssen.

1.2.7 Darstellung von Merkmalen (Features) in GPSG

1.2.7.1 Feature Co-occurrence Restrictions (FCRs)

Wie der Name antönt, limitieren FCRs die Merkmale, welche in einer Konstituente zusammen auftreten können. Eine Kategorie, welche alle von FCR gestellten Bedingungen erfüllt, heisst legale Kategorie. Folgende Beispiele zeigen FCR-Konstruktionen:

- (17a) [VFORM] ==> [+V, -N]
- (17b) [NFORM] ==> [+N, -V]
- (18) [+ SUBJ] ==> [+V, -N, Bar 2]
- (19) [COMP] <==> [+SUBJ]
- (20a) [Bar 0] <==> [Subcat] & [N] & [V]
- (20b) [Bar 1] ==> ~[Subcat]
- (20c) [Bar 2] ==> ~[Subcat]
- (21) [Past] ==> [Fin]

In Beispiel (17a) wird ausgesagt, dass eine Merkmalsstruktur, die das Merkmal *VFORM* enthält, auch die Merkmale *+V* und *-N* enthalten muss, also eine verbale Kategorie sein muss. Analoges gilt für *NFORM*, ebenso für *PFORM*. Das Merkmal-Wert-Paar *[+ SUBJ]* darf laut Regel (18) nur in phrasalen Merkmalsstrukturen auftreten. Der Doppelpfeil in (19) zeigt eine zweiseitige Implikation an. Die Regel sagt aus, dass satzartige Kategorien einen Complementizer haben und umgekehrt. Wie bereits erwähnt, besitzen nur lexikalische Heads das Merkmal *Subcat*. Dies wird in den Regeln (20a) bis (20c) festgehalten. Eine Gesetzmässigkeit des Englischen stellt die Regel (21) dar. Im Englischen gibt es im past tense keinen Infinitiv, weshalb ein in der Vergangenheitsform stehendes Verb zwangsläufig finit sein muss, was durch Regel (21) beschrieben wird. Feature Cooccurrence Restrictions sind feste Regeln, dessen Werte nicht überschrieben werden.

1.2.7.2 *Feature Specification Defaults (FSDs)*

- (22a) I hate them.
- (22b) I gave the book to her.
- (22c) I would prefer for her to stay.
- (22d) Him, he's a complete head-case.

Beobachtet man die Fälle in den Beispielen (22a) – (22d), so sieht man, dass das Subjekt eines finiten Verbs überall im Nominativ und das Objekt von Verben und Präpositionen im Akkusativ steht. Um diese Erkenntnis im ID-Regelformat zu formulieren, benötigte man eine sehr grosse Anzahl neuer ID-Regeln. Man hat daher beschlossen, einen Default-Wert für syntaktische Kategorien zu definieren. In Beispiel (23) steht, dass eine NP den Wert [Acc] zugeteilt bekommt, falls nicht das Gegenteil anderswo gefordert wird. Diese Standard-Einstellungen können allerdings –im Gegensatz zu FCRs- von ID-Regeln, FCR und HFP überschrieben werden, falls ein spezifischer Wert durch eine Regel erfordert wird.

(23) FSD: [+N, -V, Bar 2] ≡ [Acc]

In Beispiel (23) steht, dass eine NP den Wert [Acc] zugeteilt bekommt, falls nicht das Gegenteil anderswo gesagt wird. Da FSDs dieselbe Notation wie FCRs benutzen, muss man die Regeln mit ihrem Kürzel markieren.

FSDs sind sehr nützlich für GPSG, können aber auch problematisch werden. Ihre Interaktionen mit anderen Feature Principles werden schnell sehr komplex, und es ist unter Umständen sehr schwer zu entscheiden, welche lokale Syntaxstruktur nun die wohlgeformte ist.

1.3 **GPSG und das Englische**

GPSG für das Englische wurde zu Beginn der 80er Jahre entwickelt. Als Standardreferenz und umfassendste Darstellung gilt [Gazdar et al. 85] (im Anhang ist ihre Liste der Merkmale und Regeln aufgeführt). [Bennett 95] erweiterte das Standardschema um einige Regeln.

1.3.1 **Nominalphrasen**

Im GPSG-Standardwerk ([Gazdar et al. 85]) werden Nominalphrasen nur kurz abgehandelt. Es werden sieben Regeln zur Subkategorisierung von Nomina angegeben und durch drei nicht-lexikalische ID-Regeln ergänzt.

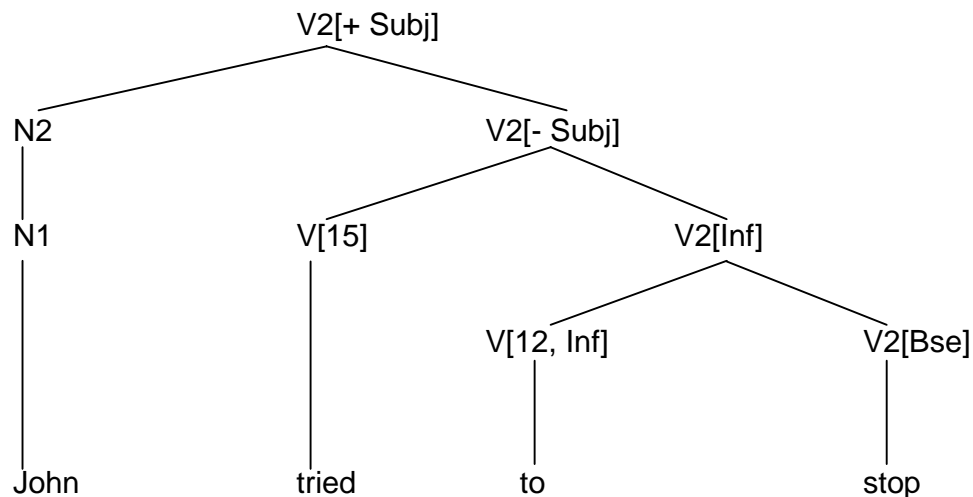
1.3.2 Infinitivkonstruktionen

- (24a) Jack made Rose laugh at Jim.
- (24b) The rich girl likes to drive her red sports car.
- (24c) Sam prefers to leave.
- (25) V2 → H[n], N2, V2[Vform Bse]
- (26a) V2 → H[15], V2[Vform Inf]
- (26b) V2[Vform Inf, Aux +] → H[12], V2[Vform Bse]

Bei den Infinitivkonstruktionen wird zwischen Sätzen mit *to*-Infinitiv und reinem Infinitiv unterschieden. Für reine Infinitivsätze genügt eine Regel, welche das Auftreten einer Grundform an der Stelle V2 postuliert (Siehe Regel (25)). *To*-Infinitiv-Konstruktionen erfordern dagegen spezielle Regeln, die den Einsatz von *to* festlegen. In GPSG wird *to* als nicht-finites Hilfsverb betrachtet, weil es wie andere Hilfs- und Modalverben unmittelbar vor einem nicht-finiten Verb auftreten kann.

Mit den beiden Regeln (26a) und (26b) lässt sich zum Beispiel folgendes herleiten:

(T2)

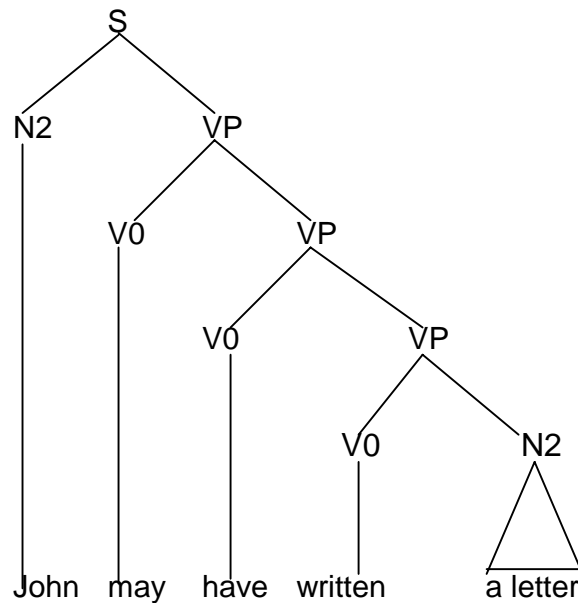


1.3.3 Hilfsverbkonstruktionen

Hilfs- und Modalverben gelten in GPSG als Teil der Verbklasse und werden durch das Merkmal *Aux +/-* unterschieden. Der nicht-auxiliäre Verbteil bildet zusammen mit den Komplementen eine Konstituente. Beispiel (26c) zeigt die Schachtelung der Elemente in GPSG an.

- (26a) He may have been writing a letter.
- (26b) He may [write a letter].
- (26c) He [may [have [been [writing a letter]]]]].
- (27) V2[+ Aux] → H[n], V2[Vform Bse] (can, may, should)
- (28) V2[+ Aux] → H[n], V2[Vform Psp] (have)
- (29) V2[+ Aux] → H[n], V2[Vform Prp] (be)

(T3)



Mit Hilfe der Regeln (27) bis (29) lassen sich Strukturen wie in (T3) herleiten.

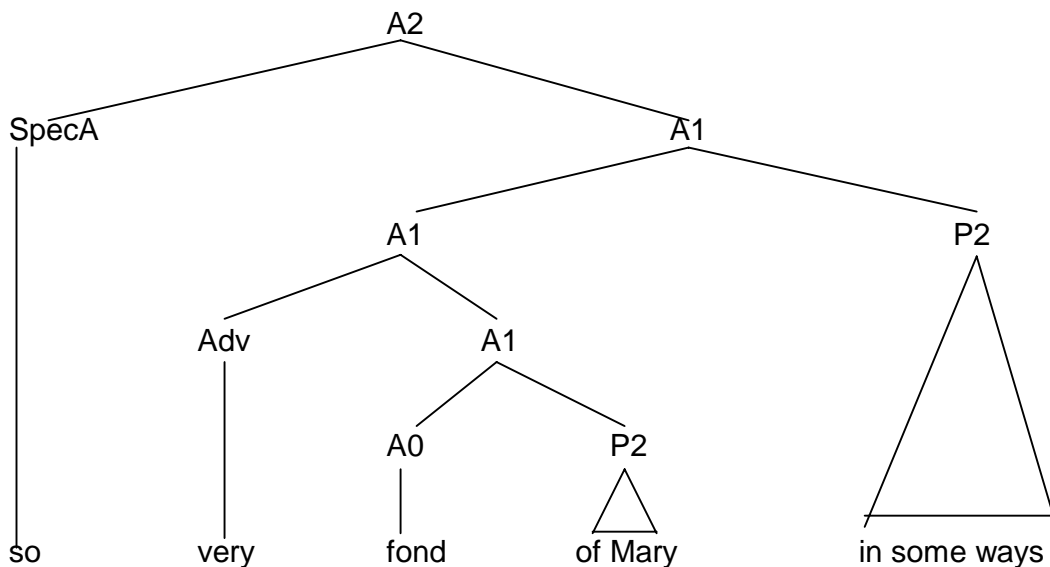
1.3.4 Adjektive

Adjektive können Adjektivphrasen bilden. Auch in GPSG wird dies erfasst. So müssen die Regeln zum Beispiel folgende Phrasen abdecken können:

- (30) very happy
- (31) so fond of his sister
- (32) similar to his sister's neighbour
- (33a) A2 → (SpecA), A1
- (33b) A1 → A0[Subcat 25], P2[Pform to]
- (33c) A1 → Adv, A1
- (33d) A1 → P2, A1
- (33e) SpecA < [Bar 1]
- (33f) Adv < A1 < P2

Eine der Regeln, die notwendig sind um eine Adjektivphrase abzudecken, ist Regel (33). Diese drückt das Adjektiv 'similar' aus. Das Ergebnis ist eine Baumstruktur wie in (T4).

(T4)



1.3.5 Adverbien

In GPSG werden Adverbien als Unterklasse der Adjektive angesehen und durch das Merkmal *Adj +/-* unterschieden. Durch diese Ähnlichkeit können auch Adverbien Adverbphrasen bilden. Im Unterschied zu Adjektiven erscheinen bei Adverbien normalerweise keine Komplemente (34). Es gibt natürlich auch Ausnahmen (35).

- (34) He jumped into the car hastily.
- (35a) Jim plays similarly to Alan
- (35b) John reached a conclusion independently of Fred.

Die Beschreibung der Verteilung von AdvP in einem Satz ist schwierig. Ausserdem gibt es eine kleine Anzahl Verben, die Komplemente in Form einer AdvP fordern. Adverbien treten in verschiedenen Klassen auf. Satzmodifizierende Adverbien (zB. *probably*) treten sowohl am Rande eines Satztes, wie auch innerhalb einer VP auf, während verbmodifizierende Adverbien nur innerhalb der VP auftreten können. Adverbien scheinen, wie wir sehen werden, ein Problem des später vorgestellten Alvey-Parser zu sein.

1.3.6 Weitere Konstruktionen

Die Behandlung von weiteren Konstruktionen wird im Teil 3 dieser Arbeit anhand von Beispielsätzen erklärt.

2. Der Alvey-Parser

2.1 Die Alvey Tools

Der Alvey-Parser ist Teil des Grammar Development Environment (GDE) und wird von seinen Erfindern als mächtiges Werkzeug zur Entwicklung und Erforschung natürlichsprachlicher Grammatiken beschrieben. Die Implementation von GDE erfolgte in Lisp und wurde am Institut für künstliche Intelligenz der Universität Edinburgh entwickelt. Die Alvey Tools gliedern sich in drei grössere Module: Parser, Generator und morphologischer Analysator.

2.1.1 Der Parser

Der von Gazdar et al. (1985) entwickelte Regelsatz von GPSG dient dem Parser, der von den Forschern Phillips und Thompson 1987 gestaltet wurde, als Basis. Im Parsermodul enthalten ist ausserdem ein Wörterbuch des Englischen. Der Parser geht mit den Regeln auf eine etwas andere Art um, indem er die Metagrammatik in eine 'Objektgrammatik' überführt, welche auf Unifikation basiert. Es geht –kurz gesagt- in dieser Objektgrammatik darum, die Regeln in eine für den Computer einfachere (für uns vermutlich kompliziertere) Form zu bringen.

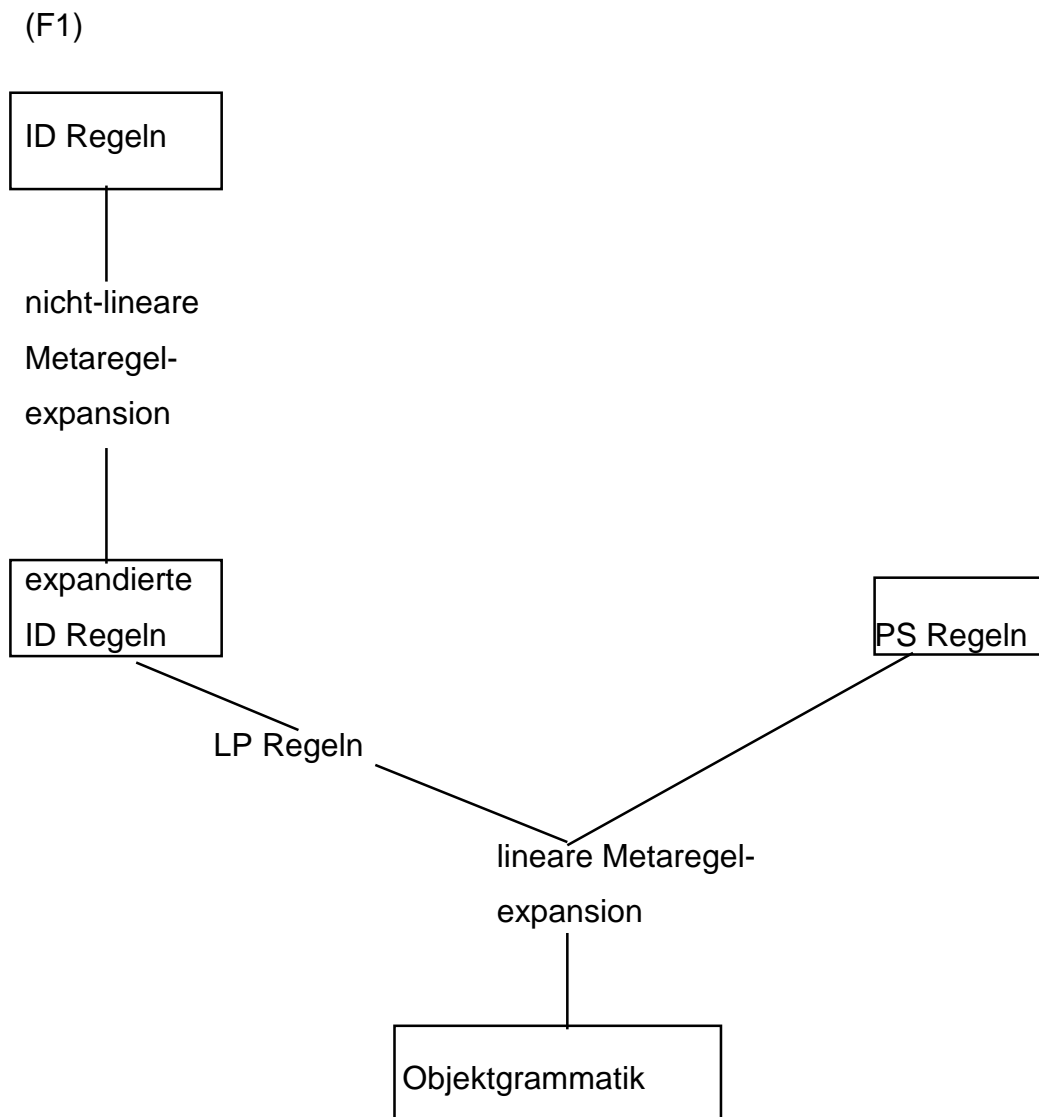
Der metagrammatische Formalismus des Parsers weist einige Unterschiede zur GPSG Konvention auf. Das Ziel der Entwickler bestand darin, den Parser durch zusätzliche Möglichkeiten flexibler zu gestalten. So besteht die Möglichkeit, anstelle der ID/LP-Struktur mit herkömmlichen Phrasenstrukturregeln zu operieren, um beispielsweise gewisse Ausnahmen, die das ID/LP Format nicht abdecken kann, zu erfassen. Beispiel (36) illustriert eine solche Phrasenstrukturregel im Alvey-Parser, die die Verwendung von *either* beschreibt.

(36) PSRULE V/PRO/SO2: ; (kim) didn't either. (won't either, mustn't either)
V[+AUX, +PRO, SUBCAT NULL, NEG +, -INV] --> H[-PRO, SUBCAT
@s, SUBTYPE @ss, NEG +, -INV] [SO +, NEG -, CONEG +].

Ein problematisches Konzept des ID/LP Formats ist das Konzept der Gleichzeitigkeit. Alle Regeln werden bekanntlich gleichzeitig zur Projektion auf eine lokale Baumstruktur eingesetzt. Dies ist für den Computer nicht machbar. Die Objektgrammatik von GDE dagegen verfügt über eine inhärente Deklaration der Regelexpansion.

2.1.2 Metagrammatische Kompilierung

Bei der Kompilierung wird aus jeder Regel ein Set aus zwei Regeln mit und ohne Optionskategorie gebildet. Bei der Abarbeitung der Objektgrammatik wendet der Parser zuerst Vererbungsregeln, dann Defaultregeln und Kategoriedeklarationen auf den ID-Regelsatz an. Auf den entstandenen ID-Regelsatz wendet er darauf jede einzelne Metaregeln an. Anders gesagt, ist die Reihenfolge der Regeldeklaration der Metaregeln entscheidend. Nach jeder Anwendung einer Metaregel wird die daraus resultierende ID-Regel generiert und dem Regelsatz hinzugefügt (falls sie nicht schon existiert). Nach Abarbeitung der Metaregeln wird das Resultat nach den LP-Regeln linearisiert. In dieser Phase werden auch vorhandene PS-Regeln abgearbeitet (Siehe Figur F1). Vererbungsregeln werden in GDE in der Regel vor den Default-Regeln abgearbeitet. Dies kann jedoch durch setzen eines Attributs umkonfiguriert werden. Weiterhin lässt sich auch bestimmen, ob mehrfache identische Metaregelexpansionen derselben ID-Regel auf eine einzige reduziert werden sollen. Dieses Feature scheint auf den ersten Blick interessant zu sein. Tatsächlich steht dieser Schalter default-mässig auf ON. Für semantische Betrachtungen aber kann es von Vorteil sein, mehrere identische Parses zuzulassen.



Das Kompilierungsverfahren für die semantische Repräsentation sei hier ausser Acht gelassen.

2.1.3 Deklarationen

Um die Ähnlichkeit von GDE und GPSG zu verdeutlichen, möchten wir hier die verschiedenen Deklarationen mit je einem Beispiel vorstellen.

2.1.3.1 Merkmalsdeklaration

Im Gegensatz zu GPSG erlaubt GDE, dass Merkmale nicht nur fixe Werte, sondern auch Variablen beinhalten können. Merkmale werden verwendet um Kategorien zu bilden, wobei eine Kategorie eine ungeordnete Sammlung von Merkmalen mit einem Wert darstellt.

(37) BAR {0, 1, 2}

Das Beispiel (37) zeigt das Feature BAR, welches die Werte 0, 1 oder 2 annehmen kann.

2.1.3.2 Set Declarations

Set Deklarationen definieren eine Gruppe von features, die dasselbe Verhalten aufweisen.

(38) NOMINALHEAD = {PLU, PER, CASE}

Das Beispiel (38) zeigt, dass die drei Features PLU, PER und CASE in einem Set NOMINALHEAD gruppiert werden könnten.

2.1.3.3 Alias Declarations

Aliasdeklarationen stellen einfach ein Mittel zur Abkürzung komplexerer Kategorien dar.

(39) N2 = [N +, V -, BAR 2]
 + PLU = [PLU +]

2.1.3.4 Extension Declarations

Dieser Deklarationstyp wird dazu verwendet, Strukturen, die nicht grundsätzlich in einer Kategorie auftauchen, darzustellen. Ein Beispiel ist das SLASH Feature. Solche Merkmale, die in keiner Kategoriedeklaration auftauchen, können über eine Extension Declaration deklariert werden.

2.1.3.5 Top Declarations

Dieser Deklarationstyp besteht aus einer Anzahl Kategorien, die als Filter dienen. Er zwingt GDE dazu, nur Parses zu akzeptieren, bei welchen der oberste Knoten einer der Kategorien (oder einer Erweiterung einer Kategorie) entspricht.

(40) S[FIN +, COMP NORM], N2.

Das Beispiel (40) zwingt den Parser dazu, nur Parses, dessen Anfangsknoten die Form S[FIN +, COMP NORM] oder N2 aufweist, zurückzuliefern. Wenn keine Top-Deklarationen in einer Grammatik vorliegen, werden alle vollständigen Parses zurückgeliefert.

2.1.3.6 ID-Regel Deklaration

Dieser Deklarationstyp dient dazu, ID-Regeln darzustellen.

(41) VP/TAKES_NP : VP \rightarrow H[SUBCAT NP], N2.

2.1.3.7 PS-Regeln

Wie schon erwähnt, können in GDE ausser ID-Regeln für spezielle Konstruktionen auch PS-Regeln eingesetzt werden.

(42) Heavy_NP_Shift : VP \rightarrow H[SUBCAT NP_PP] PP NP[+Heavy].

2.1.3.8 Vererbungsregeldekларation

Vererbungsregeln definieren den Merkmalstransport zwischen Mutter- und Tochterknoten. Vererbungsregeln werden eingesetzt, um beispielsweise die Head Feature Convention umzusetzen

(43) HFC_NOMINAL :
[N +, V -] \rightarrow [H +], U. F(0) = F(1), F in NOMINALHEAD.

Im vorliegenden Beispiel entspricht dabei die Null dem Mutterknoten [N +, V -] und die Eins dem Tochterknoten [H +].

2.1.3.9 Defaultregeldekларation

Wie bereits erwähnt, dient eine Defaultregel dazu, einer Variable einen bestimmten Wert zuzuordnen, und hat also keine Wirkung, wenn das spezifizierte Merkmal bereits mit einem Wert aus einer ID Regel versehen ist.

2.1.3.10 Metaregeldekларation

Metaregeln vergrössern automatisch und systematisch die Objektgrammatik und bedienen sich dabei des Basisregelsets aus ID- und PS-Regeln.

Ein Beispiel:

(44) PASS: VP \rightarrow W, N2. \implies VP [PAS] \rightarrow W, (P2[by]).

Diese Metaregel soll passive VPs aus aktiven VPs ableiten. Die Regel wird beispielsweise auf folgende ID-Regel zutreffen:

(41) VP/TAKES_NP : VP \rightarrow H[SUBCAT NP], N2.

Der Parser führt dabei mit Hilfe der Metaregel etwas ähnliches wie eine Unifikation aus. Der Unterschied zur Unifikation besteht darin, dass falls bei verschiedenen in einer Kategorie verschiedene Werte vorliegen, der Wert der Metaregel vorrang erhält. Das Resultat sieht folgendermassen aus:

(45) $VP/TAKES_NP(PASS) : VP[PAS] \rightarrow H[SUBCAT\ NP], (P2[by])$.

2.1.3.11 LP-Regeldeklaration

Die LP-Regeln werden wie in GPSG eingesetzt.

(46) $L1 : [SUBCAT] < \sim[SUBCAT]$.

2.1.3.12 Wortdefinition

Worte sind nicht Teil des metagrammatischen Formalismus, können aber dazu benutzt werden, um mittels Parser und Generator eine Grammatik zu testen. Eine Wortdefinition besteht aus dem Wort und einer oder mehreren syntaktischen Kategorien, die mit dem Wort assoziiert werden sollen.

(47) $cats: \quad N[-POSS, PLU+, PRO-, PN-, SUBCAT\ NULL] : (plu\ cat')$,
 $\quad N[+POSS, PRO-, PN-, SUBCAT\ NULL] : (poss\ cat')$.

2.2 Der Generator

Der Generator spielt für unsere Betrachtungen keine entscheidende Rolle. Wir beschränken uns daher darauf zu erwähnen, dass es ihn gibt. Der Generator erzeugt die Syntaxstruktur und soll den Entwickler dabei unterstützen, das Phänomen der Übergenerierung zu untersuchen. Der Generator kann in einen automatischen oder manuellen Modus geschaltet werden. Jeder Knoten der Syntaxstruktur kann gezielt expandiert werden.

2.3 Der Morphologieanalysator

In GDE ist ein morphologischer Analysator eingebaut. Ein spezielles Feature sei hier beschrieben. Die vorliegende Version des morphologischen Analysators erlaubt die Assoziierung von Semantik und Wort-Grammatikregeln. Die Morpheme *cat* und *+s* (wobei *+s* das Pluralsuffix repräsentiert) sind zum Beispiel folgendermassen definiert:

(48) (cat cat (N COUNT +)) cat' ())
 (+s +s ((STEM (N (COUNT +))) plu ()))

Zusammen mit der Wort-Grammatikregel *N-SUFFIXES*, der wir in unseren Beispielssätzen wieder begegnen werden und die die zwei Morpheme kombiniert, würde die Morphemsequenz *cat +s* die semantische Analyse (*plu cat'*) erhalten.

2.3.1 Das Lexikon

Das zum Teil automatisch generierte (und somit fehlerhafte) Lexikon soll hier auch nicht unerwähnt bleiben. Beispiel (48a) zeigt einen Originaleintrag aus dem Lexikon.

(48a) (|car| || ((FIX NOT) (BAR |0|) (SUBCAT NULL) (INFL +) (N +) (V -)
 (POSS -) (PRO -) (PROTYPE NONE) (PN -) (PLU -) (NFORM NORM)
 (PER |3|) (ADV -) (PART -) (DEMON -) (NUM -) (CONJ NULL) (AT +)
 (LAT +) (COMPOUND NOT)) CAR NIL)

2.4 Benchmarks

Die folgenden Testresultate sollen einen kleinen Überblick über die Geschwindigkeit von GDE geben. Die Zahlen drücken dabei die Anzahl Sekunden CPU-Zeit aus.

	Mac Ilci Procyoil CL	053100 Allegro CL	Sun Sparc l+ AKCL	HP 9000/350 HP CL 11
(1) Read	56	14	7.9	13
(2) Compile	230	105	77	140
(3) Word lookup	3.2	1.4	0.7	2.1
(4) Parse	1.6	0.6	0.8	1.3

Der erste Test beinhaltet die Testversion der Alvey Grammatik vom Stand Juni 1991. Der zweite Test zeigt die Dauer der Kompilierung der ganzen Grammatik an. Der dritte Test misst, wie lange das System benötigt um einen lexikalischen Eintrag nachzuschlagen, während der vierte Test schliesslich das Parsen eines ganzen Satzes misst (ohne das Nachschlagen im Wörterbuch).

3. Beispielsätze

3.1 Einleitung

Im Rahmen des diesjährigen Seminars in der Computerlinguistik werden verschiedene Grammatiktheorien besprochen und deren Lösungsansätze anhand von Beispielsätzen illustriert. So stehen uns vierzehn Beispielsätze, die verschiedene syntaktische Probleme beinhalten, zur Verfügung. Damit wir die Beispielsätze mit dem Alvey-Parser analysieren konnten, mussten wir sie teilweise verändern, weil nicht alle Wörter im Lexikon vorhanden sind. Wir stellten aber auch fest, dass gewisse Wörter zu unzulänglichen Parse-Resultaten führen, während andere mit derselben syntaktischen Struktur keinerlei Probleme verursachen. Wir führen dies auf gewisse Inkonsistenzen im Alvey-Lexikon zurück.

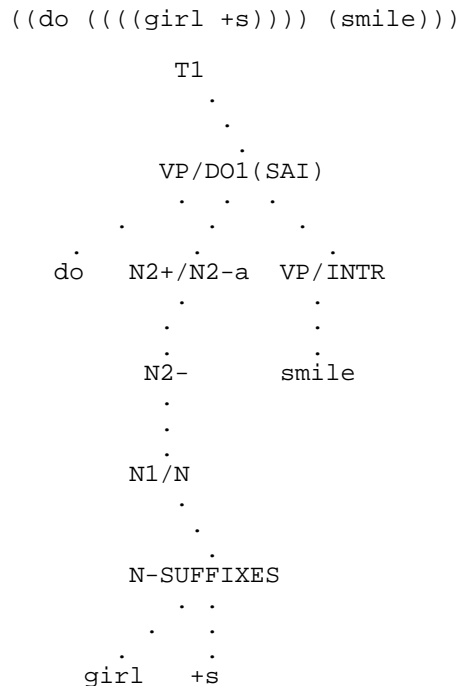
Ein anderes Problem bereitet uns die Anzahl der von Alvey ausgegebenen Syntaxstrukturen. Der Alvey-Parser stellt in den meisten Sätzen mehrere mögliche Syntaxstrukturen her. Dies ist insofern noch kein Problem, da dies mit den Ambiguitäten in den natürlichen Sprachen gut erklären lässt und ein Parser vorzugsweise übergenerierend ist. In gewissen Fällen (vor allem in Sätzen mit Adverbien mit flexiblen Positionen) errechnet der Alvey-Parser aber über hundert Syntaxstrukturen. Dies ist unserer Meinung nach zu viel, denn damit lässt es sich nicht mehr in einem vernünftigen Zeitrahmen arbeiten und es kann durchaus passieren, dass das Programm nicht mehr funktioniert. Das Problem der Übergenerierung hat für uns auch die Auswirkung, dass wir nicht alle Beispielsätze behandeln haben. Der Aufwand für die Untersuchung von so vielen Syntaxstrukturen hätte den Umfang einer Seminararbeit gesprengt. Wir haben uns deshalb auf nur eine, nämlich die korrekte Syntaxstruktur zu jedem Beispiel beschränkt, obwohl auch das Zustandekommen der überflüssigen Parses eine interessante Fragestellung wäre.

Eine weitere Bemerkung muss zum Aufbau der Alvey-Regeln gemacht werden. Wir zeigen nur ID-Regeln, da diese die interessantesten Informationen beinhalten, und lassen die lexikalischen, LP-, Propagation, Feature Default und anderen Regeln des Umfangs wegen beiseite. Eine weitere Kürzung haben wir bei den ID-Regeln vorgenommen, indem wir die gesamte Semantik, die dort kodiert ist, gestrichen

haben. Semantik interessiert und in dieser Arbeit nicht und würde die Leserlichkeit der Regeln erschweren.

3.2 Frageesätze [y/n, wh], Aussagesätze, Nebensätze)

3.2.1 Do girls smile?



Der Anfang jedes Satzes wird im Alvey-Parser mit dem Symbol T1 beschrieben. Hinter T1 verbirgt sich die Information, dass jeder gültige Satz finit sein muss.

(49) IDRULES T1

root symbol for the parser. It is here that the information that root sentences are always finite is encoded.

[T S] →

S [H+, COMP NORM, +FIN] :

In GPSG wird *do* als Hilfsverb behandelt, das vor einer Grundform eines Verbes auftritt. Folgende ID-Regel kommt im Alvey-Parser zum Zug:

(50)IDRULE VP/DO1

"does dance"

VP[+ AUX, + FIN, VFORM NOT, ELLIP -, COORD -] →

H[SUBCAT DO],

VP[AUX -, BSE, ELLIP -, COORD -].

Regel (50) verlangt also, dass *do*, welches mit den Merkmalen [+ AUX, + FIN] die finite Hilfsverbform enthält, ein Komplement der Form VP [AUX -, BSE] haben muss, d.h. ein Hauptverb in der Grundform, in diesem Fall *smile*.

Damit nun diese Struktur für den ganzen Satz gültig wird, benutzt man eine Metaregel, welche aus dem lokalen Baum eine Regel für den ganzen Satz erstellt. Sie wird METARULE SAI für *Subject Auxiliary Inversion* genannt, weil gerade die Position des Hilfsverbs vor dem Subjekt des Satzes die Metaregel nötig macht.

(51)METARULE SAI

Subject Auxiliary Inversion metarule. Applies to all non- [ELLIP +] auxiliary rules except for VP/TO. The listing of possible input SUBCAT values excludes VP/TO. SLASH propagation is done here too.

VP [+AUX, VFORM (@, NOT), ELLIP (-, @), COORD (-, @) →
H [SUBCAT (DO, MODAL_BSE, MODAL_INF, FUT, HAVE, BE),
X2 [~COMP, BEGAP (@, -)],
W.
==> S [+INV, +FIN, COMP NORM, SLASH @s] →
H [+INV],
X2 [SLASH @s],
N2 [+NOM, -PRD, SLASH NOSLASH],
W.

Die Metaregel (51) macht aus einer VP, die ein Hilfsverb und in ihrem Kopf das SUBCAT-Merkmal DO enthält, und eventuell einer anderen kompletten Phrase, z.B. einer PP, einen Satz mit der erwünschten Reihenfolge der Komponenten. Das Merkmal [+INV] zeigt, dass eine Inversion stattgefunden hat, hier die Inversion von Subjekt und Hilfsverb, damit ein Fragesatz entsteht.

Um beim Verb zu bleiben, muss noch die Regel für intransitive Verben veranschaulicht werden.

(52)IDRULE VP/INTR

"he sings", "it rains" (either NORM or IT subject)

VP [AGR N2] →
H [SUBCAT NULL].

In ID-Regel (52) wird bestimmt, dass ein intransitives Verb kein Objekt hat und dass es mit den Merkmalen der Subjekts-Nominalphrase übereinstimmt [AGR N2]. Das Merkmal [AGR] kann neben Verben auch bei Adjektiven und Artikeln auftreten und deren Übereinstimmung mit einer anderen Kategorie verlangen⁴.

⁴ Im Alvey-Parser sind die Werte von [AGR] eingeschränkt worden, was nicht ganz genau mit der Theorie von Gazdar et al. (1985) übereinstimmt. Bei Gazdar et al. (1985) beinhaltet [AGR] eine komplette Kopie aller Merkmale der zu übereinstimmenden Kategorie, im Alvey-Parser nur ein eingeschränktes Set.

Die Struktur der NP entsteht aus den folgenden Regeln.

- (53) IDRULE N2+/N2-a
a 'determinerless' N2+. Can be plural or mass (binding of values between PLU and COUNT excludes singular count and the result is indefinite.
N2 [SPEC +, DEF -, PLU @pc, KIND @k] →
H2 [SPEC -, DEF -, KIND @k, QFEAT -].
- (54) IDRULE N2-
simple N2 dominating an N1 head.
N2 [-SPEC, QFEAT -] →
H1.
- (55) IDRULE N1/N
an N with no complements.
N1 →
H [SUBCAT NULL].
- (56) N-SUFFIXES (siehe 2.3 Der Morphologieanalysator)

Im Beispielsatz *do girls smile* besteht die NP aus einem im Plural stehenden Nomen ohne Komplemente oder Artikel. Der Alvey-Parser bestimmt die Struktur des Nomens mit Regel (56) und gibt diese Information durch Regeln (54) und (55) weiter. Mit Regel (53) wird schlussendlich der Kopf der Nominalphrase bestimmt. Eine Verunsicherung könnte das Merkmal [SPEC +] in Regel (53) beinhalten. [SPEC] wurde in dieser Implementation so definiert, dass bis zum zweiten Barlevel nur einen + Wert haben kann. Der Grund dafür ist das Bedürfnis für mehrere Strukturlevel, ohne dass die phrasale Projektion von N2 frühzeitig aufgehoben würde. In Regel (53) wird diese Einschränkung jedoch aufgehoben und gegebenenfalls korrigiert. Zusätzlich wurde das Merkmal [QFEAT] geschaffen, das angibt, ob eventuell in einer [SPEC -] NP ein zusätzlicher Artikel erscheint. (Bsp. *the many books* vs. *all the books*). Dieses Merkmal wird vor allem benötigt, weil die semantische Übersetzung mit diesen Merkmalen beträchtlich variiert. Da dies aber nicht unser Thema ist, wenden wir uns dem nächsten Beispiel zu.

3.2.2 Who hits sheep?

Als generelle Bemerkung zur Behandlung von WH-Fragesätzen und Relativsätzen muss vorausgeschickt werden, dass in GPSG im Gebiet der unbegrenzten Abhängigkeiten ein Unterschied zwischen Subjekts-Fragen bzw. Subjekts-Relativen und anderen syntaktischen Funktionen gemacht werden muss.

(57) The man who saw you.
I wonder who did it.
Who did it?

(58) The man who you saw.
I wonder what he did.
Who were you talking about?

In den Sätzen in Beispiel (57) nehmen die Relativ- bzw. Interrogativpronomen die Rolle des Subjekts, in Beispiel (58) eine andere Funktion als das Subjekt ein. Die Problemstellung der unbegrenzten Abhängigkeiten der Art in Beispiel (58) wird mit dem nächsten Beispielsatz *Who do dogs eat* behandelt, denn dazu muss das Merkmal SLASH eingeführt werden. Dies ist aber in diesem Beispiel noch nicht der Fall.

Im Satz *Who hits sheep* agiert das Interrogativpronomen *who* als Subjekt. Damit die korrekte Weitergabe der Merkmale gewährleistet ist, kommt das Foot Feature Prinzip (FFP) zum Zug (siehe auch Abschnitt 1.2.5 Das Foot Feature Prinzip (FFP)). [Q] für engl. Question ist ein Merkmal, das nicht via Head Feature Principle weitergegeben werden kann, da es in einem Tochterknoten unterhalb des Mutterknotens auftaucht und nach oben weitergegeben werden muss.

Die angestrebte Struktur dieses Satzes ist NP (WH-Pronomen) + VP, und der Alvey-Parser leitet genau diese Struktur ab:

unterscheidet zwischen den verschiedenen Arten von unbegrenzten Abhängigkeiten, namentlich zwischen WH-Fragen [UB Q], Relativsätzen [UB R] oder keiner solchen Konstruktion [UB NO]. [WH +] kodiert einen Konstituenten als Interrogativpronomen, und gibt diese Spezifikation auch an die umliegende Umgebung weiter. [EVER] ist in diesem Satz nicht relevant, steht aber für wh-Wörter wie *whichever*, *whoever*, etc. und ist hier notiert, weil auch solche Wörter eine unbegrenzte Abhängigkeit ausmachen können (z.B. *Whatever Kim does annoys Lee*). Es handelt sich also nicht nur um eine morphologische Information. Regel (60) erlaubt ein Non-Nominatives Subjekt. Diese Formulierung mag verwirrend klingen, meint aber in diesem Fall ein Pronomen, das morphologisch nicht nur als Nominativ erkannt wird. Es könnte auch für einen Relativsatz oder eine andere Struktur, die nicht nur aus einer klassischen Nominalphrase besteht, stehen. Regel (61) gibt genauere Angaben zum Pronomen. Die restliche Satzstruktur besteht schliesslich aus Regel (62), die transitive Verben beschreibt, und der Nominalphrase, die nach schon bekannten Regeln aufgebaut ist.

3.2.3 Who do dogs eat?

Anders als unter 3.2.3 kann dieser Satz nicht mehr als NP (Pronomen) + VP analysiert werden, denn die Subjektsposition ist von *dogs* besetzt, was wiederum die Position des Objekts offenlässt. Englisch ist eine Sprache, die das Objekt nach dem Subjekt und dem Prädikat plaziert, und dies ist im Satz *Who do dogs eat* nicht mehr der Fall. Um diese Lücke oder *Gap* zu füllen, verwendet man das Merkmal SLASH. SLASH ist ein Merkmal, das sowohl dem HFP als auch dem FFP untersteht. Dies kommt daher, dass SLASH in nicht-lexikalischen ID-Regeln von einem Mutter- zu einem Tochterknoten übergeben wird (HFP), in lexikalischen ID-Regeln aber nicht zu den Kopfmerkmalen gehören kann und mit dem FFP geholt werden muss. Die Konsequenz davon ist, dass die einzigen Kategorien, welche das Merkmal SLASH einführen können (sie sind im Alvey-Parser markiert mit [NULL +]), Argumentspositionen, also Schwesterknoten eines lexikalischen Kopfes, sind. So wird das Problem der unbegrenzten Abhängigkeiten elegant gelöst.

Die Weitergabe des Merkmals SLASH erfolgt durch Metarules. So wird eine Struktur, die eine Lücke hat, durch die passende Metaregel nach oben propagiert. Ein Nachteil davon ist, dass anhand der Metaregeln eine grosse Zahl von neuen ID-Regeln kreiert wird. Die vierte Version des Alvey-Parsers, mit der wir arbeiten, hat

dieses Problem gelöst, in dem sie das Merkmal [SLASH] für alle nicht-lexikalischen Kategorien mit einer Serie von Propagation Rules und Metarules spezifiziert hat. Dieser Schritt, der ein wenig von der Theorie von Gazdar et al. (1985) abweicht, reduziert die Anzahl Grammatikregeln um ungefähr die Hälfte (Version 2: ~1'500 Grammatikregeln vs. Version 3: 782 Grammatikregeln).

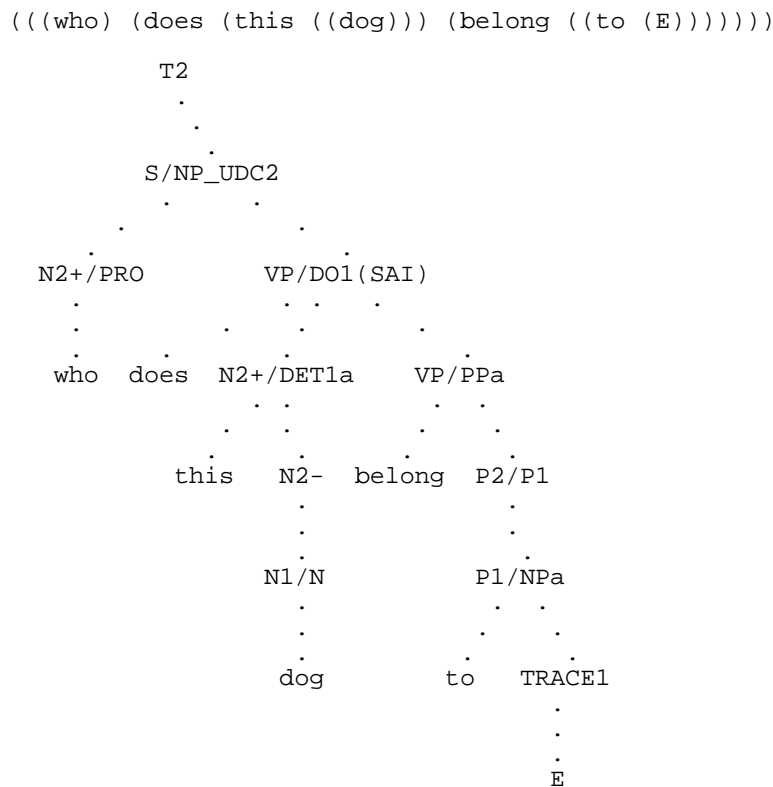
Die von Alvey errechnete Struktur sieht folgendermassen aus:

T2 wurde mit Regel (59) in Beispiel 3.2.2 schon erwähnt und bedeutet, dass ein Satz als Frage erkannt wird. Mit Regel (63) wird diese Frage weiter spezifiziert. [UDC+] bedeutet, dass eine Unbounded Dependency Construction vorhanden ist. Die Satzstruktur sieht mit Regel (63) so aus, dass sie eine Fragestellung mit Inversion und unbegrenzter Abhängigkeit beinhaltet. Daraus folgert, dass die Nominalphrase nicht mit einer „dummy form“ wie *it* oder *there* beginnt ([NFORM NORM]), sondern ein Fragepronomen [Q+] vorangeht. Die Struktur dieses Pronomens ist mit Regel (64) festgelegt, die zusätzlich unterscheidet, ob es modifizierbare oder partitive Pronomen sind.

Mit diesen drei Regeln wird die gesamte Topstruktur des Satzes mit einer unbegrenzten Abhängigkeit festgelegt. Es fehlt noch der Aufbau der VP/NP-Struktur, die zum grössten Teil schon in Beispiel *Do girls smile* behandelt worden ist, denn es handelt sich wieder um ein Subjekt-Hilfsverb-Inversion (IDRULE VP/DO1(SAI)). Die entscheidenden Merkmale für die Problemstellung dieses Satzteiles finden sich aber in Regeln (65) und (66), die das Merkmal TRACE beschreiben. Regel (65), die Regel für ein transitives Verb, fordert ein Objekt, das aber in dieser Phrase nicht gefunden werden kann. Regel (66) stellt nun das Merkmal TRACE zur Verfügung und setzt den ganzen Prozess in Gang, in dem das Merkmal bis zu den oben erwähnten Top-Strukturen vordringt. Erwähnenswert an Regel (66) ist, dass sie sehr viele @ enthält, die sogenannte ungebundene Variablen darstellen, die später unifiziert werden müssen. Ebenso ist am Ende der Regel das Merkmal [NULL+] ersichtlich, das fordert, dass an dieser Stelle ein solcher Lexikoneintrag vorhanden sei.

3.2.4 Who does this dog belong to?

In diesem Beispielsatz wird die gleiche Struktur wie im vorherigen gesucht. Neu ist, dass das Verb anstelle einer Nominalphrase eine Präpositionalphrase verlangt.



(67) IDRULE N2+/DET1a
"the", "a", "this dog". A number of different specifiers attach under N2[+SPEC] ie Det, POSS NP, A2[+QUA] – in complementary distribution. This rule does Det and only [PRD -]. [PRD +] must be indefinite and the semantics is different.

N2[+SPEC, PRD -] →
 DetN[AGR N2, WH NO, UB NO, EVER NO],
 H2[-SPEC].

(68) IDRULE N2-
simple N2 dominating an N1 head.

N2[-SPEC, QFEAT -] →
 H1.

(69) IDRULE N1/N
an N with no complements

N1 →
 H[SUBCAT NULL].

(70) IDRULE VP/PPa

"kim looks at sandy" - [SUBTYPE PVERB] The PP is [PRD -] and in effect has the same translation as its NP object. The preposition combines with the verb to make a complex predicate. The semantics happens to give the right translation whether the PP is [SLASH NOSLASH] or [SLASH X2] so there is no need for the usual conditions.

VP →

H[SUBCAT PP, PFORM @pf, SUBTYPE PVERB],
P2[PRD -, PFORM @pf].

(71) IDRULE P1/NPa

*Separate rule for possessive 'of' PP means that -POSS has to appear on P1. To prevent PP[UB R, WH -] from occurring - for example in "*the table on that I put the book, *the man a book about that I read" - this rule has to be split into two. This is a non-wh version; P1/NPb is a [WH +] version.*

P1[-POSS, -GER, WH NO, UB NO, EVER NO] →

H[SUBCAT NP],

N2[-POSS, NFORM NORM, PRD -, ADV -].

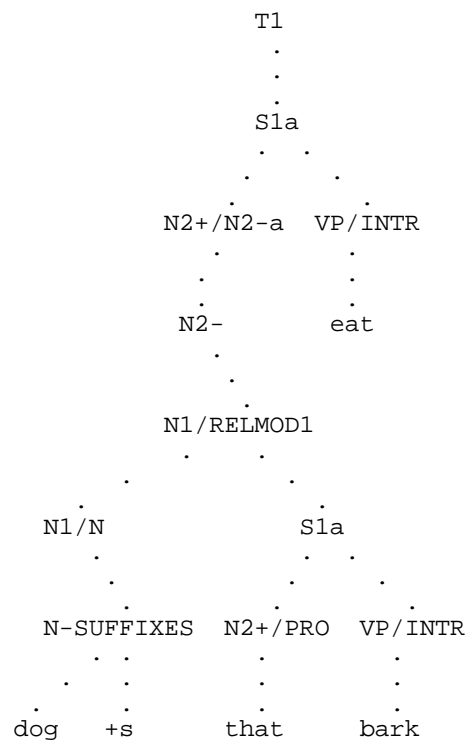
Regeln (67) bis (69) illustrieren eine Nominalphrase mit einem Demonstrativartikel wie in *this dog*. Regel (70) beschreibt Verben, die eine Präpositionalphrase fordern, mit den Merkmalen [SUBCAT PP] und [SUBTYPE PVERB]. Damit die Präposition im Verb *to belong to somebody* nicht mit einem Possessivpronomen verwechselt wird, benötigt man Regel (71), die die Verbstruktur als solche erkennt. Schlussendlich ist das Merkmal SLASH, das im Abschnitt 3.2.3 behandelt wurde, entscheidend.

Im Satz *Who does this dog belong to* kommen schon eine grössere Anzahl syntaktische Schwierigkeiten zusammen. Der Alvey-Parser meistert diese hervorragend, denn für diesen Satz erhalten wir nur eine, nämlich die richtige, Syntaxstruktur. Bei der Behandlung von solchen Strukturen ist eine Stärke von GPSG und dem Alvey-Parser ersichtlich.

3.2.5 Dogs that bark eat.

Im Gegensatz zu den vorherigen Beispielen mit Fragesätzen behandelt dieser Abschnitt eine Nebensatzkonstruktion, oder genauer, ein Aussagesatz mit eingeschobenem Relativsatz. Die Schwierigkeit dabei ist, dass der Parser die Übereinstimmung des Subjekts und des Prädikats, die durch die Relativklausel getrennt sind, erkennen muss.

(((((dog +s)) ((that) (bark)))) (eat)))



(72) IDRULE S1a

split up ordinary S rules to allow for non-nominative subjects. This one = finite S.
 S[COMP NORM, -INV, +FIN, ELLIP -, COORD -, UDC -] →
 N2[+NOM, -PRD],
 H2[-SUBJ, AGR N2, ELLIP -, COORD -].

(73) IDRULE N1/RELMOD1

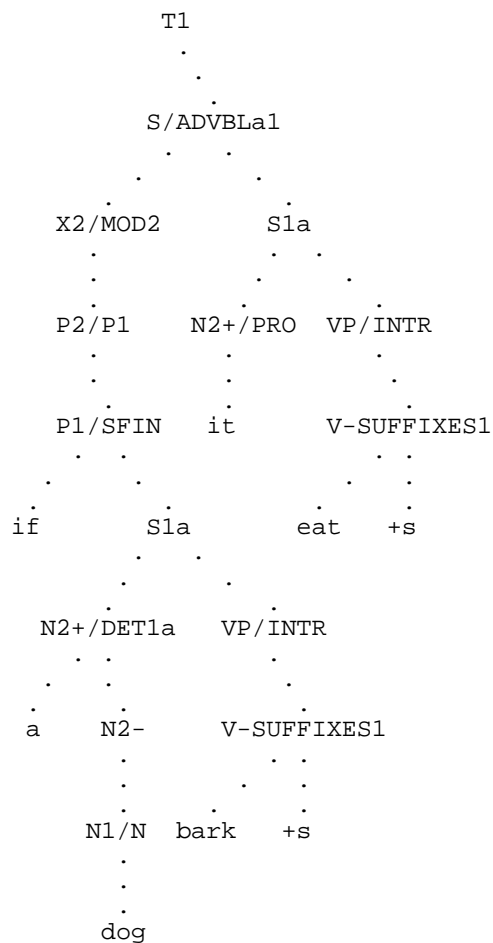
"sheep who/that attacks fido", "sheep who/that fido attacks". The N1 defaults to [DEMON -] so a separate rule is needed for demonstrative pronouns.
 N1[MOD POST] →
 H1[WH NO, UB NO, EVER NO],
 S[+R, -EVER, WH @wh].

Als Ergänzung zum Kommentar zur IDRULE S1a, der schon in Abschnitt 3.2.2 gemacht wurde, kann hier gesagt werden, dass das „Non-Nominative Subjekt“ in

diesem Fall aus einem im Plural stehenden Nomen und einem Relativsatz besteht. Die Bezeichnung im Kommentar von Regel (72) ist nach wie vor verwirrend. Entscheidend für das korrekte Parsen des Satzes *dogs that bark eat* ist aber Regel (73), in der steht, dass N1 postmodifiziert ist, aus einem Kopf besteht, der keine Merkmale einer Fragestruktur oder unbegrenzter Abhängigkeit aufweist und dem dann ein Relativsatz [+R] folgt. So beschreibt Regel (73) eigentlich die gesamte Struktur eines eingeschobenen Relativsatzes. Erwähnenswert zu Regel (73) ist auch, dass im Kommentar dazu die Default-Werte für das Merkmal [DEMON], das besagt, ob ein Pronomen Demonstrativ ist oder nicht, angegeben werden. Da diese auf [DEMON-] eingestellt sind, muss für den Fall, dass es sich doch um ein Demonstrativpronomen handelt, eine andere Regel erstellt werden.

3.2.6 If a dog barks it eats.

(((((if ((a ((dog)) ((bark +s)))))) ((it) ((eat +s))))))



(74a)IDRULE S/ADVBLa1

allows for sentence initial adverbials. The ADVP can be expanded as an AP[+ADV] or as a PP or by the MOD/COORD rules. -INV version.*

S[WH @a, UB @b, EVER @c, UDC +] →
ADVP[WH @a, UB @b, EVER @c],
S[H +, INV -, ELLIP -, COORD -].

(74b)IDRULE ELL/S/ADVBLa1

[ELLIP +] version of S/ADVBLa1

S[WH @a, UB @b, EVER @c, UDC +] →
ADVP[WH @a, UB @b, EVER @c],
S[H +, INV -, ELLIP +, COORD +].

(75) IDRULE X2/MOD2

an adverbial modifier can be a PP.

X2[+ADV, CONJ NULL, WH @a, UB @b, EVER @c, PRD +, QUA -, LOC @1]
→
P2[PFORM NORM, NEG -, WH @a, UB @b, EVER @c, PRD +, GERUND @g,
LOC @1].

(76) IDRULE P2/P1

P2 →
H1 : 1.

(77) IDRULE P1/SFIN

“before he went to bed”

P1[-POSS, -GER] →
H[SUBCAT SFIN],
S[+FIN, COMP NORM, -INV].

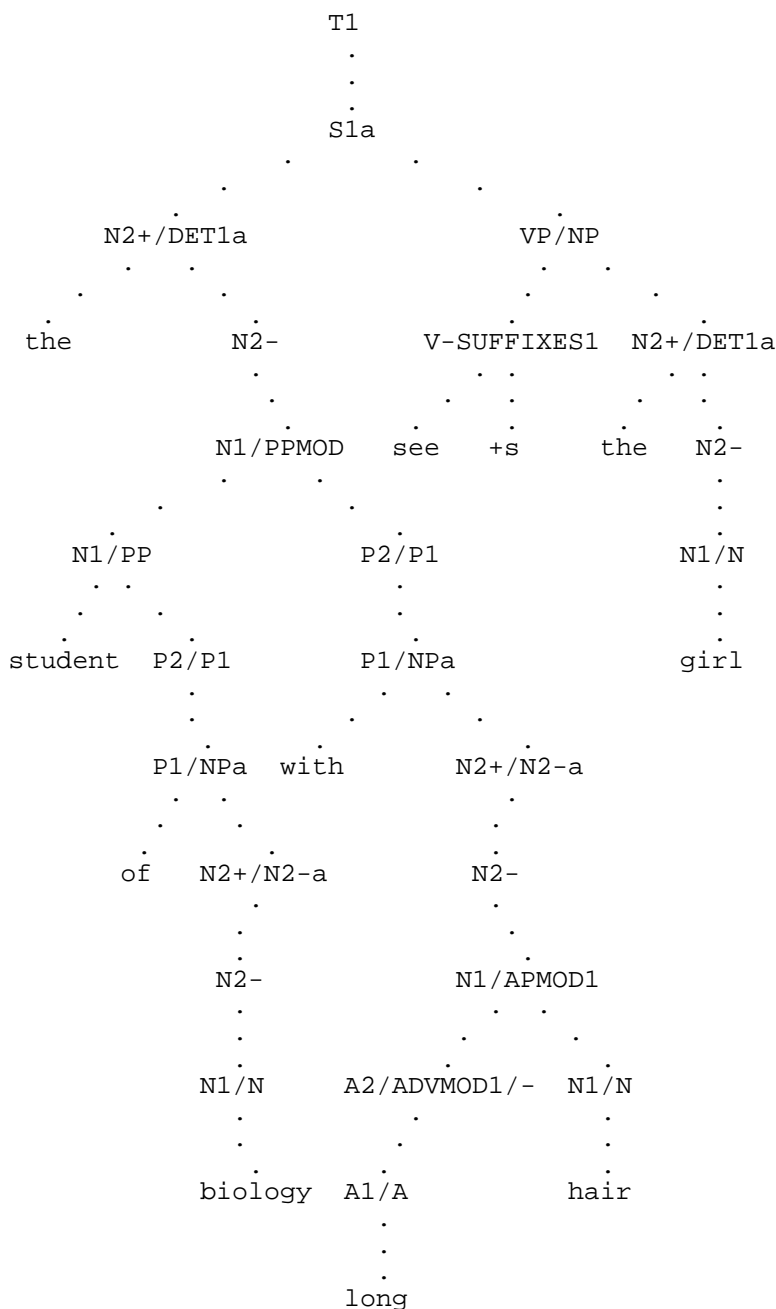
3.3 Unterscheidung Komplement / Adjunkt

3.3.1 The student of biology with long hair sees the girl

Bei diesem Beispielsatz liefert der Alvey-Parser zehn Syntaxstrukturen. Lassen wir aber den Satz *The student of biology with long hair eats the dog* parsen, so erhalten wir 58 Syntaxstrukturen. Wie schon in der Einleitung zu diesem Teil erwähnt, vermuten wir in einem solchen Fall, dass gewisse Inkonsistenzen im Lexikon vorliegen.

Die beiden Prepositionalphrasen *of biology* und *with long hair* sind Adjunkte zum Studenten, wobei *of biology* dem Nomen nähersteht als *with long hair*. Es ist nun wichtig, die richtige Hierarchie der einzelnen Satzglieder zu bestimmen. Der Alvey-Parser hat dies in einem von zehn Fällen getan:

((the ((student ((of ((biology)))))) ((with (((long)
 (hair)))))))) ((see +s) (the ((girl))))))



(78) IDRULE N1/PPMOD
 "man with the umbrella", "a book of fido's"
 N1[MOD POST] →
 H1,
 P2[-GER, PFORM NORM, PRD +, MODTYPE NML, PRO -].

(79) IDRULE N1/PP
 "picture of kim", "discussion about kim"
 N1 →
 H[SUBCAT PP, PFORM @pf],

P2[PFORM @pf, -POSS, PRD +].

(80) IDRULE P2/P1

P2 →
H1.

(81) IDRULE P1/NPa

*Separate rule for possessive 'of' PP means that -POSS has to appear on P1. To prevent PP[UB R, WH -] from occurring - for example in "*the table on that I put the book, *the man a book about that I read" - this rule has to be split into two. This is a non-wh version; P1/NPb is a [WH +] version.*

P1[-POSS, -GER, WH NO, UB NO, EVER NO] →
H[SUBCAT NP],
N2[-POSS, NFORM NORM, PRD -, ADV -].

(82) IDRULE N1/APMOD1

"busy man". The -PRD, DISTR ATT restriction on the A2 prevents adjectives with complements matching. The feature MOD on the N1 cuts down on the number of parses by making premodifiers attach lower than postmodifiers.

N1[MOD PRE] →
A2[-PRD, DISTR ATT, -QUA],
H1[MOD NONE, PRO -].

(83) IDRULE A2/ADVMOD1

"exceptionally clever", "nearly all".
A2 →
(A2[+ADV, -PRD, -QUA]),
H1.

(84) IDRULE A1/A

an A with no complements.
A1 →
H[SUBCAT NULL].

Regeln (78) bis (81) beschreiben die Abhängigkeit des Komplements vom Nomen. Es wird spezifiziert, dass eine Postmodifikation stattfindet (78) und dass die Postmodifikation eine Präpositionalphrase ist (79). Wichtig bei Regel (78) ist, dass sie weitere Postmodifikation in irgendeiner Form nicht ausschliesst, denn genau an diesem Knoten wird das nächste Komplement angefügt. Weiter wird die Struktur der PP mit Regel (81) als Possessiv deklariert und dies mit Regel (80) unverändert ein Satzlevel weitergereicht.

Dasselbe geschieht noch einmal anhand einer fast gleichen Struktur für die PP *with long hair*. Der einzige Unterschied ist, dass hier die Regel (79) nicht zum Zug kommt, weil die PP nicht direkt an eine NP angehängt wird. Der Knotenpunkt, an dem die PP

with long hair anschliesst, ist Regel (78), also ein Knoten oberhalb des vorherigen Anschlusses. Dieser Aufbau ermöglicht eine NP, die beliebig erweiterbar wäre.

Bisher noch nie besprochen wurde der Aufbau einer Phrase mit einem Adjektiv wie in Regeln (82) bis (84) beschrieben. Regel (84) erkennt ein Adjektiv ohne Komplemente. Regel (83) steht ein Satzlevel höher, und würde dem Adjektiv erlauben, ein Adverb als Premodifikation anzunehmen. In Regel (82), wo die AP und die NP zusammenkommen, muss die Übereinstimmung dieser beiden kontrolliert werden. Dies tun die Merkmale [PRD] für prädikativ und [DISTR] für Distribution (Verteilung). Sie treten immer in Kombination auf, da sie zusammen die Stellung der verschiedenen Adjektive einschränken können. [DISTR] unterscheidet zwischen drei möglichen Positionen für die AP: Prenominale Modifikation, Postnominale Modifikation und Prädikative Position. Zusammen mit [PRD] entstehen also drei mögliche Kategorien für Adjektive.

1. prenominal [DISTR ATT, PRD -]
2. postnominal [DISTR PRD, PRD +]
3. prädikativ [DISTR @, PRD +]

Damit kann ein Adjektiv genauestens kontrolliert werden. Die PP *with long hair* gehört zum ersten, prenominalen Fall. Somit ist die Struktur einer Adjektivphrase geklärt.

3.3.2 The student gives a bone to the dog every day.

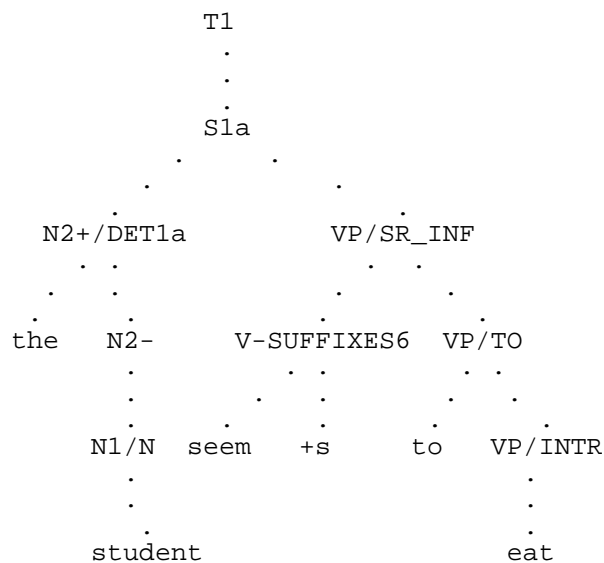
Dieser Satz ergibt kein Resultat. Der Alvey-Parser stürzt nach einer gewissen Rechenzeit mit einer Fehlermeldung ab. Wir vermuten, dass das Satzadverb *every day* der Grund für die Probleme ist.

3.4 Raising-Konstruktionen, Infinitive, Hilfsverben

3.4.1 The student seems to eat.

In Sätzen mit Raising-Verben (z.B. *seem*, *believe*, etc.) wird das Subjekt des untergeordneten Satzes in den übergeordneten gehoben. Dabei beeinflusst das untergeordnete Verb das übergeordnete Subjekt, wobei eine unbeschränkte Abhängigkeit entsteht, weil dies eine nicht-lokale Bedingung über Syntaxbäume beinhaltet.

((the ((student))) ((seem +s) (to (eat))))



(85) IDRULE VP/TO

to + base VP - infinitival VP of the kind which occurs in infinitival S and in numerous control constructions.

VP[+AUX, VFORM TO, -FIN, ELLIP -, COORD -] →

H[SUBCAT TO],

VP[BSE, ELLIP -, COORD -].

(86) IDRULE ELL/VP/TO

[ELLIP +] version of VP/TO

VP[+AUX, VFORM TO, -FIN, ELLIP @e, COORD +] →

H[SUBCAT TO],

VP[BSE, ELLIP +, COORD +].

(87) IDRULE VP/SR_INF

"appear to be crazy" (subject raising).

VP →

H[SUBCAT SC_INF, SUBTYPE RAIS],

VP[TO].

ID-Regel (87) beschreibt das Raising-Konstrukt. Dem Verb des untergeordneten Satzes, das im Infinitiv mit *to* stehen muss, wird die Unterkategorie RAIS für Raising zugeordnet. Damit kann die unbeschränkte Abhängigkeit elegant gelöst werden. Wie schon erwähnt wird in GPSG *to* als nicht-finites Hilfsverb behandelt und benötigt deshalb eine separate Regel (ID-Regel (86)), die die Arten der *to*-Konstruktionen beschreibt.

VP[EN, PRD -, ELLIP -, COORD -].

(89) IDRULE VP/MOD2

allows AP (but not PP) modifiers to precede non-finite, non-predicative VPs: "has frequently happened, may frequently have happened, will certainly happen" etc.

VP[-FIN, -PRD, ELLIP -, COORD -] →

A2[+ADV],

VP[H +, ELLIP -, COORD -].

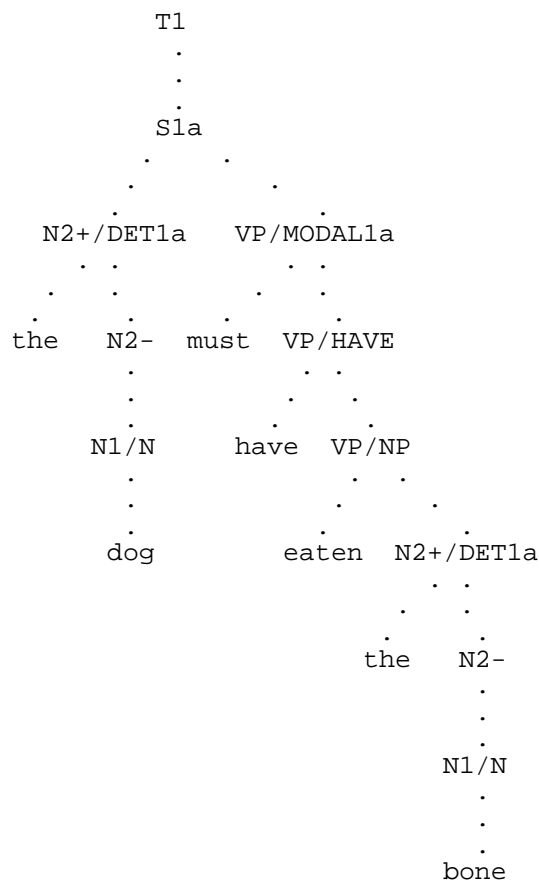
Neu werden die Regeln für das Hilfsverb *have* und seine Formen (88) und für die Position eines Adverbs in einer Hilfsverbskonstruktion (89) eingeführt.

3.4.5 The dog must have eaten the bone.

Die Behandlung von Hilfsverben ist in der Generativen Grammatik seit langem ein kontroverser Punkt. Man ist sich nicht einig, zu welcher syntaktischen Kategorie sie eigentlich gehören, denn man kann sie als separate Kategorie oder als den Verben zugehörig betrachten. In GPSG wird diese Kategorisierung weniger wichtig, weil GPSG mit Merkmalstrukturen arbeitet und so flexible Merkmale, die nicht unbedingt einer Kategorie angehörig sein müssen, verteilen kann. Man kann in GPSG sowohl von der „Klasse aller Verben mit Hilfsverben“ als auch von der „Klasse der Hilfsverben“ sprechen.

In GPSG und im Alvey-Parser erhalten Hilfsverbsequenzen eine rekursive Struktur, wobei das Hilfsverb jeweils als Kopf seiner VP behandelt wird und sein Komplement eine andere VP darstellt (siehe dazu auch 1.3.3 Hilfsverbkonstruktionen). Zu den Hilfsverben werden auch die Modalverben gezählt, die in diesem Beispiel eine tragende Rolle spielen. Hilfsverben werden mit dem Merkmal [AUX +] gekennzeichnet und können dadurch bei ihrem Auftreten als solche Konstruktion erkannt werden.

((the ((dog))) (must (have (eaten (the ((bone))))))))



(90) IDRULE VP/MODAL1a

"can dance", "may be dancing" etc. For (SUBTYPE NONE) entries - ones which are ambiguous between epistemic and deontic readings. The ambiguity is only available in the case where the VP is AGR N2[NFORM NORM]. In other cases, only the epistemic reading is possible.

VP[+AUX, +FIN] →

H[SUBCAT MODAL_BSE, SUBTYPE NONE],

VP[BSE].

Die einzige Regel, die noch nicht behandelt wurde, ist die Regel über Modalverben. Sie werden mit dem SUBCAT-Merkmal MODAL gekennzeichnet und verlangen eine VP in ihrer Grundform.

4. Konklusion

Die GDE-Tools bieten der LinguistIn alle Möglichkeiten zur Entwicklung einer Grammatik. Die Modulbauweise der Alvey-Tools lässt eine unabhängige Betrachtung von Parsing, Generierung, Morphologie oder Semantik zu. Trotz der beinahe komplett konfigurierbaren Entwicklungsumgebung sind wir von den gebotenen Parse-Resultaten enttäuscht. Viele syntaktische Konstruktionen lassen sich auf sehr elegante Weise parsen. Wie aber aus unseren Beispielsätzen hervorgeht, erhielten wir bei den vorgegebenen Beispielsätzen oft mangelhafte oder komplett unzulängliche Resultate. Liegt dies nun am Parser oder an GPSG? Dies ist eine Frage, die wir mit dem heutigen Wissen nicht ausführend beantworten können. Wir konnten –wie schon erwähnt- feststellen, dass der Parser häufig dann falsche Resultate liefert, wenn er einige Wörter der Satzes, den er parst, nicht kennt. Dies spricht eindeutig für ein zu wenig umfangreiches Lexikon, obschon die Alvey Natural Language Tools für ihre Ausführlichkeit bekannt sind. Die allgemeine Performanz der Alvey Natural Language Tools ist für uns momentan schwierig zu beurteilen, da wir zu diesem Zeitpunkt noch über keine Vergleichsobjekte verfügen.

Um zu einer abschliessenden Schlussfolgerung zu gelangen, müssen wir feststellen, dass die natürliche Sprache (immer noch) ungelöste Probleme für die ComputerlinguistInnen bietet. Da die Entwicklung von GPSG schon einige Jahre her ist und schon wieder als veraltet gilt, liegt unsere Hoffnung in den neueren Syntaxtheorien und deren Implementationen.

5. Bibliographie

Bennett, Paul. 1995. *A Course in Generalized Phrase Structure Grammar*. London: UCL Press.

Borsley, Robert D. 1999. *Syntactic Theory. A Unified Approach*. 2nd ed. London: Arnold.

Gazdar, Gerald und Klein, Ewan und Pullum, Geoffrey und Sag, Ivan. 1985. *Generalized Phrase Structure Grammar*. Cambridge, MA: Harvard University Press.

Grover, Claire und Carroll, John und Briscoe, Ted. 2000. *The Alvey Natural Language Tools Grammar (4th release)*. Cambridge: Lynxvale WCIU Programs.

Volk, Martin. 1988. „Parsing German with GPSG: The Problem of Separable-Prefix Verbs“. <http://www.fi.unizh.ch/CL/volk/SyntaxVorl/GPSG.thesis.html>.

Volk, Martin. 1999. Vorlesung „Formale Grammatiken und Syntaxanalyse“
Vorlesungen 4. – 6. <http://www.ifi.unizh.ch/CL/volk/SyntaxVorl4.GPSG.html>,
<http://www.ifi.unizh.ch/CL/volk/SyntaxVorl5.GPSG.html>,
<http://www.ifi.unizh.ch/CL/volk/SyntaxVorl6a.GPSG.html>

6. Anhang

ANHANG A Liste der Merkmale und Regeln

Merkmale und Merkmalswerte

CAT = { *C* | *C* is a category }

<i>feature</i>	<i>value range</i>
CASE	{ACC, NOM}
COMP	{ <i>for, that, whether, if</i> , NIL}
CONJ	{ <i>and, both, but, neither, either, nor, or</i> , NIL}
GER	{+, -}
NEG	{+, -}
NFORM	{ <i>there, it</i> , NORM}
NULL	{+, -}
POSS	{+, -}
REMOR	{RECP, REFL}
WHMOR	{R, Q, FR, EX}

Kopfmerkmale

<i>feature</i>	<i>value range</i>
AGR	<i>CAT</i>
ADV	{+, -}
AUX	{+, -}
BAR	{0, 1, 2}
INV	{+, -}
LOC	{+, -}
N	{+, -}
PAST	{+, -}
PER	{1, 2, 3}
PFORM	{ <i>to, by, for, ...</i> }
PLU	{+, -}
PRD	{+, -}
SLASH	<i>CAT</i>
SUBCAT	{1, ..., <i>n</i> }
	{ <i>for, that, and, both, ...</i> }
SUBJ	{+, -}
V	{+, -}
VFORM	{BSE, FIN, INF, PAS, PRP, PSP}

Fussmerkmale

<i>feature</i>	<i>value range</i>
RE	<i>CAT</i>
SLASH	<i>CAT</i>
WH	<i>CAT</i>

Feature co-occurrence restrictions

FCR 1:	[+ INV] \supset [+ AUX, FIN]
FCR 2:	[VFORM] \supset [+V, -N]

FCR 3:	[NFORM] \supset [-V, +N]
FCR 4:	[PFORM] \supset [-V, -N]
FCR 5:	[PAST] \supset [FIN, -SUBJ]
FCR 6:	[SUBCAT] \supset \sim [SLASH]
FCR 7:	[BAR 0] \equiv [N] & M & [SUBCAT]
FCR 8:	[BAR 1] \supset \sim [SUBCAT]
FCR 9:	[BAR 2] \supset \sim [SUBCAT]
FCR 10:	[+ INV, BAR 2] \supset [+ SUBJ]
FCR 11:	[+ SUBJ] \supset [+ V, - N, BAR 2]
FCR 12:	[AGR] \supset [-N, + V]
FCR 13:	[FIN, AGR NP] \supset [AGR NP[NOM]]
FCR 14:	([+ PRD] & [VFORM]) \supset ([PAS] V [PRP])
FCR 15:	[COMP] \equiv [+SUBJ]
FCR 16:	[WH, + SUBJ] \supset [COMP NIL]
FCR 17:	[COMP <i>that</i>] \supset ([FIN] V [BSE])
FCR 18:	[COMP <i>for</i>] \supset [INF]
FCR 19:	[+NULL] \supset [SLASH]
FCR 20:	\sim ([SLASH] & [WH])
FCR 21:	A' \supset \sim NH
FCR 22:	VP \supset \sim [WH]

Feature specification defaults

FSD 1:	[-INV]
FSD 2:	\sim [CONJ]
FSD 3:	\sim [NULL]
FSD 4:	\sim [NOM]
FSD 5:	[PFORM] \supset [BAR 0]
FSD 6:	[+ADV] \supset [BAR 0]
FSD 7:	[BAR 0] \supset \sim [PAS]
FSD 8:	[NFORM] \supset [NFORM NORM]
FSD 9:	[INF, +SUBJ] \supset [COMP <i>for</i>]
FSD10:	[+ N, - V, BAR 2] \equiv [ACC]
FSD11:	[+ V, BAR 0] \supset [AGR NP[NFORM NORM]]

Immediate dominance rules (ID-Regeln)

VP \rightarrow H[1]	(<i>die</i>)
VP \rightarrow H[2], NP	(<i>love</i>)
VP \rightarrow H[3], NP, PP[<i>to</i>]	(<i>give</i>)
VP \rightarrow H[4], NP, PP[<i>for</i>]	(<i>buy</i>)
VP \rightarrow H[5], NP, NP	(<i>spare</i>)
VP \rightarrow H[6], NP, PP[+ LOC]	(<i>put</i>)
VP[+ AUX] \rightarrow H[7], XP[+ PRD]	(<i>be</i>)
VP \rightarrow H[8], NP, S[FIN]	(<i>persuade</i>)
VP \rightarrow H[9], (PP[<i>to</i>]), S[FIN]	(<i>concede</i>)
VP \rightarrow H[10], S[BSE]	(<i>prefer</i>)
VP \rightarrow H[11], (PP[<i>of</i>]), S[BSE]	(<i>require</i>)
VP[INF, +AUX] \rightarrow H[12], VP[BSE]	(<i>to</i>)
VP \rightarrow H[13], VP[INF]	(<i>tend</i>)
VP \rightarrow H[14], VI[INF, +NORM]	(<i>prefer</i>)
VP \rightarrow H[15], VP[INF, +NORM]	(<i>try</i>)
VP \rightarrow H[16], (PP[<i>to</i>]), VP[INF]	(<i>seem</i>)
VP \rightarrow H[17], NP, VP[INF]	(<i>believe</i>)
VP \rightarrow H[18], NP, VP[INF, +NORM]	(<i>persuade</i>)
VP \rightarrow H[19], (NP), VP[INF, +NORM]	(<i>promise</i>)
VP[AGR S] \rightarrow H[20], NP	(<i>bother</i>)
VP[+ <i>it</i>] \rightarrow H[21], (PP[<i>to</i>]), S[FIN]	(<i>seem</i>)
VP[AGR NP[<i>there</i> , α PLU]] \rightarrow H[22], NP[α PLU]	(<i>be</i>)
AP \rightarrow ({<SUBCAT, 23>}), H ¹	

A ¹	→ H[24], PP[<i>about</i>]	(<i>angry</i>)
A ¹	[AGR S] → H[25], PP[<i>to</i>]	(<i>apparent</i>)
A ¹	→ H[26], S[FIN]	(<i>afraid</i>)
A ¹	→ H[27], S[BSE]	(<i>insistent</i>)
A ¹	→ H[28], VP[INF]	(<i>likely</i>)
A ¹	→ H[29], V ² [INF, +NORM]	(<i>eager</i>)
N ¹	→ H[30]	(<i>death</i>)
N ¹	→ H[31], PP[<i>with</i>], PP[<i>about</i>]	(<i>argument</i>)
N ¹	→ H[32], S[COMP <i>that</i>]	(<i>belief</i>)
N ¹	→ H[331], S[BSE, COMP <i>that</i>]	(<i>request</i>)
N ¹	→ H[34], V ² [INF]	(<i>plan</i>)
N ¹	→ H[35], PP[<i>of</i>]	(<i>love</i>)
N ¹	→ H[36], PP[<i>of</i>], PP[<i>to</i>]	(<i>gift</i>)
N ¹	→ H[37], PP[<i>of</i> , GER]	(<i>dislike</i>)
P ¹	→ H[38], NP	(<i>in</i>)
P ¹	→ H[39], PP[<i>of</i>]	(<i>out</i>)
VP	→ H[401], S[FIN]	(<i>believe</i>)
P ¹ [+POSS]	→ H[41], NP[+POSS]	(<i>of</i>)
A ¹	→ H[42], V ² [INFI/NP[- NOM]	(<i>easy</i>)
VP	→ H[43], S[+ Q]	(<i>inquire</i>)
VP[+ <i>if</i>]	→ H[44], NP, S[+ R]	(<i>be</i>)
VP[+ <i>if</i>]	→ H[44], X ² , S[FIN]/ X ²	(<i>be</i>)
VP	→ H[45], PP[<i>of</i>]	(<i>approve</i>)
VP[+ AUX]	→ H[46], VP[- AUX, BSE]	(<i>do</i>)
VP	→ H[47], PP[<i>to</i>], PP[<i>about</i>]	(<i>talk</i>)
VP	→ H[48], H[CONJ <i>and</i>]	(<i>go</i>)

Nonlexical ID rules

S	→ X ² , H[-SUBJ]
S[COMP α]	→ {[SUBCAT α]}, H[COMP NIL]
where α is in { <i>that, for, whether, if</i> }	
S	→ X ² , H/ X ²
VP	→ H, ADVP
NP	→ NP[+POSS], H ¹
N ¹	→ H, PP[+POSS]
N ¹	→ B, S[+R]
AP	→ (A ² [+ADV]), H,
X[CONJ NIL]	→ H
X[CONJ α]	→ {[SUBCAT α]}, H

Iterating coordination scheme (CS⁺)

X	→ H[CONJ α_0], H[CONJ α_1] ⁺
where α is in {< <i>and</i> , NIL>, <NIL, <i>and</i> >, < <i>neither</i> , <i>nor</i> >, < <i>or</i> , NIL>, <NIL, <i>or</i> >}	

Binary coordination schema (CS²)

X	→ H[CONJ α_0], H[CONJ α_1]
where α is in f < <i>both</i> , <i>and</i> >, < <i>either</i> , <i>or</i> >, <NIL, <i>but</i> >}	

Linear precedence statements

[SUBCAT] < ~[SUBCAT]

$[+N] < P^2 < V^2$
 $[\text{CONJ } \alpha_0] < [\text{CONJ } \alpha_1]$
where α_0 is in $\{\text{both, either, neither, NIL}\}$
and α_1 is in $\{\text{and, but, nor, or}\}$.

Metarules

Passive Metarule

$VP \rightarrow W, NP$
 \rightarrow
 $VP[\text{PAS}] \rightarrow W, (\text{PP}[\text{by}])$

'Subject-Aux Inversion'(SAI) Metarule

$V^2 [- \text{SUBJ}] \rightarrow W$
 \rightarrow
 $V^2 [+ \text{INV}, + \text{SUBJ}], W, NP$

Extraposition Metarule

$X^2 [\text{AGR S}] \rightarrow W$
 \rightarrow
 $X^2 [\text{AGR NP}[\text{it}]] \rightarrow W, S$

Complement Omission Metarule

$[+N, \text{BAR } 1] \rightarrow H, W$
 \rightarrow
 $[+N, \text{BAR } 1] \rightarrow H$

Slash Termination Metarule 1 (STM1)

$X \rightarrow W, X^2$
 \rightarrow
 $X \rightarrow W, X^2 [+ \text{NULL}]$

Slash Termination Metarule 2 (STM2)

$X \rightarrow W, V^2 [+ \text{SUBJ}, \text{FIN}]$
 \rightarrow
 $X/\text{NP} \rightarrow W, V^2 [- \text{SUBJ}]$