

Abstract

In generative grammar the competence of an ideal speaker is related to a set of language invariant but parametricizable principles. This knowledge is represented in a multilevel theory, where the levels build a hierarchy in which the different stages are connected by wellformedness conditions on structural descriptions. A competence based theory in that sense requires a view that focuses on the abstract knowledge a speaker has about complete sentences. From the point of view of an on-line operating performance system knowledge of language has to be applied not only to complete sentences but far more often to parts of the phonetic/graphematic input in a timelike fashion. The assumption that language processing is on-line implies that the structural descriptions for a given sentence at any level have to be computed simultaneously. Under the view that syntactic structures are projected from the lexicon the structure of lexical items plays a main role in determining the resulting X-bar structures. The performance system therefore needs a specific procedure where syntactic structures are projected by these lexical items.

Especially in the case of head final constructions, the performance system needs some kind of memory in order to store constituents having no clearly defined status until the head (lexical or functional) of the projection they are linked to appears in the course of the time sequential signal. Furthermore, the performance system has to have a linking module which joins these structurally undetermined projections. In this paper I will show how a linking system based on government directions and saturation principles works in an LR(0) parser. This is a subtype of the class of deterministic push down automata which parse an input string from left to right without using any look-ahead ability. This means that the parser processes one word at a time without any need to see elements following in the input string.

0. Einleitung

Die grammatische Kompetenz von Sprechern wird in der GB-Theorie durch eine Menge von Prinzipiensystemen erfaßt, deren Anwendung auf verschiedenen syntaktischen Ebenen stattfindet. Das Standard-Grammatikmodell unterscheidet vier Ebenen: D-Struktur, S-Struktur, PF und LF. D-Struktur-Repräsentationen sind durch die Θ - und X-bar-Theorie und die kategorialen Eigenschaften der lexikalischen Einheiten festgelegt. Die Beziehung zwischen D-Struktur und S-Struktur ist durch eine allgemeine Bewegungsregel (move α) bestimmt, die ihrerseits aber durch unabhängige Teiltheorien (Barrieren- und Kasus-theorie) beschränkt ist. Eine S-Struktur repräsentiert dabei einerseits die in PF zu kodierende lineare Konstituentenabfolge und phonetisch nicht zu realisierende leere Kategorien. Andererseits wird aus einer S-Struktur durch

Quantorenanhebung und Rekonstruktionsprozesse die logische Form (LF) von Sätzen abgeleitet. Dabei sind LF-Repräsentationen von PF in dem Sinne unabhängig sind, als aufgrund von LF-Derivationen keine phonetischen Effekte entstehen, sondern nur Lesartvarianten. Die Ableitung einer wohlgeformten LF setzt also zum einen wohlgeformte D- und S-Strukturen voraus und erfordert zum anderen zulässige Operationen, die die strukturellen Konfigurationen der einzelnen Ebenen aufeinander beziehen. Dabei werden wesentliche Eigenschaften der S-Struktur (X-bar-Bedingungen und Θ -Rollen-Verteilung) bereits durch die D-Struktur festgelegt, und bestimmte Eigenschaften von LF sind bereits durch die S-Struktur determiniert, so z.B. die Ausgangsposition von Quantorenbewegung aber natürlich auch die phrasale Struktur und -via Projektionsprinzip- die thematischen Verhältnisse.

Nun ist aber der Begriff D-Struktur nur im Zusammenhang mit vollständigen Sätzen sinnvoll zu verwenden, d.h. zu einem Satzfragment wie etwa (1) läßt sich keine vernünftige D-Struktur angeben.

(1) Wen glaubt Peter, daß ▶

Beim Sprachverstehen tritt jedoch die Situation auf, daß das Performanzsystem an dem Punkt ▶ bestimmte Strukturkomplexe gebildet hat, auch wenn der Satz noch nicht vollendet ist. So ist etwa klar, daß das Fragepronomen wen in einer Position aufgetreten sein muß, die akkusativ-markiert ist. Die thematische Rolle des Pronomens ist aber völlig unbestimmt, da das Θ -Rollen zuweisende Prädikat noch nicht aufgetreten ist. Es ist aber völlig klar, daß das Performanzsystem auch dieser Teilkette (mindestens partiell) schon eine Struktur zugewiesen hat. Die Frage ist aber: Welche?

Da ein Hörer einerseits Strukturen abhängig von der Äußerung aufbaut, andererseits dazu jedoch invariante grammatische Kenntnis verwendet, stellt sich die Frage, auf welche Weise die grammatische Kenntnis verwendet wird, und inwieweit diese Verwendung durch die in der Äußerung enthaltenen Elemente gesteuert wird. Das Performanzsystem eines Hörers, also der Parser, ist ja in der Lage, einer phonetischen/graphematischen Kette eine Logische Form zuzuweisen, und es ist ebenso offensichtlich, daß eine solche Logische Form das abstrakte Resultat komplexer Berechnungsprozesse des Performanzsystems darstellt. Wie kann man sich nun den Übergang von PF nach LF vorstellen? Welche Rolle spielen die syntaktischen Ebenen jeweils, wenn dieser Übergang berechnet wird? Und wie kann der Aspekt, daß sprachliche Äußerungen von links nach rechts, d.h. zeitlinear berechnet werden, in diese Sichtweise integriert werden? Nun, der Parser muß zur Verarbeitung und Strukturierung einer zeitlinear organisierten und sukzessiv bekanntwerdenden phonetischen Kette u.a. Prozesse wie Phonemsegmentierung, Silbifizierung, Morphemerkennung, lexikalischen Zugriff, Applikation von Wortbildungsprinzipien, syntaktische Komplexbildung, semantische Struktur, konzeptuelle Interpretation usw. ausführen. Um eine syntaktische Struktur zu berechnen, wollen wir annehmen, daß der Parser auf der Grundlage voll flektierter Wörter operiert. Probleme der Worterkennung, der morphologischen Analyse usw. sollen dabei aus Einfachheitsgründen nicht mit einbezogen werden (vgl. dazu etwa Marslen Wilson/Tyler 1980 und Tyler/Frauenfelder 1987). Es läßt sich dann vorstellen, daß der Parser zu einem identifizierten phonologischen/graphematischen Wort dessen Lexikoneintrag kennt. Ein Lexikoneintrag

¹

In: *Zeitschrift für Sprachwissenschaft* 13.1, 1994, 56-109.

besteht aus den drei Komponenten $\langle P, MS, S \rangle$, wobei P für phonologische, MS für morphosyntaktische und S für semantische Information steht. Während P eine im wesentlichen lineare Struktur aufweist, ist die prädikatenlogische Darstellung, die S zugrundeliegt, hierarchisch. Da die Komponente MS beide Strukturaspekte besitzt, kann sie P auf S abbilden (vgl. Bierwisch 1983a). Wir können jetzt annehmen, daß der Parser zu jedem Wort dessen lexikalische Information verwendet, um die syntaktische Struktur zu berechnen. Dabei steht ihm zum einen die Argumentstruktur (bzw. das Θ -Raster) dieses Lexems zur Verfügung und zum anderen dessen syntaktische Kategorie. Wenn der Parser weiterhin die Prinzipien der X-bar-Theorie kennt, so kann er aus dieser Information die syntaktische Konfiguration berechnen, in dem das in Frage stehende Lexem auftritt. Dazu benötigt er als invariante grammatische Kenntnis die Prinzipien der X-bar-Theorie und hat als variable Information die Struktur von Lexikoneinträgen. Diese beiden Komponenten legen die D-Struktur fest. Was der Parser aber mit dem Lexikoneintrag etwa eines Verbs zugleich kennt, ist auch die Kasusrealisierung von dessen Argumenten. Kasusinformation ist aber erst auf S-Struktur relevant. Handelt es sich bei dem Lexem gar um einen Quantorenausdruck wie etwa jeder, so verfügt der Parser mit der Kenntnis dieses Lexems auch über relevante LF-Informationen. Insbesondere weiß er, daß der Skopus dieses Quantors der gesamte Satz ist und daß der Operorteil des Quantors an der linken Peripherie des Satzes auftreten muß. Mit einem Satzanfang wie (2) stehen dem Parser also Informationen zur Verfügung, deren Relevanz auf allen syntaktischen Ebenen des T-Modells wirksam wird.

(2) Liebt jeder ▶

Entscheidend ist nun, daß kein vollständiger Satz vorliegen muß, damit der Parser solche Informationen verarbeiten kann. Er muß vielmehr Prozesse ausführen können, damit schon diese lokal zur Verfügung stehenden Informationen richtig behandelt werden. Diese Situation skizziert, in welcher Weise D-Struktur-, S-Struktur- und LF-Eigenschaften gleichzeitig zu verarbeiten sind, ohne daß auch nur eine Struktur vollständig wäre.

Im folgenden interessieren mich die Mechanismen zur Bildung syntaktischer Strukturen, und ich werde das idealisierte Modell eines Parsers darlegen, welches es erlaubt, anhand weniger und einfacher Prinzipien aus der Information der in der Lautkette auftretenden Lexikoneinheiten eine syntaktische Struktur zu berechnen.

Dazu werden zunächst einige Bedingungen diskutiert, denen Sprachverstehensprozesse unterliegen. Dabei werde ich kurz darauf eingehen, daß die meisten der derzeit existierenden Modelle keine psycholinguistische Interpretation zulassen (Abschnitt 1). Sodann werde ich an einem Beispiel erörtern, wie ich mir die Berechnung einer sukzessiv bekanntwerdenden Lautkette vorstelle (Abschnitt 2), und daran anschließend werde ich zeigen, welche grammatischen Kenntnisse den Parser bestimmte Operationen ausführen lassen (Abschnitt 3).

In welcher Weise die syntaktischen Strukturen aus der Information über das Θ -Raster und aus der flexionsmorphologischen Markierung der Lexikoneinheiten projiziert werden können, wird sodann erörtert und grammatiktheoretisch eingebettet (Abschnitt 4).

Welche von diesen Projektionen als Spezifikator, Komplement oder Adjunkt zu behandeln ist, wird durch andere Berechnungen, sog. Linking-Prozesse, festgelegt, mit deren Hilfe sich

schließlich angeben läßt, wie die einzelnen Projektionen zu einer Gesamtstruktur zusammengesetzt werden (Abschnitt 5).

Dies geschieht durch bestimmte Verarbeitungsstrategien in einem sog. Kellerautomaten. Anschaulich gesprochen verfügt ein solcher Automat über einen Speicher, in dem Projektionen abgelegt werden können, deren struktureller Status zum Berechnungszeitpunkt nicht klar ist. Eine besondere Art der Speicherverwaltung macht es darüber hinaus möglich, das Auffinden von Komplementen, Spezifikatoren und Adjunkten bei kanonischer Realisierung als verhältnismäßig einfachen Prozeß zu verstehen (Abschnitt 6). Schließlich werden einige Parallelen zwischen den Berechnungsprozessen des Parsers und den leitenden Ideen des minimalistischen Programms (Chomsky 1992, 1994) erörtert (Abschnitt 7).

1. Grammatik und Parser

Für Sprachverstehensprozesse (SVP) will ich für das weitere annehmen, daß sie den folgenden Bedingungen in (3) unterliegen.

- (3) (i) SVP verlaufen von links nach rechts.
- (ii) SVP sind wesentlich bottom-up.
- (iii) SVP sind effizient.
- (iv) SVP sind so on-line wie möglich.

Davon ist (3)(i) die offensichtlichste Bedingung. Daß die Verarbeitung sprachlicher Ausdrücke von links nach rechts erfolgt, hat die Konsequenz, daß die Linearisierung innerhalb der Terminalkette bekannt ist, wenn man idealisierend davon ausgeht, daß Sprecher in der Regel grammatisch korrekte Ausdrücke formen. Damit könnte man annehmen, daß die Aufgabe des Parsers lediglich darin bestünde, nur die hierarchische Struktur dieser Kette zu berechnen. Diese Annahme ist jedoch falsch. Wäre nämlich nur die syntaktische Hierarchie zu rekonstruieren, so dürfte die Links-Rechts-Bedingung überhaupt keine Rolle spielen, da der Parser lediglich die Prädikat-Argument-Hierarchie zu berechnen hätte. Tatsächlich ist es aber ganz offensichtlich, daß Hörer auch fehlerhafte Linearisierungen feststellen, und das heißt, daß die Kenntnis über mögliche Linearisierungen einen festen Bestandteil der grammatischen Kompetenz auch des Hörers darstellt und daß das Performanzsystem diese Kenntnis verwendet. Es wird in einem der folgenden Abschnitte gezeigt, daß gerade die lineare Anordnung der Teilausdrücke eine notwendige Bedingung für die Berechenbarkeit syntaktischer Strukturen darstellt.

Die Bedingung (3)(ii) betrifft den Modus der Prozeßsteuerung und macht dabei die Einschränkung, daß wesentliche Teile der syntaktischen Struktur aus den Eigenschaften der terminalen Kette berechnet werden. Ein anderer Berechnungsmodus wäre etwa die top-down-Analyse. Dabei versucht der Algorithmus, zunächst Phrasensymbole (ausgehend vom Startsymbol S) zu expandieren und eine Ableitung zu finden, die der Terminalkette entspricht. Nun verlangen solche Algorithmen allerdings, daß eine Regelkomponente vorhanden ist, die Auskunft über mögliche Phrasenstrukturen liefert und daß unabhängig von den Elementen der Terminalkette Entscheidungen über mögliche Strukturen getroffen werden. Mit (3)(ii) sollen

derartige Berechnungsmodi aber gerade verhindert werden, da sie einer psycholinguistischen Rechtfertigung ebenso entbehren wie sie den grammatiktheoretischen Forschungsstand ignorieren. Vielmehr will ich hier den Standpunkt vertreten, daß sämtliche Konfigurationen der syntaktischen D-Struktur allein aus den Eigenschaften der Terminalkette (und den allgemeinen Bedingungen der X-bar-Theorie) abgeleitet werden können.

Nun ist spätestens seit Fodor (1983) klar, daß das Sprachverarbeitungssystem zur Klasse der Inputsysteme gehört, deren Berechnungen sich durch außerordentliche Schnelligkeit auszeichnen und die als Resultat einen relativ seichten Output liefern. Diese beiden Eigenschaften hängen eng mit der informationellen Abgeschlossenheit der Inputsysteme zusammen. Nur wenn wenig Information konsultiert wird, kann die Berechnung schnell sein, und nur dann ist die Komplexität der Berechnung gering. Um die Komplexität von Sprachverarbeitungsprozessen zu messen, bieten sich verschiedene, aus der Automatentheorie bekannte Metriken an. So ist es üblich, zwischen einem Komplexitätsmaß für die Speicherbelegung und einem Komplexitätsmaß für die Berechnungszeit zu unterscheiden. Die Metrik für die Berechnungszeit kann durch die Anzahl der Rechenschritte einer Turingmaschine (die wir in einem späteren Abschnitt genauer betrachten) angegeben werden. Die Metrik für den Speicherplatz mißt die Anzahl der Zellen, die während der Berechnung beschrieben werden. In der Regel hängt die Berechnungszeit vom Umfang der Grammatik $|G|$ (also der Anzahl der grammatischen Regeln) und der Länge n der eingegebenen Wortkette ab. So benötigt etwa der von Earley (1968) angegebene Algorithmus zur Erkennung kontextfreier Sprachen eine maximale Berechnungszeit, die proportional zu dem Polynom $|G|^2 \cdot n^3$ ist. Berechnungen heißen im komplexitätstheoretischen Sinne effizient, wenn sie durch einen Algorithmus ausgeführt werden können, für den sowohl die Berechnungszeit als auch der benötigte Speicherplatz durch ein Polynom von niedrigem Grad beschränkt ist. Das bedeutet vereinfacht, daß die Berechnungszeit nicht zu lang und der notwendige Speicherplatz nicht zu groß sein darf. Will man also ein Modell für das rezeptive Performanzsystem entwerfen, so kann es durchaus sinnvoll sein, Effizienzerwägungen mit einzubeziehen. Gleichwohl finden sich aber auch neben den bekannten 'garden-path'-Sätzen und tatsächlichen Ambiguitäten Konstruktionen, die dem menschlichen Parser Schwierigkeiten bereiten, wie etwa in (4).

- (4) (i) Steuern Steuern Steuern?
(ii) wenn Lügner Lügner Lügner nennen.

Aber es scheint, daß derartige Fälle eher die Ausnahme darstellen.

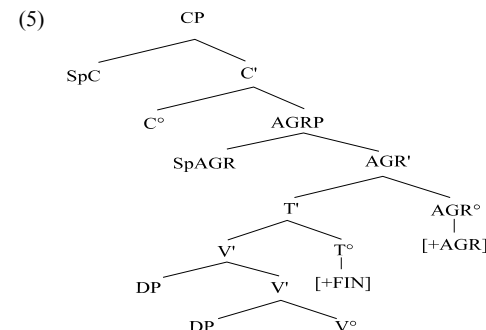
Speziell mit der Speicherplatz-Komplexität hängt die Bedingung (3)(iv) zusammen. Eine Sequenz von Berechnungen ist ein strikter on-line Prozeß, wenn keine Berechnung auf das Resultat einer früheren Berechnung zurückgreift, die durch andere, unabhängige Berechnungen getrennt ist. Dies besagt im wesentlichen, daß bei strikten on-line Prozessen keine Berechnungsergebnisse gespeichert werden müssen, sondern daß diese direkt in den Folgeberechnungen verwendet werden. Wie sich leicht zeigen läßt, sind strikte on-line Berechnungen in der Verarbeitung natürlicher Sprache nur dann möglich, wenn diese Sprachen durchgängig kopf-initial sind und darüber hinaus keine Bewegung nach links erlauben. Soweit sich beim gegenwärtigen Forschungsstand sehen läßt, existieren derartige natürliche Sprachen

nicht. Gleichwohl formuliert die Bedingung so on-line wie möglich eine allgemeine Restriktion darüber, wie viele Berechnungsergebnisse wie lange gespeichert werden, nämlich: so wenige wie möglich und diese nur so lange wie nötig. Eine Präzisierung dieser Eigenschaft erfolgt im weiteren Verlauf (vgl. auch Bierwisch (1983), der den Zusammenhang zwischen grammatisch determinierten Abhängigkeiten und den Operationen, die zu ihrer Berechnung führen, formal und sehr differenziert erörtert).

Unter den Bedingungen in (3) operiert das Performanzsystem also bereits auf Teilen der Lautkette, ohne daß diese vollständig bekannt ist, d.h. die Struktur eines komplexen Ausdrucks muß aus der Kenntnis der einzelnen Teile unter den in (3) genannten Bedingungen berechnet werden.

Ein Modell mit diesen Eigenschaften zu konstruieren, ist der Versuch, einen psycholinguistisch motivierten Parser zu entwickeln. Die Frage ist nun, wie ein solches Modell organisiert sein muß, welches einerseits die Repräsentationen einer GB-Grammatik berechnen kann, andererseits aber auch den genannten Verarbeitungsprinzipien unterliegt. Dazu ist zunächst der Linearisierungsaspekt des sukzessiv bekanntwerdenden Sprachsignals zu betrachten. Die Situationen, in denen sich der Parser während des Verstehensprozesses befindet, lassen sich durch die folgende abstrakte Betrachtung charakterisieren.

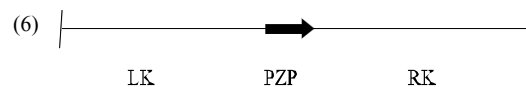
Der Parser startet in einem Zustand S_0 , in dem noch keine Information über die zu strukturierende Kette vorhanden ist. Dies wäre etwa die Situation, in der sich ein Hörer befindet, wenn nichts gesagt wird. Sobald jedoch ein Sprachsignal geäußert wird, beginnt der Parser, dieser Äußerung eine Struktur zuzuweisen. Dies geschieht über eine Sequenz von Zwischenzuständen S_n , in denen die Fortführung der Äußerung jeweils stückweise bekannt wird. In irgendeinem Zustand S_n hat der Parser aufgrund seiner grammatischen Kenntnis eine Repräsentation aufgebaut, die einer vollständigen Satzstruktur entspricht. Legt man zu Modellierungszwecken die in einer Variante der Rektions- und Bindungstheorie angenommene Satzstruktur in (5) zugrunde, so heißt dies, daß der Parser im Zustand S_n aufgrund seiner internen Konfiguration und der Lautkette rekonstruiert hat, daß eine Satzkonstituente vollständig ist.



Es ist nun so, daß der Parser dazu einerseits das Sprachsignal selbst kennen muß, daß er andererseits aber mehr Informationen benötigt, um die Struktur in (5) zu bilden. Bemerkenswert daran ist, daß die grammatische Kenntnis, die zur Durchführung solcher Berechnungen

notwendig ist, von der grammatiktheoretisch repräsentierten Kenntnis in nicht unerheblichem Maß abweicht, da der Parser nur im Zustand S_n die vollständige Kette kennt, in allen anderen Zuständen S_i aber nur Teile davon. Ein an der Rektions- und Bindungstheorie orientierter Parser benötigt zur Berechnung einer S-Struktur aber eine vollständige D-Struktur und zur Berechnung von LF eine vollständige S-Struktur. Die im T-Modell ausgedrückte Hierarchie der einzelnen Ebenen legt also die Serialisierung der Berechnungen in Abhängigkeit von der Hierarchie der syntaktischen Ebenen derart fest, daß auf jeder Ebene vollständige Repräsentationen vorliegen müssen. Ein von links nach rechts arbeitender Parser hingegen muß auf Teilen des Signals operieren und zu diesen die Strukturen aller Ebenen gewissermaßen simultan berechnen.

Um die Vorstellung dieser zeitlinearen Verarbeitungsweise etwas zu präzisieren, will ich das Signal in die drei Partitionen linker Kontext (LK), Prozeßzeitpunkt (PZP) und rechter Kontext (RK) zerlegen, wie im Diagramm (6) dargestellt.



Der Parser muß daher stets in dem durch PZP gekennzeichneten Zustand Entscheidungen darüber treffen, welche Struktur er aufbauen soll bzw. welcher Prozeß stattfinden muß. Dazu kann er auf die Kenntnis des linken Kontextes zurückgreifen, nicht aber auf das lexikalische Material, welches im rechten Kontext auftreten wird, da dieses zum Prozeßzeitpunkt unbekannt ist.

Rückt man diejenigen S_i , die durch Wortgrenzen festgelegt werden, in den Aufmerksamkeitsbereich, so werden die Informationen relevant, die mit den lexikalischen Einheiten verbunden sind. Zu fragen ist also, welche Teilstrukturen der Parser in Kenntnis eines Lexems in S_i aufbauen kann. Da ein prinzipienbasierter Parser ohne Phrasenstrukturregeln arbeitet, können diese auch nicht zur Strukturbildung herangezogen werden. Verzichtet man aber auf PS-Regeln, so sind zwei Berechnungsalternativen denkbar, um eine vollständige Satzstruktur abzuleiten:

- Entweder man gibt eine Struktur vor (etwa eine vollständige CP-Struktur wie in (5)) und prüft anhand der Signalkette, ob die auftretenden Phrasen und Lexeme in diese Struktur integriert werden können, oder
- die gesamte Struktur muß aus dem lexikalischen Material kompositionell berechnet werden.

Wenngleich des öfteren bzgl. der funktionalen Kategorien von einem 'universellen Phrasenskelett' gesprochen wird, ist die erste Alternative nicht zu realisieren, denn die Struktur von Phrasen mit lexikalischem Kopf variiert mit dessen lexikalischen Eigenschaften, so daß die Idee, Phrasengerüste via lexikalischer Einsetzung mit Terminalsymbolen zu füllen, stets voraussetzt, daß die passenden Strukturen bereits abgeleitet sind oder an die Subkategorisierungsinformation der in einem Satz auftretenden Lexeme angepaßt werden. Nun

können aber erstens passende Strukturen nicht unabhängig von den Lexemen durch PS-Regeln abgeleitet werden, und daher kann zweitens auch nicht von einem universellen Phrasenskelett ausgegangen werden, da die Kenntnis eines Lexems notwendig ist, um die zugehörige Struktur aufzubauen. Zu klären ist also, aufgrund welcher Kenntnis die syntaktische Struktur eines Satzes aufgebaut wird, und welche Rolle die lexikalischen Einheiten dieses Satzes dabei spielen. Dazu haben Haider und Bierwisch (1989) im Anschluß an Higginbotham (1985) eine Konzeption vorgelegt, die die Entfaltung syntaktischer Konfigurationen wesentlich aus den Eigenschaften der beteiligten Lexikoneinheiten ableitet. Die in Form des Θ -Rasters kodierte Information enthält bereits alle für die syntaktische Konfiguration des Argumentbereichs notwendige Information. Syntaktische Strukturen sind demnach Abbildungen des im Lexikon enthaltenen Θ -Rasters, so daß keine PS-Regeln notwendig sind, um diese Strukturen aufzubauen. Einem Parser genügt unter dieser Sichtweise also offensichtlich die Information, die mit einem lexikalischen oder funktionalen Kopf verbunden ist, um die zugehörige syntaktische Projektion festzulegen.

Die meisten der derzeit existierenden Parser sind nun aber gerade durch PS-Regeln gesteuert, die in den jeweiligen Zuständen S_i spezifizieren, welche Strukturen aufzubauen sind (vgl. Berwick/Abney/Tenny 1991). Da in einer prinzipienorientierten Grammatiktheorie der Begriff der konstruktionsspezifischen PS-Regel aber vollständig obsolet ist, kann ein derart gesteuerter Parser eine solche Grammatiktheorie auch nicht implementieren. Um die verschiedenen Konstruktionstypen einer Sprache zu strukturieren, müßte der Parser über mehrere tausend solcher Regeln verfügen, was unter Lernbarkeitsbedingungen und grammatiktheoretischen Erwägungen höchst unplausibel ist.

Eine bestimmte andere Klasse von Parsern basiert auf den Prinzipien von Theorembeweisern, so etwa Fong (1991) oder Kolb (1993). Die Struktur und Prozeßweise dieser Parser läßt jedoch keine psycholinguistische Interpretation zu. So geschieht etwa die Berechnung der syntaktischen Struktur im Sinne der X-bar-Theorie in dem von Kolb (1993) dargestellten Parser in der Weise, daß ein Strukturzuweisungsmodul zunächst alle x-bar-theoretisch möglichen Phrasengerüste aufbaut, die anschließend durch die Prinzipiensysteme der jeweiligen Ebenen wieder verworfen werden. Dies führt schon bei relativ einfachen Sätzen zu einer Anzahl von Strukturalternativen, die enorm hoch ist, woraus sich direkt das Problem ergibt, diese strukturellen Varianten auf die Menge der adäquaten Repräsentationen zusammenzukürzen. Dies ist de facto ein Berechnungsaufwand, der unter Verwendung lexikalischer Information vermieden werden kann, womit das Problem der Reduktion von unnötigerweise aufgebauten Strukturen ebenfalls entfällt.

In dem folgenden Abschnitt will ich die Vorstellung von einem Parser, der allein aus der Kenntnis der Lexikoneinheiten die Struktur eines Satzes festlegt, an einem Beispiel illustrieren. Bevor wir uns dieser Beispielanalyse zuwenden, sollte jedoch geklärt werden, über welche Kenntnisse der Parser verfügt, wenn er sich im Zustand S_0 befindet. Diese Kenntnis stellt gewissermaßen die Kompetenz Parsers dar. In Abschnitt 4 wird sie explizit formuliert und theoretisch eingebettet. Da der Parser eine Satzstruktur wortweise von links nach rechts berechnet und zu jedem auftretenden Lexem eine syntaktische Projektion aufbaut, muß er zunächst über die Informationen in (7) verfügen.

(7) Grammatische Kenntnis:

- (i) lexikalische und funktionale Kategorien
- (ii) Rektionsrichtung dieser Kategorien

Mit (7)(i) weiß er also, daß die Menge {N, A, V, P} die lexikalischen Kategorien darstellt und die Menge {D, C, T, Agr} die funktionalen Kategorien. Dies ist in der folgenden Weise repräsentiert: {(N . L), (A . L), (V . L), (P . L), (D . F), (C . F), (T . F), (Agr . F)}, wobei L und F jeweils lexikalisch und funktional abkürzen. Mit (7)(ii) besitzt er erstens die Information, daß die jeweiligen Kategorien überhaupt für eine Rektionsrichtung ausgezeichnet sind und zweitens, daß diese Eigenschaft parametrisierbar ist. Die möglichen Parameterwerte liegen in der Menge: {links, rechts}. Soll der Parser also deutsche Sätze strukturieren, so werden die Parameter wie folgt gesetzt: {(N . R), (A . L), (V . L), (P . R), (D . R), (C . R), (T . L), (Agr . L)}, wobei L und R in diesem Fall jeweils links und rechts abkürzen. Hat V etwa den Wert L, so werden die mit einem speziellen Verb verbundenen Argumentstellen zu einer kopffinalen VP-Struktur projiziert. Setzt man aber den Wert für V auf R, so projiziert der Parser eine kopfinitiale VP-Struktur. Wenn der Parser im Lexikon ein Wort gefunden hat, so ist ihm dessen syntaktische Kategorie bekannt, so daß er weiß, welche Orientierung die zugehörige Projektion hat. Zugleich ist ihm aber auch die Argumentstruktur bzw. das Θ -Raster dieses Lexems bekannt. Das Θ -Raster ist eine spezifische Eigenschaft der lexikalischen Einheiten. Im Standardfall sind die Elemente des Θ -Rasters merkmalslos, d.h. etwa für ein bitransitives Verb wie geben, daß die Θ -Rollen im Lexikon keine Merkmale aufweisen. Werden diese Rollen jedoch zu einer syntaktischen Struktur projiziert, so werden sie mit verschiedenen Merkmalen versehen. In diesem Fall werden den Θ -Rollen ihre 'default'-Kasus zugewiesen, d.h. Akkusativ für das erste interne Argument, Dativ für das zweite interne Argument und Nominativ für das externe Argument. Für das Verb helfen ist das interne Argument bereits im Lexikon für Dativ markiert und erhält daher keine weitere Markierung. Die Kennzeichnung einer Θ -Rolle als intern oder extern bzw. als referentiell oder nicht-referentiell wird über die Position im Θ -Raster berechnet. Weiterhin weiß der Parser, daß interne Argumente innerhalb der Projektion eines Kopfes realisiert werden müssen und externe Argumente außerhalb, d.h. er kennt die X-bar-Bedingungen für Komplementation. Der Parser verfügt zur Berechnung dieser Merkmale, erstens über die Informationen in (8)(i), die im Lexikon enthalten sind und zweitens über die Informationen in (8)(ii), die zur grammatischen Kenntnis gehören.

- (8) (i) Lexikon:
 - a. syntaktische Kategorie
 - b. Θ -Raster (gegebenenfalls mit Markierungen, wie lexikalische Kasus, Status usw.)
- (ii) grammatische Kenntnis:
 - a. Bedingungen für Θ -Rollen (referentiell, intern)
 - b. X-bar-Bedingung für Komplementation
 - c. Kasusbedingungen

Mit diesen Informationen ist der Parser nun in der Lage, die Komplementbereiche der

lexikalischen Kategorien vollständig zu berechnen.

Für die funktionalen Kategorien sind weitere Informationen notwendig. So muß der Parser wissen, von welchen morphosyntaktischen Merkmalen die funktionalen Kategorien inhaltlich gefüllt werden. Die Kategorie D beinhaltet sowohl Genus- und Numerus-Merkmale als auch Kasusmerkmale, die Kategorie Agr Person- und Numerus-Merkmale, die Kategorie T Tempusmerkmale und die Kategorie C Modusmerkmale. Diese Informationen sind kategorienspezifisch und werden in der folgenden Weise repräsentiert: {(D . [G . ()], N . () K . ())}, (Agr . [P . (), N . ()]), (T . [T . ()]), (C . [M . ()])}, wobei G, N, P, T, M, K jeweils Genus, Numerus, Person, Tempus, Modus und Kasus abkürzen und die leeren, runden Klammern jeweils die Merkmalslosigkeit darstellen. Wenn der Parser aber ein finites Verb der 1. Person, Singular, Präsens, Indikativ wie etwa lache berechnet, so werden diese morphosyntaktischen Merkmale verwendet, um die Eigenschaften der funktionalen Kategorien zu spezifizieren. Nach der Projektion des Komplementbereichs rekonstruiert der Parser aus diesen Merkmalen die Köpfe der funktionalen Kategorien. Diese Köpfe werden nun ihrerseits unter Bezug auf ihre jeweilige Rektionsrichtung projiziert. Die Einbindung der f-selektierten 'Komplemente' fällt dabei unter die X-bar-Bedingung für Komplementation, so daß die grammatische Kenntnis noch um die Kopf-Spezifikator-Beziehung erweitert werden muß. Da der Parser nun weiß, durch welche morphosyntaktischen Merkmale die funktionalen Kategorien festgelegt werden, und da nur funktionale Kategorien Spezifikatoren aufweisen, genügt es, die Spezifikator-Merkmale dieser Kategorien anzugeben. Da sowohl für D als auch für T nicht ohne weiteres klar ist, von welcher Art die Kopf-Spezifikator-Relation ist, beschränke ich mich auf C und Agr. Für diese beiden Kategorien muß eine Übereinstimmung von Kopf und Spezifikator jeweils in den Merkmalen [$\pm$ wh] und [Person, Numerus] vorliegen. Der Parser muß also weiterhin über die Kenntnisse in (9) verfügen.

- (9) (i) Lexikon:
 - spezifizierte morphosyntaktische Merkmale
- (ii) grammatische Kenntnis:
 - a. unspezifizierte morphosyntaktische Merkmale, die funktionale Kategorien etablieren
 - b. X-bar-Bedingung für Spezifikation
 - c. Merkmale für Kopf-Spezifikator-Kongruenz

Mit dieser Information ausgestattet, kann er nun auf jedem Element der Terminalkette Berechnungen anstellen, die als Resultat die zugehörigen syntaktischen Projektionen liefern. Weitere Prozesse der strukturellen Integration müssen diese Projektionen aber noch zu einer zusammenhängenden Struktur verbinden. Der Parser muß daher für bestimmte Merkmale sensitiv sein, die festlegen, welche fertige Projektion an welche Position in einer anderen Projektion gebunden wird. Dazu dienen Linking-Merkmale. Wir betrachten hier nur die verbalen Linking-Merkmale {Kasus, Status, wh}. Die Repräsentation der jeweiligen Kenntnisse läßt es zu, diese Menge einfach zu erweitern, indem weitere Linking-Merkmale hinzugefügt werden.

Da der Parser von links nach rechts Projektionen aufbaut und nur die Informationen

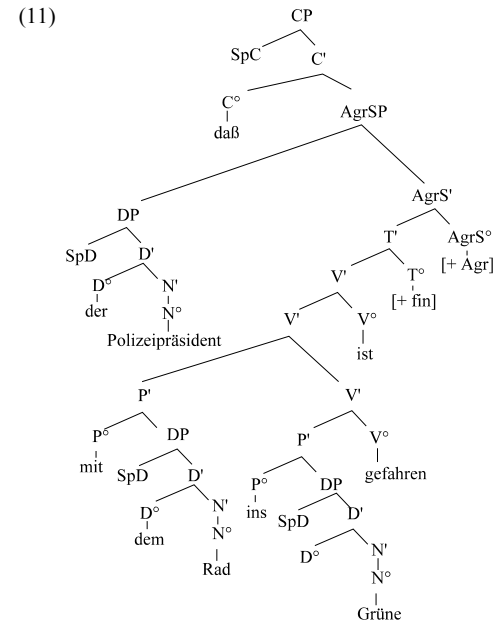
verwendet, für die Evidenz in der Signalkette auftritt, kann es vorkommen, daß die Lizenzierung von Teilstrukturen zu bestimmten Prozeßzeitpunkten noch nicht klar ist. Dies ist z.B. dann der Fall, wenn ein lizenzierender Kopf im rechten Kontext liegt. Diese Teilstrukturen müssen daher zwischengespeichert und verwaltet werden. Die Art dieser Verwaltung geschieht in dem hier vorgestellten Parser auf spezielle Weise, die sicherlich bei der Betrachtung des folgenden Beispiels deutlich wird. Um die Vorstellung von dieser Parserkonzeption daher weiter zu präzisieren, wollen wir im nächsten Abschnitt ein Beispiel analysieren und uns die Arbeitsweise genauer ansehen.

2. Eine Beispielanalyse

Da der Parser ohne PS-Regeln arbeitet, benötigt er Kriterien, nach denen er entscheidet, wann und wie Teil-Projektionen zusammengefügt werden müssen. So besteht in dem folgenden Satz (10) bzgl. der linearen Abfolge zunächst kein Bezug zwischen Subjekts- und Finitumsposition.

(10) ..., daß der Polizeipräsident mit dem Rad ins Grüne gefahren ist.

Erst mit dem Auftreten der am Finitum realisierten Agr-Merkmale und dem Aufbau von deren Sp-Position wird dieser Bezug durch Sp-Kopf-Kongruenz hergestellt. Da zwischen Subjekt und phrasenfinalem Agr-Kopf aber beliebig viele lexikalische Einheiten auftreten können, muß der Parser zwischenzeitlich auf anderen Projektionen operieren. Wenn er aber auf das Finitum und die dort realisierten funktionalen Kategorien trifft, muß sichergestellt werden, daß er das Subjekt identifizieren kann.

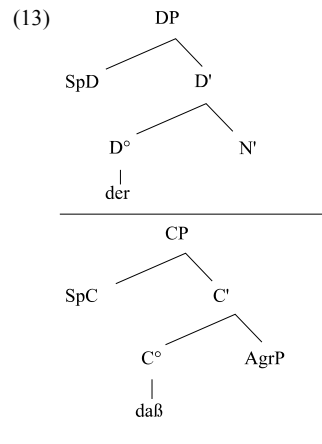


Da keine PS-Regeln der Form $\text{AgrSP} \rightarrow \text{SpAgrS AgrS'}$ und $\text{AgrS'} \rightarrow \text{T' AgrS}^0$ zur Verfügung stehen, muß in anderer Weise sichergestellt werden, daß Subjekt und AgrS^0 an irgendeiner Stelle des Prozesses adjazent auftreten und unter den gemeinsamen Mutterknoten AgrSP integriert werden können. Dazu ist erforderlich, daß sämtliche Strukturen von der Terminalkette bis zum obersten Knoten so aufgebaut werden, daß SpAgrS und AgrS^0 wieder zusammengeführt werden. Der Kopf der AgrSP befindet sich am Ende der Terminalkette, und der Parser hat, während er von links nach rechts Projektionen aufbaut, keine Kontrolle darüber, welches Wortmaterial im weiteren Verlauf des Satzes auftritt. Die zwischenzeitlich zu prozessierenden Projektionen können links- oder rechtsorientiert sein. Diese Orientierung erlaubt dem Parser, Erwartungen hinsichtlich der Position von Argumenten im linken oder rechten Kontext aufzubauen.

Lassen wir den Prozeßzeitpunkt PZP nun von links nach rechts wortweise die Terminalkette passieren, so muß der Parser in den verschiedenen Zuständen, die mit S_i ($0 \leq i \leq 10$) gekennzeichnet sind, Entscheidungen treffen.

(12) S_0 daß S_1 der S_2 Polizeipräsident S_3 mit S_4 dem S_5 Rad S_6 ins S_7 Grüne S_8 gefahren S_9 ist S_{10}

Wie werden diese Entscheidungen getroffen? In S_0 ist der linke Kontext leer, und der Parser befindet sich im Ausgangszustand. Beim Übergang nach S_1 prozessiert er das C^0 -Element daß und baut eine CP-Struktur auf, die eine Agr-Projektion selektiert. Deren Merkmale sind aber noch unbestimmt, da sich sowohl das Subjekt als auch das Finitum in RK befindet und somit beide Träger der Kongruenzmerkmale zu diesem Zeitpunkt unbekannt sind. Die C-Projektion bildet die maximale Projektion, die in Kenntnis des Lexems daß projiziert und spezifiziert werden kann. Da ein C^0 -Element eine AGR-Phrase selektiert und diese nach rechts regiert, muß die C-Projektion auf weiteres lexikalisches Material in RK warten und wird daher zunächst gespeichert. Der Parser schreitet nun weiter zum Zustand S_2 . Dort findet er ein definites D^0 -Element. Damit hat er zunächst die Operationen auf der CP beendet und befindet sich nun in einer D-Projektion, die eine N-Projektion selektiert und nach rechts regiert. Auch in diesem Zustand muß der Parser auf Elemente in RK warten, so daß auch die D-Projektion gespeichert werden muß. Es befinden sich nun zwei unvollständige Projektionen im Speicher, von denen die zuletzt berechnete die oberste Zelle füllt, vgl (13).



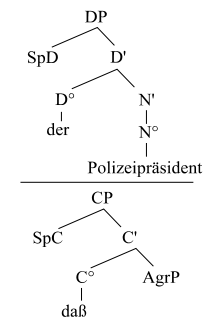
Ich folge bei der Festlegung der obersten Projektionsstufe für lexikalische Kategorien einer Überlegung von Fukui/Speas (1986), derzufolge lexikalische Kategorien die X' -Ebene iterieren und gewissermaßen nicht abgeschlossen werden. Funktionale Kategorien werden hingegen durch einen Spezifikator abgeschlossen und projizieren daher zur XP-Ebene. Die Idee dieses Ansatzes besteht darin, daß nur funktionale Kategorien Spezifikatorpositionen lizenzieren während lexikalische Kategorien diese Eigenschaft nicht aufweisen sondern maximal sind, wenn sie nicht weiter projizieren können; vgl. zu dieser Sichtweise auch Chomsky (1994:10).

In S_3 findet der Parser das zugehörige Nomen, baut die N-Projektion auf und kann diese Projektion nun direkt zu einer komplexen Struktur in die oberste Zelle des Speichers LGL integrieren, d.h. daß für die N-Projektion keine neue Zelle geschaffen werden muß, sondern daß diese vielmehr lokal in die zuletzt berechnete Struktur eingebunden werden kann. Der Terminus

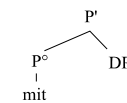
lokal meint, daß nur die oberste Zelle von LGL und die aktuell berechnete Projektion in PZP betrachtet werden muß, nicht aber untere Zellen in LGL. Dazu greift der Parser von der aktuell berechneten Position aus auf die oberste Zelle in LGL zu und berechnet Linking-Relationen. Zu der Entscheidung, daß die D- und die N-Projektion zu einem Komplex zusammengebunden werden müssen, dienen die funktionale Selektion sowie die Kongruenz- und Kasusmerkmale an den beiden N' -Positionen. Daraus entsteht eine neue Konfiguration durch strukturelle Integration eines Teilkomplexes in eine bereits vorhandene Struktur und nicht durch Speichern einer Teilstruktur.

Der Parser schreitet weiter und findet in S_4 nun eine Präposition, zu der die zugehörige PP aufgebaut wird, vgl. (14). Aufgrund der Rechtsorientierung präpositionaler Köpfe wird die zugehörige Dativ-DP im rechten Kontext als nächste Projektion erwartet.

(14) LGL:

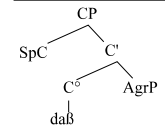
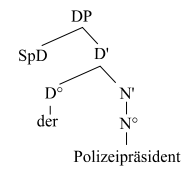
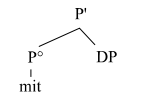
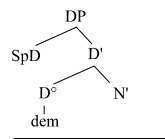


PZP:



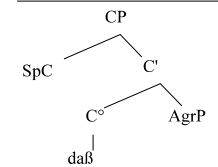
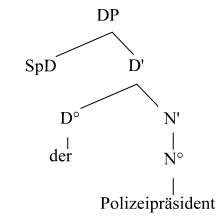
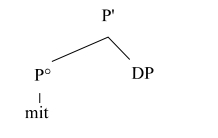
Die PP wird also in eine neu geschaffene oberste Kellerzelle gesetzt, und der Parser geht nach S_5 . Er findet dort das definite D^0 -Element dem, baut zu diesem eine DP-Struktur, die nach rechts eine N-Projektion erwartet und legt die Struktur in eine neu geschaffene oberste Zelle. Obwohl an dieser Stelle auch die Integration der DP in die PP aufgrund der Kasusmarkierung möglich wäre, wird sie nicht ausgeführt, da die DP noch unvollständig ist. Würde der Parser bereits hier die unvollständige DP integrieren, so müßte er im nächsten Schritt die N-Projektion in die PP integrieren, so daß ein Bezug zwischen P- und N-Projektion, nicht aber zwischen D- und N-Projektion hergestellt werden müßte. Ein fundamentales Linking-Prinzip besteht also darin, daß nur vollständige Projektionen gelinkt werden können. Dies führt zu der Konfiguration in (15), in der der Parser bereits nach S_6 weitergeschritten ist und das zugehörige Nomen gefunden hat.

(15) LGL: PZP:

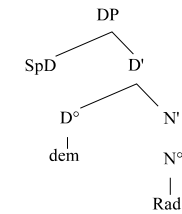


Zu diesem baut er eine N'-Projektion auf und integriert sie in die oberste Zelle. Damit liegt dort eine vollständige DP. Diese wird nun nach PZP versetzt, um als vollständige Phrase zum Linking in die PP zur Verfügung zu stehen, vgl. (16).

(16) LGL:



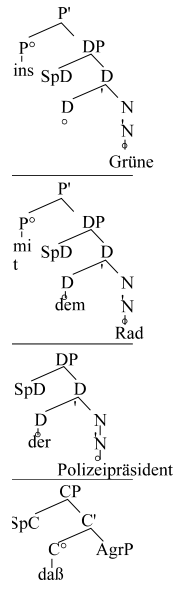
PZP:



Die aktuell projizierte DP in PZP wird also in die P-Projektion integriert, so daß sich in der obersten Zelle eine vollständige PP befindet.

Der Parser schreitet weiter nach S_7 , und es findet ein analoger Prozeß bei der Berechnung der zweiten PP statt. Es befinden sich nun die beiden PPn in den zwei oberen Zellen, wobei die zuletzt berechnete PP die oberste Zelle belegt. Diese PP ist nun aber diejenige, die als nächste auch wieder in die aufzubauende V-Projektion integriert werden muß. Indem der Parser zum nächsten Terminalelement weiterschreitet, erreicht er den Zustand S_9 , in dem er die Infinitform im dritten Status findet, vgl. (17).

(17) LGL:
:

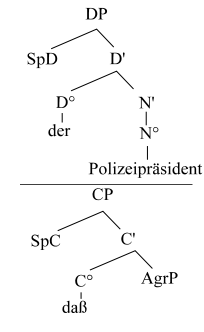


PZP:
gefahren

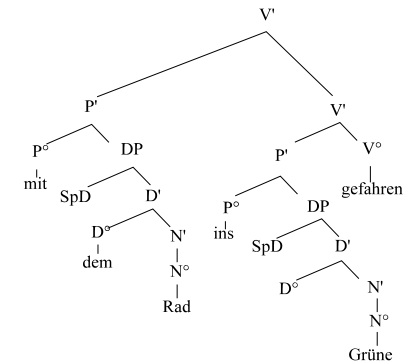
Der Parser beginnt nun mit dem Aufbau der V-Projektion. Mit einem Blick in die oberste Zelle identifiziert er das präpositionale Objekt und bindet es in die V-Projektion ein, so daß nun das Adjunkt oberstes Element ist.

In einem zweiten Schritt wird das Adjunkt in die V-Projektion integriert, so daß sich nun in PEL eine komplexe infinite Verbalprojektion und in der obersten Zelle von LGL das Subjekt befindet. Da das infinite Verbum keine Kongruenzmerkmale aufweist, kann die aufgrund der AgrS-Merkmale lizenzierte Subjekt-DP nicht in die verbale Projektion integriert werden, vgl. (18).

(18) LGL:

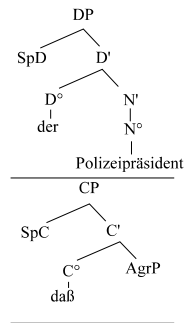


PZP:

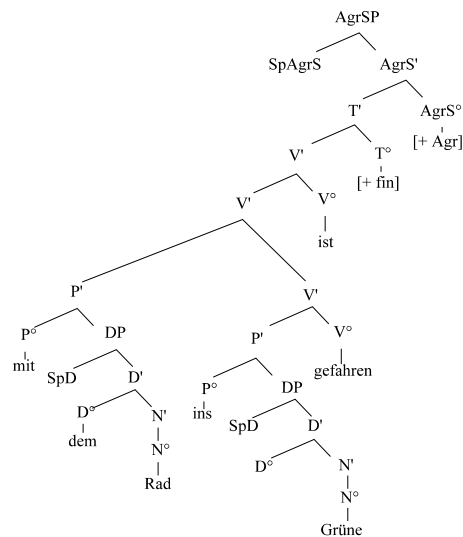


Der Parser versetzt daher die V-Projektion in die oberste Zelle und schreitet weiter in den Zustand S₁₀. Dort trifft er auf das Finitum, welches eine Verbalprojektion im dritten Status regiert. Diese befindet sich aber gerade in der obersten Zelle, vgl. , so daß sie zur Bildung des Verbalkomplexes direkt zur Verfügung steht. Der Parser integriert daher den statusregierten V-Komplex in die vom Finitum projizierte Struktur. Damit wird erneut die Subjekt-DP zur obersten Zelle. Die am Finitum realisierten flexionsmorphologischen Merkmale { Tempus, Agr, Modus } bilden nun die Köpfe der funktionalen Projektionen T, AgrS und C deren Projektionen relativ zu den jeweiligen f-selektionalen Beziehungen aufgebaut werden. Da AgrS eine T-Phrase selektiert, wird zunächst die T-Projektion aufgebaut und im Anschluß daran die AgrS-Projektion. Als deren letzte Position wird die SpAgrS-Position aufgebaut. Während dieses Prozesses befindet sich die Subjekt-DP nach wie vor in der obersten Zelle, so daß nun beide Positionen adjazent auftreten und direkt aufeinander bezogen werden können. Strukturelle Integration der Subjekt-DP in die SpAgrS-Position kann also stattfinden, vgl. (19).

(19) LGL:

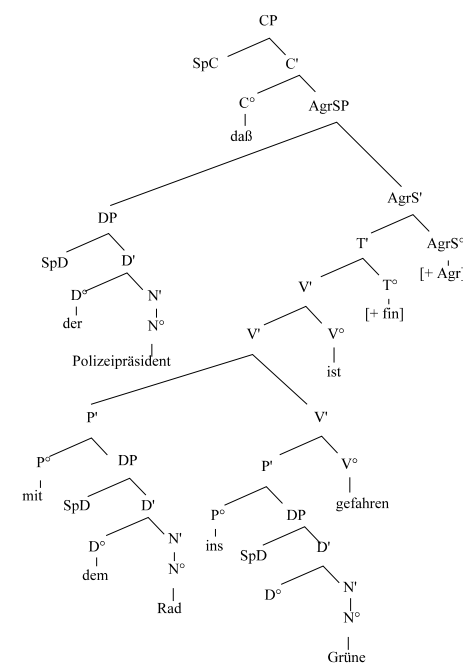


PZP:



Dies hat zur Konsequenz, daß CP oberste Zelle wird, während sich zugleich eine vollständige AgrP in PEL befindet. Der Parser prüft nun wiederum die Eigenschaften des obersten Knotens in PZP und dem fehlenden Argument der CP-Struktur in der obersten Zelle. Da dieses Argument vom Typ AgrP ist, wird die projizierte AgrP aus PEL in die CP-Struktur integriert. Nach diesem Schritt liegt eine vollständige Satzstruktur vor wie in (20).

(20) LGL:



PZP:

Unter dieser Konzeption kann die für die syntaktische Komplexbildung relevante Information lexemweise behandelt werden. In den zu betrachtenden Zuständen S_i führen Projektionsprozesse zu Teilstrukturen, die zunächst strukturell unverbunden sind und in der zeitlichen Abfolge ihres Auftretens nebeneinander stehen. Zur Komplexbildung benötigt das Performanzsystem Linking-Prozeduren, die diese partiellen Projektionen zu einer Struktur zusammenbinden. Linking-Prozesse beziehen eine Projektion mittels eines Linking-Merkmals auf die Position in einer Struktur. Damit teilen sie wesentliche Eigenschaften mit generalisierten Transformationen, woraus sich interessante Parallelen zum minimalistischen Programm (vgl. Chomsky 1992) ergeben. Linking-Prozesse finden in den durch das Θ -Raster und die Flexionsmorphologie determinierten Domänen statt. Innerhalb dieser Domänen werden Merkmale geprüft. Wenn keine Merkmale an bestimmten Positionen geprüft werden können, so bricht die Berechnung zusammen, wiewohl aufgebaute Teilstrukturen bestehen bleiben. Dies hat den Effekt, daß grammatisch wohlgeformte Teile eines Satzes erkannt und interpretiert werden können, daß aber an Stellen, die den grammatischen Bedingungen nicht genügen, keine Strukturen rekonstruiert werden. Betrachtet man etwa den Satz (12) im Zustand S_9 und ersetzt man das Auxiliar ist durch hat, so wird dieser Satz ungrammatisch.

(21) *daß der Polizeipräsident mit dem Rad ins Grüne gefahren hat

Tatsächlich konnten aber bis zum Zustand S_9 weite Teile dieses Satzes richtig strukturiert werden. Da die syntaktischen Kongruenzbeziehungen ebenfalls richtig ausgedrückt sind, liegt also nur ein Verstoß gegen das Θ -Kriterium vor, da in der Konstruktion mit hat ein weiteres Argument erforderlich ist. Fehlt dieses Argument, so ist eine wesentliche Schnittstellenbedingung auf LF verletzt und die Berechnung scheitert partiell, wiewohl Teilkomplexe strukturiert werden konnten. Wenn dieses Argument jedoch in der Terminalkette auftritt, so berechnet der Parser auch die richtige Struktur.

(22) daß der Polizeipräsident seinen Hund mit dem Rad ins Grüne gefahren hat

Wenn er also in dem Beispiel (22) im Zustand S_9 auf das Auxiliar hat trifft, so befindet sich zugleich in der obersten Zelle die in (22) unterstrichene Akkusativ-NP. Er kann dann, gesteuert von der Θ -Struktur des Auxiliars hat, die richtigen Integrationsprozesse berechnen. Dies geschieht nun aber völlig unabhängig von Regeln in der Kontrollstruktur des Parsers sondern wird vollständig durch die Argumentstruktur des Auxiliars bestimmt.

Da lexikalischer Zugriff voll flektierte Lexikoneinheiten liefert, werden gewisse Bewegungen (Infl-Bewegung) obsolet. Stattdessen werden Prüfbedingungen notwendig, die die Merkmalspezifikationen in gewissen Domänen testen. So muß in (22) geprüft werden, ob der Kasus sowohl an der Projektionsposition, die von dem verbalen Komplex aufgebaut wird, als auch am obersten Element von LGL, der DP seinen Hund, jeweils das Kasusmerkmal mit dem Wert Akkusativ auftritt.

Bewegung wird in der minimalistischen Konzeption durch die [$\pm$ stark]-Unterscheidung morphologischer und kategorialer Merkmale induziert, um wohlgeformte PF- und LF-Objekte abzuleiten. Syntaktische Bewegung wird damit auf Eigenschaften und die Parametrisierung des morphologischen Systems bezogen. Die in der syntaktischen Struktur auftretenden Merkmale werden aber während der Projektionsprozesse durch Perkolation und Vererbung an die relevanten syntaktischen Positionen transferiert. In der Struktur in (5) sind sowohl die beiden DPn des Komplementbereichs als auch die inhaltliche Spezifikation der funktionalen Kategorien aus einem ditransitiven Finitum, d.h. aus einer voll flektierten Lexikoneinheit, (und den X-bar-Bedingungen) ableitbar.

Ein Linking-Prozeß bezieht dann eine ungebundene Projektion über ein syntaktisches Linking-Merkmal auf die Position in einer unsaturierten Struktur, woraus eine komplexere Struktur als Resultat entsteht. Parallel dazu wird aber auch -je nach Linking-Relation- eine Θ -Rolle vergeben.

Diese Anmerkungen sollen dazu dienen, einen (mehr oder minder direkten) Bezug zwischen Kompetenz- und Performanztheorie anzudeuten, den ich hier aber nicht im Detail ausführen will. Was aber offensichtlich für Linking-Prozesse notwendig ist, sind die syntaktischen Merkmale, die eine Projektion P auf eine Position K in einer Struktur S beziehen; und es sind wesentlich diese Merkmale, die das Performanzsystem abhängig von der Rektionsrichtung der aktuell prozessierten Teilstruktur als Adresse für die zu integrierenden Projektionen verwendet.

3. Die Operationen des Parsers und die Verwendung grammatischer Kenntnis

Der soeben illustrierte Berechnungsmodus ist aufgrund der Äquivalenz von kontextfreien Grammatiken und Kellerautomaten für beliebig komplexe Strukturen anzuwenden. Der relevante Punkt liegt in der Art der Prozeßsteuerung des Automaten. Während ein Kellerautomat üblicherweise durch Tabellen gesteuert wird, die PS-Regeln enkodieren, wird die Struktur nun aufgrund der X-bar-theoretischen Eigenschaften syntaktischer Strukturen und der Eigenschaften lexikalischer Elemente ohne Regeln abgeleitet. Die vollständige Struktur ergibt sich damit aus dem auftretenden Wortmaterial und seiner Kombinatorik aufgrund der inhärenten Eigenschaften der beteiligten Lexeme, d.h. der Parser wird nur durch die in der Eingabe enthaltenen Lexikoneinheiten und deren Eigenschaften gesteuert, so daß er keine syntaktische Regelkomponente benötigt, die weitere Aktionen festlegt. Lediglich die strukturellen Konfigurationen der X-bar-Theorie werden zum lexemweisen Aufbau einzelner Projektionen herangezogen, und die kategorienspezifische Rektionsrichtung legt fest, ob der Parser die der aktuell berechneten Projektion zugehörigen Argumente im linken Kontext finden kann oder im rechten Kontext erwartet.

Nach dieser anschaulichen Betrachtung können die folgenden fünf elementaren Berechnungsaktionen unterschieden werden.

- (234) Aufbau einer Projektion
- (i) Linking der obersten Kellerzelle
 - (ii) Linking in die oberste Kellerprojektion
 - (iii) Speichern einer Projektion in der obersten Zelle
 - (iv) Oberste Zelle zum Linking zur Verfügung stellen

Welche dieser Aktionen aber in welchem Zustand ausgeführt werden muß, bedarf einer eingehenderen Betrachtung. So hatten wir in der Beispielanalyse (vgl. etwa (15)) bereits gesehen, daß die Vollständigkeit einer Projektion eine relevante Bedingung für Linking-Prozesse darstellt. Genauer gesagt ist diese Bedingung so zu formulieren, daß nur vollständige Projektionen in unvollständige Projektionen gelinkt werden dürfen.

Zum einen ergeben sich daraus verschiedene Probleme ergeben sich für Adjunkte, auf die ich hinweisen will, die ich hier aber nicht näher ausführen kann. Der kritische Punkt liegt darin, daß bei Adjunkten die externe Θ -Rolle des Modifikators vergeben wird. In der Terminologie Higginbothams (1985) findet Θ -Unifikation statt. Für die Linking-Prozesse bedeutet dies, daß der Modifikator unvollständig ist und demzufolge die Modifikanden-Projektion linken könnte. Dies würde zu einem Komplex vom Typ des Modifikators führen, was ausgeschlossen werden muß, da modifizierte Phrasen vom Typ des Modifikanden sind. Linking-Prozesse müssen daher neben dem Kriterium der Vollständigkeit auch Kriterien über die Art der Θ -Rollen-Vergabe heranziehen. Wiewohl das theoretische Konzept für die Konstruktionen klar ist, wie ich weiter unten erläutern werde, ist die Entscheidung im Verarbeitungsprozeß äußerst kompliziert, und ich möchte es bei diesen Ausführungen belassen.

Zum anderen erscheint die Vollständigkeitsbedingung problematisch zu sein, wenn man

Konstruktionen mit extrahierten Konstituenten betrachtet. Für Extraktionen in den linken Kontext muß dann verlangt werden, daß die bewegten Konstituenten on-line rekonstruiert werden und sich in ihrer Basisposition befinden, bevor die Domäne, aus der extrahiert wurde, gelinkt wird. Dies führt zu der Situation, daß Kettenbildung vom Kopf der Kette zu deren Fuß führt. Ein Vorschlag zu dieser Berechnungsweise findet sich in Lohnstein (1993:174ff). Noch problematischer sind allerdings Extrapositionen, bei denen die Möglichkeit zur on-line Rekonstruktion überhaupt nicht besteht. Auch diese Fragen bedürfen weiterer Klärung, die ich hier ebenfalls nicht vornehmen will.

Ein weiteres Entscheidungskriterium liefert die Rektionsrichtung der berechneten Kategorien, die dem Parser Informationen darüber liefern, wo er Argumente findet. Die Orientierung der aktuell zu berechnenden Projektion kann den Wert L(inks) oder R(echts) annehmen. Hat die Orientierung den Wert L(inks), so müssen sich die zugehörigen Argumente im linken Kontext befinden, und der Parser muß versuchen, diese dort zu finden. Im Falle einer kanonischen Realisierung befindet sich das als nächstes einzubindende Argument stets in der obersten Zelle. Hat die Orientierung den Wert R(echts), so weiß der Parser, daß sich ein zugehöriges Argument im rechten Kontext befinden und die aktuelle Projektion in die oberste Zelle gelegt werden muß. Die nächste vollständige Projektion muß dann jedoch genau dieses Argument sein.

Wenn wir die oberste Kellerzelle als lastLGL und die oberste Stufe der aktuell berechneten Projektion als topPEL bezeichnen, so sind nach jedem Projektionsschritt die Eigenschaften Vollständigkeit und Orientierung sowohl von topPEL als auch von lastLGL zu prüfen, wie die Matrix in (25) zeigt.

(25)

	TopPEL	LastLGL
Vollständigkeit	+ -	+ -
Orientierung	L R	L R

Vollständigkeit ist dabei im Sinne der Saturiertheit von Θ -Rastern zu verstehen, d.h. daß der Parser nach jedem Aufbau einer X-bar-Projektion prüfen muß, ob in der aktuell berechneten Projektion weitere Θ -Rollen zu vergeben sind. Wenn dies nicht der Fall ist, liegt eine vollständige (maximale) Projektion vor, die ihrerseits in eine Struktur integriert werden kann. Sind weitere Θ -Rollen zu vergeben, so wird der Projektionsprozeß iteriert, und weitere Linkingprozesse werden durchgeführt, bis die Projektion vollständig ist.

Eine Linking-Operation kann als eine dreistellige Relation zwischen dem obersten Knoten einer vollständigen Projektion, einer aufgebauten Position in einer Projektion und einem gemeinsamen Merkmal an beiden Positionen charakterisiert werden. Sie bezieht eine vollständige Projektion mittels eines Merkmals auf eine projizierte Position in einer Struktur. Um einen Linking-Prozeß auszulösen, ist es erforderlich, daß das Merkmal sowohl am obersten Knoten einer Projektion als auch an der projizierten Position mit gleichen Werten auftritt. Wenn dies der Fall ist, wird die Projektion in die projizierte Position integriert; anderenfalls werden

andere Prozesse ausgeführt.

Die verbalen Linkingmerkmale, die wir im folgenden eingehender betrachten wollen, sind im Deutschen die Merkmale Kasus, Status, wh. Die Spezifikation dieser Merkmale zeigt die Matrix in (26).

(26)

Merkmal	Werte			
Kasus	Nom	Gen	Dat	Akk
Status	1	2	3	
wh		+	-	

Um nun zu entscheiden, wann Linking-Prozesse stattfinden, benötigt der Parser drei Funktionen, deren Resultate festlegen, welche Aktionen im weiteren auszuführen sind. Die erste Funktion entscheidet, ob eine Projektion vollständig ist, die zweite berechnet die Orientierung des Kopfes, und die dritte Funktion prüft, ob ein Linking-Merkmal an zwei Positionen den gleichen Wert hat.

- (27)
- (i) Compl ist eine Funktion, die als Argument eine Projektion nimmt, prüft, ob diese vollständig ist und als Wert ein Element aus { true, false } zurückgibt.
 - (ii) RR ist eine Funktion, die als Argument eine Projektion nimmt, prüft, welche Orientierung der Kopf hat und als Wert ein Element aus { L(inks), R(echts) } zurückgibt.
 - (iii) Poss-Link ist eine Funktion, die eine Position in einer Projektion und eine vollständige Projektion als Argumente nimmt, und abhängig von den spezifizierten Linkingmerkmalen ein Element aus { true, false } zurückgibt, d.h. Poss-Link prüft, ob eine Linkingrelation zwischen einer Position und einer Projektion besteht.

Mit diesen drei Funktionen sind die Prozeßzustände auf jedem Terminalelement, d.h. hinsichtlich der Linearität, und auf jeder Projektionsebene, d.h. hinsichtlich der Hierarchie, eindeutig festgelegt, so daß in jeder Situation effektiv entschieden werden kann, welche Aktion aus (23) durchzuführen ist. Zu einer detaillierteren und illustrativeren Betrachtung der möglichen Fälle vgl. Lohnstein (1993: Kap. 7).

Eine willkommene Besonderheit dieses Berechnungsmodells besteht nun darin, daß der Parser die Linking-Bedingungen stets unter strikter Adjazenz der jeweils relevanten m-Kommando-Domänen prüft. Dies wird sichergestellt, weil die hinsichtlich der Linearisierung entfernt auftretenden Projektionen entweder in LGL in unteren Zellen liegen oder bereits in PEL in strukturelle Positionen integriert sind. Diese Verarbeitungsstrategie garantiert somit, daß überhaupt nur minimale Regenten für die strukturelle Integration maximaler Phrasen in Frage kommen. Regent und Regens können nur in PEL oder in der obersten Zelle von LGL auftreten und sind damit immer strikt adjazent. Wenn ein Integrationsprozeß stattgefunden hat, so ist die integrierte Projektion für andere Linking-Prozesse nicht mehr sichtbar, da stets nur die obersten Knoten konsultiert werden. Damit kann also nie ein anderer Regent in einer solchen

Konfiguration intervenieren. Dies hat nun den wünschenswerten Effekt, daß die Minimalitätsbedingung für Rektion (vgl. Chomsky 1986:42ff) automatisch erfüllt ist. Der Parser hat gar keine andere Möglichkeit, als nur minimale Regenten zur Berechnung zu betrachten, so daß sich argumentieren läßt, daß diese Bedingung aus den Eigenschaften des Performanzsystems abgeleitet werden kann.

Im folgenden Abschnitt werde ich darlegen, aus welchen Strukturkomponenten die lexikalischen Einheiten gebildet sind, und wie daraus syntaktische Strukturen aufgebaut werden können.

4. Die Struktur von Θ -Rastern und syntaktischen Projektionen

Bis hierher habe ich erörtert, wie ein psycholinguistisch motivierter Parser organisiert sein könnte. Dabei bin ich davon ausgegangen, daß nach Prozessen der Worterkennung zwei weitere Operationsweisen für die Strukturbildung wesentlich sind. Dies ist erstens die wortweise Projektion lexikalischer Information zu syntaktischen Strukturen und zweitens die Integration vollständiger syntaktischer Strukturen in andere, unvollständige syntaktische Strukturen. Ich will nun die Idee der wortweisen Projektion weiter konkretisieren und auf die Eigenschaften von Θ -Rastern und Argumentstrukturen von Lexikoneinheiten beziehen. Dabei folge ich Vorschlägen von Higginbotham (1985), Fukui/Speas (1986) und Bierwisch/Haider (1989). Die Zielvorstellung besteht darin, die Information, die mit einer phonologisch/graphematischen Marke im Lexikon verbunden ist, zu verwenden, um daraus die syntaktische Umgebung im Sinne der X-bar-Theorie abzuleiten. Dieser erste Schritt führt dazu, daß zu jedem in einem Satz auftretenden Wort dessen syntaktische Projektion berechnet werden kann. Die strukturelle Komposition dieser Projektionen folgt dann in weiteren Schritten, die wir anschließend betrachten werden.

4.1 X-bar-Theorie

Syntaktische Projektionen sind wesentlich festgelegt durch die in den X-bar-Projektionen auftretenden Eigenschaften der Köpfe. Die lexikalischen Kategorien N, A, V und P werden als Matrix der binären Merkmale $[\pm N, \pm V]$ beschrieben, deren syntaktische Struktur durch die X-bar-Theorie determiniert wird (vgl. Chomsky 1986b:3):

- (28) (i) $X' \rightarrow X^0 Y''$
(ii) $X'' \rightarrow Y'' X'$

Y'' in (28)(i) ist das Komplement und in (28)(ii) der Spezifikator von X. Die Komplemente sind nicht nur innerhalb der Rektionsdomäne des Kopfes zu realisieren, sondern müssen sich auch hinsichtlich ihrer links-rechts-Positionierung an dessen Rektionsrichtung orientieren. Daraus

folgt die periphere Positionierung des Phrasenkopfes. Für die sprach- und kategorienspezifische Rektionsrichtung (RR) gilt dann (29).

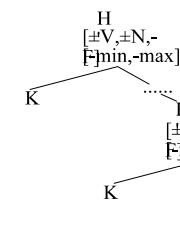
- (29) $RR \in \{ L(\text{inks}), R(\text{echts}) \}$

Komplemente sind damit hinsichtlich ihrer relativen Position zum Phrasenkopf abhängig von RR zu realisieren.

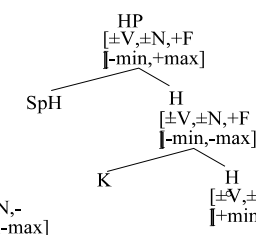
Daraus ergibt sich das folgende Bild für mögliche syntaktische Konfigurationen:

(30)

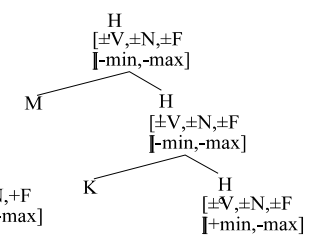
Komplementation:



Spezifikation:



Phrasenadjunktion:



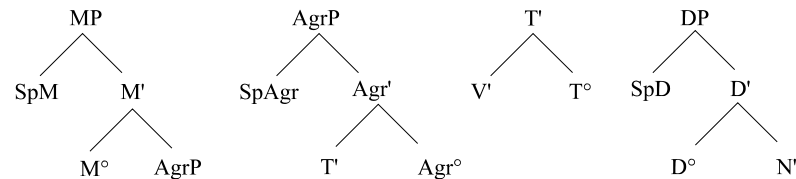
Die Abfolgebedingungen von Komplementen (K) bzw. Modifikatoren (M) und Kopf (H) sind in Abhängigkeit von der Rektionsrichtung des jeweils in Frage stehenden Phrasenkopfes festgelegt.

Die morphologisch motivierten funktionalen Kategorien AGR, T, C, DET weisen verschiedene Eigenschaften auf, die sie mit entsprechenden lexikalischen Kategorien teilen. So ist AGR sowohl nominal als auch verbal, T ist rein verbal, DET rein nominal, während C unabhängig von den typischen N- und V-Merkmalen ist und hinsichtlich seiner Eigenschaften für den Satzmodus als modal zu charakterisieren ist, mit starkem Bezug zu den verbalen T-Merkmalen. Nimmt man den Verbmodus als weitere flexionsmorphologisch motivierte funktionale Kategorie M^0 an, so ist aus der Information, die ein finites Verb liefert, eine vollständige CP-(bzw. MP-)Struktur projizierbar. In Lohnstein (1993) habe ich eine Analyse vorgeschlagen, in der der Verbmodus der funktionale Kopf eines Modussystems ist, das den Satzmodus in Abhängigkeit von der Besetzung der jeweiligen Kopf- und Spezifikatorposition festlegt. Die Verwendung lexikalischer C^0 -Elemente wird damit zu einem Spezialfall der Satzmodus-Indikation. Die Notation CP wird daher im folgenden konsequenterweise durch die Notation MP für Modusphrase ersetzt.

Der Komplementbereich ist aus der Hierarchie des Θ -Rasters und der Rektionsrichtung ableitbar. Die Projektion der funktionalen Kategorien ergibt sich aus f-selektionalen Bedingungen, wobei die funktionalen Köpfe ihren Inhalt aus den flexionsmorphologischen Charakterisierungen beziehen. Die Selektionseigenschaften dieser Kategorien führen den X-bar-theoretischen Annahmen gemäß zu den Teilstrukturen in (31) und weisen Funktionsmerkmale (F-Merkmale) zu, wie Nominativ-Kasus durch finites INFL, Genitiv-Kasus durch 's, $[\pm wh]$

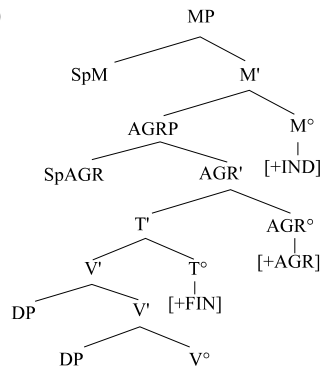
durch Comp. Diese Relationen lizenzieren die Spec-Positionen der jeweiligen funktionalen Kategorien (vgl. Fukui/Speas 1986).

(31)



M⁰ wird durch den Verbmodus, Agr⁰ durch die Merkmale Person und Numerus und T⁰ durch die Tempusmarkierung inhaltlich spezifiziert, so daß die Information, die mit einem Finitum bekannt wird, zu der Struktur in (32) projiziert werden kann.

(32)



Um nun unter den Bedingungen für die Sprachverarbeitung in (3) vollständige syntaktische X-bar-Strukturen wortweise aus der Lautkette zu rekonstruieren, müssen verschiedene Berechnungsprozesse durchgeführt werden. Erstens muß der lexemspezifische Komplementbereich aufgespannt werden. Dieser Schritt läßt sich aus dem Θ -Raster des in Frage stehenden Lexems und der Rektionsrichtung der Kategorie dieses Lexems bestimmen. Zweitens muß das Auftreten von Spezifikatoren und deren Serialisierung (relativ zum Kopf) berechnet werden. Dies läßt sich aus der spezifischen Eigenschaft funktionaler Kategorien herleiten. Drittens muß sichergestellt werden, daß die Θ -Rollen der einzelnen Lexeme mit den richtigen syntaktischen Argumenten belegt werden. Dies geschieht durch entsprechende Linking-Prozeduren. Viertens muß die Verbindung zwischen funktionalen Köpfen und deren Argumenten berechnet werden.

Diese Berechnungen können nun aber nicht derart vorgenommen werden, daß die syntaktischen Ebenen des T-Modells der Rektions- und Bindungstheorie gewissermaßen direkt

sichtbar werden. Der Grund dafür liegt in der Links-Rechts-Bedingung für Sprachverstehensprozesse. Andererseits muß die Repräsentation der thematischen Verhältnisse, d.h. die D-Struktur aber auf jeden Fall ausgedrückt werden. Um diesen Aspekt zu erfassen, will ich daher zunächst eine Konzeption der Struktur von Θ -Rastern und der Vergabe von Θ -Rollen erörtern, aus der die Eigenschaften der D-Struktur ableitbar sind. Diese Konzeption erlaubt es, gemeinsam mit der X-bar-Theorie vollständige D-Strukturen aus den Eigenschaften der Lexikoneinheiten zu spezifizieren.

4.2 Θ -Rollen-Vergabe

Folgt man Fukui/Speas (1986), so besitzen die lexikalischen Kategorien Θ -Raster, projizieren aber keine Spezifikator-Positionen, funktionale Kategorien hingegen projizieren Spec-Positionen, falls -qua lexikalischer Eigenschaft- Kase an diese Position zugewiesen wird. Diese Kategorien besitzen zwar selektionale Eigenschaften, jedoch sind diese nicht ohne weiteres mit Θ -Rollen zu identifizieren, da sie nicht die typischen Eigenschaften von Elementen der Argumentstruktur aufweisen.

Nach Higginbotham (1985:555-558) haben Verben und Nomina referentielle Θ -Rollen. Im Falle von Verben handelt es sich dabei um ein 'hidden argument' im Sinne von Davidson (1966), das üblicherweise als Variable e (für event) notiert ist. Nomina weisen ein referentielles Argument im Sinne von Williams (1980) auf. Für einfache Nomina ergibt sich daraus ein Θ -Raster wie in den spitzen Klammern (33)(i) und für ein transitives Verb wie in (33)(ii).

- (33) (i) (Hund, -V +N, <1>)
(ii) (küssen, +V -N, <1,2,e>)

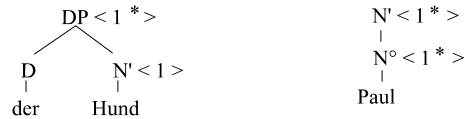
Diese beiden Typen von Θ -Rollen führen zu einem Konflikt mit dem in Chomsky (1981:36) angegebenen Θ -Kriterium. Dies hat seinen Grund darin, daß die Vergabe referentieller Rollen nicht ohne weitere Annahmen einer Funktor-Argument-Beziehung entspricht. Higginbotham (1985:561) hat daher den eindeutigen Bezug zwischen Θ -Rollen und syntaktischen Argumenten (vgl. Chomsky 1981:36) zu der der folgenden Formulierung in (34) abgeschwächt, um verschiedene Varianten der Θ -Rollen-Vergabe zu ermöglichen.

- (34) (i) Every thematic position is discharged.
(ii) If X discharges a thematic role in Y, then it discharges only one.

Die Vergabe von Θ -Rollen kann damit für die verschiedenen Typen von Θ -Rollen auf unterschiedliche Weise vorgenommen werden.

Referentielle Θ -Rollen werden durch Θ -Bindung vergeben, wie in (35) dargestellt. Der Asterisk kennzeichnet vergebene Θ -Rollen.

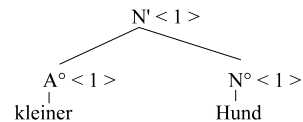
(35)



Dabei wird die referentielle Rolle des Nomens von einem D⁰-Element gebunden, während inhärent referentielle Eigennamen diese Rolle überhaupt nicht aufweisen.

Modifizierende Ausdrücke, wie etwa attributive Adjektive, adverbiale PPn usw. sind semantisch als einstellige Prädikate zu analysieren. Insofern besitzen auch sie Θ -Rollen, die in bestimmter Weise vergeben werden müssen, um die Funktor-Argument-Struktur dieser Ausdrücke zu repräsentieren. Das semantische Korrelat syntaktischer Modifikation ist die Θ -Unifikation. Dabei wird die externe Rolle des Modifikators vergeben bzw. mit der Rolle des Modifikanden unifiziert, vgl (36).

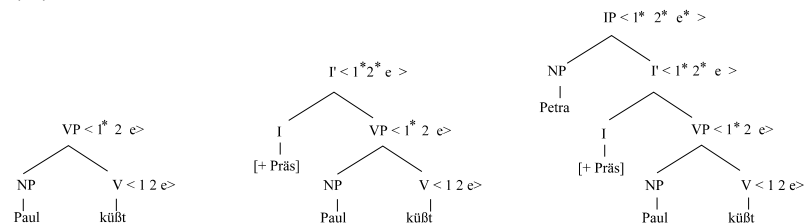
(36)



In dieser Konfiguration wird die externe Θ -Rolle des attributiven Adjektivs vergeben, indem sie mit der referentiellen Rolle des Nomens unifiziert wird.

In nicht-modifizierten verbalen Projektionen findet Θ -Markierung im traditionellen Sinne statt. Die referentielle Rolle des Verbs wird hingegen analog zur nominalen Projektion durch INFL gebunden, wie in (37).

(37)



Bierwisch (1986) hat die Art und Anzahl der Valenzstellen auf die Semantische Form der Lexikoneinheiten bezogen und die jeweiligen Θ -Rollen als λ -Operatoren repräsentiert. Das Θ -Raster wird damit zu einer Hierarchie von λ -Operatoren, die die Hierarchie der syntaktischen Argumente festlegt. Die Integration der syntaktischen Argumente als Agens, Thema,

Experiencer usw. geschieht dann durch die semantischen Teilprädikate, deren Argumente die λ -abstrahierten Variablen sind. Das hat zur Konsequenz, daß die eben genannten Etikettierungen für Θ -Rollen (Agens, Thema usw.) für syntaktische Prozesse überhaupt keine Rolle spielen, sondern gewissermaßen semantische Reflexe bei der Interpretation sind. Die Serialisierung der Argumente unterliegt dem Richtungsparameter für Rektion. Damit sind nun die beiden Strukturaspekte, Hierarchie und Linearisierung, die durch Phrasenstrukturbäume ausgedrückt werden, vollständig festgelegt: Die Hierarchie der Argumente ist aus dem lexemspezifischen Θ -Raster ableitbar, während die Linearisierung durch die kategorienspezifische Rektionsrichtung bestimmt ist.

Die Positionen im Θ -Raster unterscheiden sich hinsichtlich ihres Status als Argumente und ihrer syntaktischen Realisierung. Für alle lexikalischen Kategorien ist λx_0 die externe Θ -Rolle, für die Kategorie V ist λx_0 die referentielle Θ -Rolle, für die Kategorie N ist λx_0 zugleich externe wie referentielle Rolle, für P und A ist x_0 das externe Argument. Daraus ergibt sich das allgemeine Schema für Θ -Raster (vgl. Bierwisch 1986):

$$(38) \quad \begin{array}{lcl} X^0=V: & \lambda x_n \lambda x_{n-1} \dots & \lambda x_1 \quad \lambda x_0 \\ X^0 \neq V: & \lambda x_n \lambda x_{n-1} \dots & \lambda x_0 \end{array} \quad \begin{array}{l} \text{Komplemente} \\ \text{externes/referentielles Argument} \end{array}$$

Die Realisierung interner Argumente kann optional sein. Für die Kategorien N und A sind alle internen Argumente optional, vgl. (39).

- (39) (i) Tyson ist [_{AP} jedem Gegner überlegen]
(ii) Tyson ist [_{AP} ϕ überlegen]
(iii) [_{DP} Cauchys Beweis des Integralsatzes] ist von fundamentaler Bedeutung
(iv) [_{DP} Der Beweis] ist von fundamentaler Bedeutung

Referentielle und externe Argumente sind hingegen nie optional (vgl. Bierwisch 1986). Für die Kategorie V ist Optionalität nicht generell möglich, sondern eine lexikalisch zu spezifizierende Eigenschaft, wie etwa bei fakultativ transitiven Verben.

- (40) (i) Peter liest einen Roman
(ii) Peter liest

Θ -Rollen können daher durch drei Merkmale mit binären Werten spezifiziert werden.

- (41) (i) referentiell: [+, -]
(ii) intern: [+, -]
(iii) optional: [+, -]

Basierend auf diesen Unterscheidungen lassen sich nun die verschiedenen Vergabe-Prinzipien von Θ -Rollen i.S. Higginbothams (1985) auf Prinzipien der semantischen Komposition beziehen. Dies hat verschiedene interessante Konsequenzen für die Berechnung semantischer

Komplexe, die nun parallel zur Berechnung der syntaktischen Struktur geschehen kann. Da lexikalischer Zugriff (Worterkennung) die lexikalische Information zu einem Lexem liefert, stehen zur Berechnung der syntaktischen Projektion insbesondere das Θ -Raster und die semantische Form dieser Lexeme zur Verfügung. Wenn syntaktische Teilstrukturen über Linking-Prozeduren gebildet werden, so werden parallel dazu auch Θ -Rollen vergeben. Um nun gleichzeitig mit der syntaktischen Strukturbildung auch die zugehörige semantische Interpretation zu berechnen, müssen wir die den syntaktischen Linkingprozessen entsprechenden semantischen Operationen betrachten. Dazu ist es notwendig, allen syntaktischen Konfigurationen semantische Entsprechungen zuzuordnen. Den drei in (30) unterschiedenen Relationen, Komplementation, Spezifikation und Adjunktion entsprechen auf semantischer Seite drei Operationen, die sich in der Art der Θ -Rollen-Vergabe unterscheiden.

Der syntaktischen Θ -Markierung entspricht der semantische Prozeß der λ -Konversion.

- (42) (i) [Der Bär] tanzt
 (ii) $\lambda x \lambda e [e \text{ INST } [x \text{ TANZ}]]$ [DEF BÄR]
 λ -Konversion: $\lambda e [e \text{ INST } [DEF \text{ BÄR TANZ}]]$

Dabei wird das semantische Korrelat eines syntaktischen Arguments mit der semantischen Struktur einer anderen Projektion kombiniert. Dieser Prozeß entspricht der gewöhnlichen Argument-Anbindung. Die Rekonstruktion dieser Funktor-Argument-Beziehung geschieht auf syntaktischer Seite durch die Identifikation eines Merkmals der Argumentstruktur des Prädikats mit einem Merkmal des Arguments (etwa Kasus). Auf semantischer Seite entspricht die Argument-Anbindung der Vergabe einer Θ -Rolle des Prädikats. Ein Linking-Prozeß erfaßt immer diese beiden Aspekte der Strukturbildung. Für den syntaktischen Linking-Prozeß ist dabei nur das Linking-Merkmal relevant, während die Vergabe der Θ -Rolle als semantischer Reflex dieses Prozesses aufzufassen ist.

Θ -Unifikation ist das semantische Pendant zur modifikationellen Adjunktion. Die Beziehung zwischen einem Modifikanden und einem Modifikator wird hergestellt, indem die externe Θ -Rolle des Modifikators vergeben wird, während die Rolle des Modifikanden erhalten bleibt. Die Vergabe einer Θ -Rolle unter Modifikation entspricht auf syntaktischer Seite der Adjunktions-Konfiguration. Unter Θ -Unifikation wird die externe Rolle des Modifikators mit der zugehörigen Variablen des Modifikanden in der folgenden Weise unifiziert.

- (43) $\lambda x [\dots x \dots]$ $\lambda y [\dots y \dots]$
 Θ -Unifikation: $\lambda x [[\dots x \dots] : [\dots x \dots]]$

Wenn $\lambda x [\dots x \dots] =: P(x)$ der Modifikand und $\lambda y [\dots y \dots] =: Q(y)$ der Modifikator ist, so ergibt sich unter Θ -Unifikation die Interpretation $P(x)$ derart, daß $Q(X)$, wobei ':' ein propositionaler Konnektor ist, der als derart, daß paraphrasiert werden kann, und der im wesentlichen der logischen Konjunktion entspricht. Für eine Adjunktionsstruktur wie Bär im Garten ergibt sich somit die Ableitung (44), wobei Θ_i die externe Rolle der Präposition darstellt:

- (44) (i) [Bär] [Θ_i im Garten]
 (ii) $\lambda x [Bär \ x]$ $\lambda y [y \text{ LOC } [INT \text{ GARTEN}]]$
 $\rightarrow \lambda x [[Bär \ x] : [x \text{ LOC } [INT \text{ GARTEN}]]]$

Der Ausdruck in der zweiten Zeile von (44)(ii) bezeichnet dann die Menge aller x für die gilt, daß x ein Bär derart ist, daß sich die Lokalität von x im Inneren des Gartens befindet. Dabei bleibt die referentielle Rolle λx des Nomens erhalten, da sie ihrerseits bei der Verbindung mit einem D^0 -Element vergeben wird. Diese Operation ist die folgende.

Θ -Bindung spezifiziert die Beziehung zwischen einer funktionalen Kategorie und einer referentiellen Θ -Rolle. Da nur Verben und Nomina referentielle Rollen besitzen, sind auch nur diese referentiell verwendbar. Die funktionalen Kategorien I^0 und D^0 binden jeweils die referentiellen Rollen von Verben und Nomina. Dabei wird der Referenzindex der funktionalen Kategorie auf die zu bindende Kategorie übertragen, wie in (45).

- (45) $\lambda Y [\dots Y \ x \dots]$ $\lambda z [\dots z \dots]$
 λ -Konversion: $[\dots \lambda z [\dots z \dots] \ x \dots]$
 Θ -Bindung: $[\dots [\dots x \dots] \dots]$

Den funktionalen Kategorien ist daher ein SF-Muster der folgenden Art zuzuweisen:

- (46) (i) /der/ $\lambda Y [DEF \ Y \ x]$
 (ii) /daß/ $\lambda Y [DAB \ Y \ x]$

Daraus ergibt sich im Falle referentieller Bindung mit einem Nomen (etwa Bär/ $\lambda x [Bär \ x]$) die folgende Struktur:

- (47) /der Bär/ $\lambda Y [DEF \ Y \ z]$ $\lambda x [Bär \ x]$
 λ -Konversion: $[DEF \ \lambda x [Bär \ x] \ z]$
 Θ -Bindung: $[DEF [Bär \ z]]$

Der referentielle Index (hier als Variable z repräsentiert) wird dabei auf das referentielle Argument des Nomens übertragen. Die NP-Bedeutung in der letzten Zeile von (47) ist nun ein vollständig saturierter Ausdruck, da alle Θ -Rollen vergeben sind.

Mit diesen drei Operationen, die parallel zu den jeweiligen Linking-Prozessen stattfinden, wird die semantische Struktur also gewissermaßen simultan zur syntaktischen Struktur berechnet. Das Wissen um die semantischen Kategorien der Argumente reicht jedoch zur Berechnung der Strukturen nicht aus. Was dazu als Kenntnis weiterhin zur Verfügung stehen muß, sind die morphosyntaktischen Merkmale in der Argumentstruktur. Erst mithilfe dieser Markierungen können im Falle mehrstelliger Verben die einzelnen DP-Argumente unterschieden werden.

4.3 Morphologische Merkmale in der Argumentstruktur

Die Struktur des Θ -Rasters legt zwar Komplementpositionen fest, doch sind die Kategorie und die morphosyntaktische Realisierung dieser Argumente, die z.T. idiosynkratisch festgelegt werden müssen, nicht spezifiziert. Die einzelnen Θ -Rollen sind daher um eine Komponente, die diese Merkmale enthält, zu erweitern. Diese erweiterte Struktur, d.h. Θ -Rollen mit syntaktisch motivierten Merkmalen, bildet die Argumentstruktur. Die Elemente der Argumentstruktur sind als Positionen im Θ -Raster aufzufassen, die jeweils ein Paar $(\lambda x_i, F_i)$, bestehend aus einer Θ -Rolle und einer Menge syntaktischer Merkmale, bilden (vgl. Haider/Bierwisch 1989). Das Θ -Raster selbst ist die hierarchisch geordnete Sequenz der λ -Operatoren. Sie kann aufgefaßt werden als die Schnittstelle zwischen syntaktischer und semantischer Struktur. In den folgenden Unterabschnitten wollen wir die Linking-Merkmale Kasus, wh und Status betrachten.

4.3.1 Kasustheorie und Kasusmerkmale

Die Kasuspezifikation von NP-Argumenten ist wesentlich bestimmt von den Zuweisungseigenschaften der sie regierenden lexikalischen oder funktionalen Kategorien. Nominativzuweisung findet dabei stets durch AGR statt; Genitivzuweisung ist für einige Verben und Präpositionen lexikalisch markiert; innerhalb der DP geschieht sie strukturell. Akkusativ und Dativ werden von Verben, Adjektiven und Präpositionen regiert.

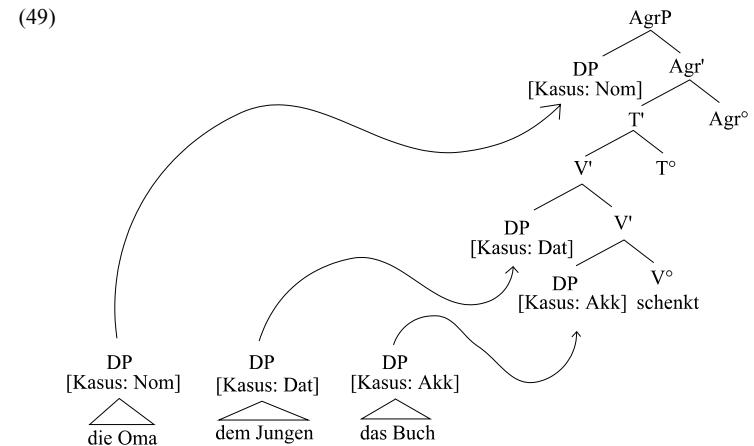
Zwei Arten von Kasus sind zu unterscheiden: abstrakter vs. morphologischer Kasus (vgl. Haider 1985). Abstrakter Kasus bezeichnet dabei eine Position in einer syntaktischen Konfiguration, während sich morphologischer Kasus auf die Realisierung von Flexionsmorphemen an DPn bezieht. Beide Arten von Kasus sind für die Argument-Anbindung relevant. Während abstrakter Kasus in der Argumentstruktur markiert ist und damit an gewisse Positionen P_i projiziert wird, dient morphologischer Kasus an den DPn selbst der Zuordnung eines Arguments an diese Position. Die Erkennung und Zuordnung der Argumente an die in SF auftretenden λ -gebundenen Variablen wird damit durch Identifikation der syntaktischen Merkmale realisiert. Die Prinzipien der Kasuszuweisung sind dabei festgelegt wie folgt (vgl. Chomsky 1981:170):

- (48) (i) NP is nominative if governed by AGR
(ii) NP is objective if governed by V with the subcategorization feature: $[_{NP}]$ (i.e., transitive)
(iii) NP is oblique if governed by P
(iv) NP is genitive in $[_{NP} _ X']$
(v) NP is inherently case-marked as determined by properties of its $[-N]$ governor

(48)(i) bis (48)(iv) spezifizieren die Standardfälle der Kasuszuweisung, während (48)(v) idiosynkratische Fälle erfaßt.

In einem finiten Satz wird die Nominativ-NP in SpAGR realisiert und bleibt in der

Argumentstruktur ohne spezifische Kasusmarkierung, da diese ableitbar ist. Analog zur Kasuszuweisung in syntaktischen Konfigurationen läßt sich (48) auf die anderen Positionen in der Argumentstruktur übertragen. Aus der Argumentstruktur werden somit die Kasuspositionen in der syntaktischen Struktur gekennzeichnet, zugleich müssen diese Merkmale aber auch an den kasustragenden Phrasen selbst auftreten, um für Linking-Prozesse sichtbar zu sein. (49) zeigt die Linking-Relationen für ein bitransitives Verb und die Adressierung der syntaktischen Argumente an die Θ -Positionen.



Die Pfeillinien illustrieren, an welche Positionen die kasusmarkierten DPn integriert werden müssen. Deren Identifikation geschieht durch die morphologische Kasusmarkierung an der maximalen Projektion. Die Positionen, in die die DPn jeweils gelinkt werden, erhalten ihren Adreßcharakter aufgrund ihrer Kasusmarkierung in der Argumentstruktur. In gewissem Sinne werden die syntaktischen Argumente erst mit dieser Markierung für die Θ -Positionen sichtbar.

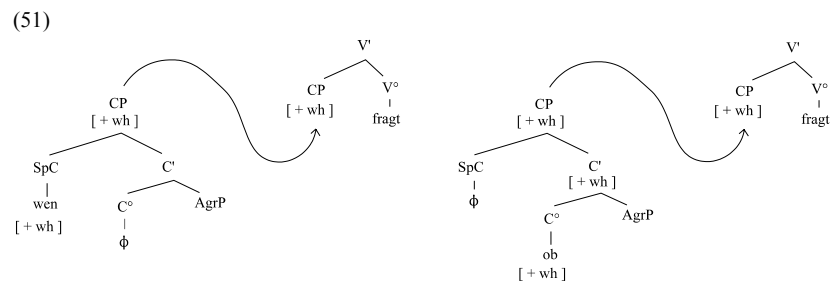
4.3.2 wh-Merkmale

Die Eigenschaften finiter sententialer Argumenttypen werden meist über die Spezifikation des Merkmals $[\pm \text{wh}]$ vorgenommen (vgl. etwa v. Stechow/Sternefeld 1988:385). Fürs Deutsche ergeben sich dabei die folgenden Verteilungen dieses Merkmals:

- (50) (i) $[+ \text{wh}]$:
- Peter fragt, ob er Maria küssen darf.
 - Peter fragt, wen er küssen darf.
 - *Peter fragt, daß er Maria geküßt hat.
- (ii) $[- \text{wh}]$:
- Peter behauptet, daß er Maria geküßt hat.
 - *Peter behauptet, ob er Maria geküßt hat.
 - *Peter behauptet, wen er geküßt hat.
- (iii) $[\pm \text{wh}]$:
- Peter weiß, daß er Maria geküßt hat.
 - Peter weiß, ob er Maria geküßt hat.
 - Peter weiß, wen er geküßt hat.

Die Distribution von ob und w-Phrase einerseits und daß andererseits wird vom Matrixverb festgelegt. (50)(i)a bis (50)(i)c zeigen, daß das Verb fragen ein Argument selektiert, das für das Merkmal $[+ \text{wh}]$ spezifiziert sein muß. Behaupten in (50)(ii)a bis (50)(ii)c spezifiziert dieses Argument als $[- \text{wh}]$, während das propositionale Argument von wissen beide Merkmale erlaubt. Obwohl die Datenlage weitaus komplexer ist und durch die einfache Selektionsmarkierung $[\pm \text{wh}]$ nicht vollständig erfaßt werden kann, wollen wir diese Notation hier als Approximation verwenden.

Das wh-Merkmal wird durch das C⁰-Element oder die vorangestellte w-Phrase in der SpC-Position geliefert (vgl. Chomsky 1981, Grewendorf 1988, Reis/Rosengren 1988). Für die syntaktischen Linkingprozesse ergeben sich damit die folgenden Konfigurationen am Beispiel $[+ \text{wh}]$ -markierter Komplementsätze:



Die Pfeillinien deuten wiederum an, welche Positionen jeweils aufeinander zu beziehen sind.

Verwendet man diese Notation, so sind die satzeinleitenden Elemente bezüglich der $[\pm \text{wh}]$ -Spezifikation zuzuordnen, wie in (52).

- (52) (i) ob $\rightarrow [+ \text{wh}]$
(ii) daß $\rightarrow [- \text{wh}]$
(iii) w-Phrase $\rightarrow [+ \text{wh}]$

Die Selektion der derart satztypspezifizierten Argumente ist nun in die Argumentstruktur der jeweiligen Matrixverben -analog zur Repräsentation der Kasusmarkierungen- aufzunehmen, wie in (53), wobei v eine Variable für propositionale Argumente ist.

- (53) (i) /frag/ $\lambda v \lambda y \lambda x \lambda e[e \text{ INST } [x \text{ FRAG } y \ v]]$
 $[+ \text{wh}]$
(ii) /behaupt/ $\lambda v \lambda x \lambda e[e \text{ INST } [x \text{ BEHAUPT } v]]$
 $[- \text{wh}]$
(iii) /wiss/ $\lambda v \lambda x \lambda e[e \text{ INST } [x \text{ WISS } v]]$
 $[\pm \text{wh}]$

Die Erörterung in diesem Unterabschnitt verwendet wieder die CP-Notation und sieht von den Eigenschaften des Modusystems und seiner Spezifikation ab. Der Grund für diese Betrachtungsweise liegt schlicht darin, daß auch die Modusystem-Analyse zu ähnlichen Repräsentationen führt, wenngleich ihre Herleitung eine andere ist. Die traditionelle CP-Analyse wird hier also nur aus Gründen der Bekanntheit und Einfachheit verwendet.

4.3.3 Status-Merkmale

Gegenüber der $[\pm \text{wh}]$ -Spezifikation finiter Komplemente sind infinite Komplemente in der Argumentstruktur durch ein Status-Merkmal markiert.

Die infinite Verbmorphologie im Deutschen weist die flexionsmorphologischen Merkmale Modus, Agr und Fin nicht auf. Sie zeigt jedoch Abhängigkeiten zwischen finiten und infiniten Verben einerseits und infiniten und anderen infiniten Verben andererseits, die analog zu den Rektionsverhältnissen zwischen Verben und kasustragenden Elementen erfaßt werden können. Das dem Kasus entsprechende Merkmal ist der Status (vgl. Bech 1983²:15). Bech hat eine umfassende Analyse deutscher Infinitivkonstruktionen vorgelegt, derzufolge die Eigenschaften der infiniten Verbalformen durch eine zweidimensionale Matrix dargestellt werden können, die zwei Stufen und drei Status unterscheidet (vgl. Bech 1983²:12):

- (58) (i) Lexikoneinheiten zu syntaktischen Strukturen projiziert,
(ii) unlizenzierte Teilstrukturen speichert,
(iii) Projektionspositionen auf bestehen Teilstrukturen bezieht, indem er
(iv) Linking-Merkmale prüft und bei Übereinstimmung
(v) komplexe Strukturen aufbaut.

Wir haben diese Organisationsform bereits in Abschnitt 3 informell erörtert und wollen sie nun in die Theorie der formalen Sprachen und Automaten einbetten.

In der Theorie der formalen Sprachen hat die Hierarchie der Chomsky-Grammatiken einen zentralen Stellenwert. Diese Grammatiken und die durch sie erzeugten Sprachfamilien stehen in isomorpher Beziehung zu einer Hierarchie von Automaten, die jeweils genau die von einem bestimmten Grammatiktyp erzeugten Sprachen erkennen.

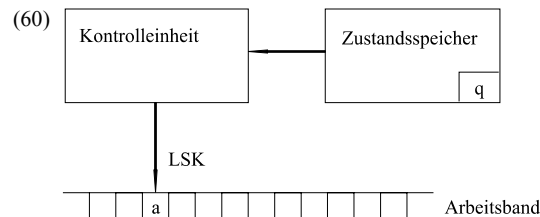
Abhängig vom abstrakten Format der Regeln können drei Grundtypen von Grammatiken G unterschieden werden, die hinsichtlich des Umfangs der von ihnen erzeugten Sprachen ein Inklusionsverhältnis aufweisen: kontextsensitive, kontextfreie und rechtslineare. Tatsächlich gibt es mehr Abstufungen in dieser Hierarchie, für unsere Zwecke genügen aber zunächst diese drei.

Die Menge der nach einer dieser Grammatiken ableitbaren Ketten ist die Sprache $L(G)$. Für die o.g. Grammatiktypen ergeben sich Sprachen, die in dem folgenden Inklusionsverhältnis stehen:

$$(59) L(\text{Typ-3}) \subset L(\text{Typ-2}) \subset L(\text{Typ-1})$$

$L(G)$ besteht aus der Menge all derjenigen Ketten K, für die gilt, daß die Wörter von K Lexikoneinheiten sind und daß K aus einem Startsymbol S (für Satz) mittels G abgeleitet werden kann. Nun kann man einerseits fragen, welche Ketten eine Grammatik zu erzeugen gestattet. Dies ist das Problem der Generierung. Eine andere Frage besteht darin, zu einer gegebenen Kette zu entscheiden, ob sie zu einer Sprache gehört und welche Struktur dieser Kette entspricht. Es ist also zu entscheiden, ob die Kette K mittels G abgeleitet werden kann. Das ist das Problem des Parsings.

Zu jeder der drei unter (59) genannten Sprachen existiert jeweils ein Automat, der genau diese Sprache erkennen kann. All diesen verschiedenen Automaten liegt das Modell der Turingmaschine TM zugrunde. Graphisch veranschaulicht sieht eine Turingmaschine wie in (60) aus.



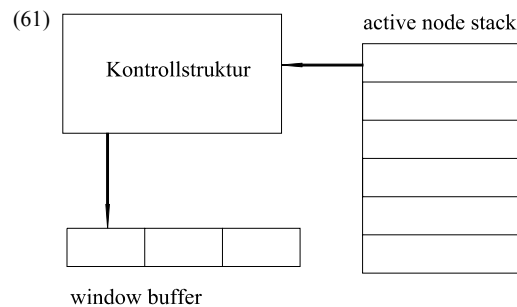
Sie besteht aus einem potentiell unendlich langen Arbeitsband AB, einem Lese-Schreibkopf LSK, einer Kontrolleinheit KE und einem Zustandsspeicher ZS. Sie führt bei der Berechnung Schritte der folgenden Art durch: Bewege den LSK, Lies einen Feldinhalt, Schreibe einen Feldinhalt usw. Dabei bezieht sie sich auf die Information im Zustandsspeicher und in der Kontrolltabelle. Wenn z.B. ein Symbol auf dem Band gelesen wurde, so befindet sich die Turingmaschine in einem bestimmten Zustand q. Eine Folgeoperation wird abhängig von diesem Symbol und dem Zustand q ausgeführt, indem etwa ein anderes Symbol rechts oder links auf das Band geschrieben wird. Dadurch geht die Turingmaschine in einen anderen Zustand über. Die Berechnung einer Funktion durch eine Turingmaschine besteht also wesentlich darin, einen Zustand Z_i in Abhängigkeit vom gelesenen Wort in den Zustand Z_{i+1} zu überführen und gegebenenfalls Schreiboperationen (nach links bzw. rechts) auf dem Band vorzunehmen. Die Anzahl der zur Berechnung erforderlichen Überführungsschritte eines Zustands in einen anderen kann dabei als Metrik für die Zeitkomplexität der Berechnung gewählt werden. Die Anzahl der Felder, die während der Berechnung beschrieben werden, dient als Komplexitätsmaß für den Speicherbedarf. Dabei hängt der Grad der jeweiligen Komplexität i.d.R. von der Länge der Eingabe ab. Je länger die Eingabe ist, umso höher ist die Komplexität. Nun ist aber nicht nur die Eingabelänge zur Bestimmung des Komplexitätsgrades relevant, da ansonsten bei unterschiedlich 'schwierigen' Problemen gleicher Länge stets die gleiche Komplexität auftreten würde. Die Komplexität ergibt sich daher aus dem Verhältnis zwischen Schwierigkeit des Problems und Eingabelänge, d.h. sie ist eine nach dem Schwierigkeitsgrad des Problems zu spezifizierende Funktion der Eingabelänge. Das Steigungsverhalten dieser Funktion gibt dann Auskunft über die Schwierigkeit des Problems. Nun läßt sich für bestimmte Klassen von Problemen zeigen, daß ihre Lösung bei hinreichend langer Eingabe nicht in einer Zeit berechnet werden kann, die etwa dem Alter unseres Universums entspricht, obwohl die Lösung prinzipiell berechenbar ist (vgl. Barton/Berwick/Ristad, 1987). Die Steigung von Komplexitätsfunktionen kann daher für Probleme, die der menschliche Parser offensichtlich sehr schnell löst, nicht allzu hoch sein. Spezieller gesagt, müssen die Komplexitätsfunktionen effizienter Berechnungen durch polynomiale Funktionen der Länge der Eingabe nach oben beschränkbar sein. Wenn sich nun für einen Algorithmus zeigt, daß dessen Komplexitätsfunktion schneller ansteigt als eine polynomiale Funktion der Eingabelänge, so ist die Berechnung nicht effizient. Speziell für Algorithmen zur Modellierung von Sprachverarbeitungsprozessen ergibt sich daraus die Notwendigkeit ihrer Effizienz, denn der menschliche Parser, so scheint es, ist äußerst effizient.

Durch bestimmte Restriktionen oder Erweiterungen kann die Berechnungsfähigkeit der Turingmaschine verändert werden. So ist es z.B. möglich, nicht nur mit einem Band zu arbeiten, sondern etwa mit zwei Bändern; es wäre auch denkbar, daß auf dem einen Band nur gelesen werden darf, während auf dem anderen Band gelesen und geschrieben werden darf usw. Uns interessiert im folgenden eine Turingmaschine mit zwei Bändern, die beschränkte Lese- und Schreibmöglichkeiten hat. So haben wir bereits bei der Betrachtung der anfänglichen Beispielanalyse gesehen, daß das Lese-Schreibband, welches wir als LGL bezeichnet haben nur auf der obersten Zelle gelesen bzw. beschrieben werden darf und daß das Band, welches wir als PEL bezeichnet haben, nur aus einer Zelle besteht. Derart konstruierte Turingmaschinen werden in der Automatentheorie als deterministische Kellerautomaten bezeichnet, wobei der hier erörterte Parser eine Unterklasse dieses Automatentyps ist.

5.1 Der deterministische Kellerautomat

Der Klasse der kontextfreien (Typ 2-) Grammatiken ist der deterministische Kellerautomat (deterministic push down automaton) (DPDA) äquivalent, d.h. dieser Automat erkennt genau die kontextfreien Sprachen. Da X-bar-Strukturen kontextfrei konstruiert sind, scheint dieser Automat das adäquate theoretische Instrument zu sein, um diese Strukturen zu berechnen. Ein DPDA kann aufgefaßt werden als on-line-Turingmaschine mit zwei Bändern, die bzgl. ihrer Schreib- und Lesemöglichkeiten beschränkt ist.

Das abstrakte Modell des Marcus-Parsers (Parsifal) (vgl. Marcus 1980) entspricht einem solchen DPDA. Dabei ist ein Zugriff im Kellerspeicher, dem sog. active node stack (ANS), nur auf das oberste und das jeweils unterste Kellersymbol möglich, während das Eingabeband, ein sog. window buffer (WB) von beschränkter Länge, dafür aber mit beliebigem Zugriff vorgesehen ist. Der Kellerautomat speichert alle Konstituenten, die während der Analyse unvollständig sind und wartet mit deren Integration, bis weitere Information zur Verfügung steht. Das Leseband ist auf drei Zellen beschränkt, und die Kontrollstruktur steuert die Aktionen des Automaten in den jeweiligen Prozeßzuständen.



In der Kontrollstruktur sind die verschiedenen Konfigurationen des WB definiert. Üblicherweise geschieht diese Festlegung durch eine (meist sehr große) Menge spezifischer Regeln, die alle möglichen Konfigurationen des WB enthalten muß. Eine Konfiguration von WB besteht aus einer möglichen Abfolge von drei Wörtern bzw. Phrasen. Die Regeln sind als sog. 'pattern-action-rules' formuliert, und der Parser aktiviert während der Verarbeitung ganze Regelpakete. Wenn bestimmte Muster (Wort- bzw. Konstituentenabfolgen) in WB auftreten, werden entsprechende Aktionen gestartet. Eine solche Aktion wäre etwa der Aufbau einer neuen Phrasenstruktur durch die Funktion CREATE. Dabei orientiert sich der Parser aber nicht nur an dem aktuell auftretenden Wort, sondern er hat die Möglichkeit, auch lexikalisches oder phrasales Material zu betrachten, welches diesem Wort folgt. Die Aktionen sind damit abhängig von dem Inhalt des WB. Das aber bedeutet, daß die Regeln letztlich jede mögliche Drei-Wort-Abfolge als WB-Muster enthalten müssen, so daß die Anzahl der Regeln astronomisch groß

wird. Es ist nun klar, daß derartige Kontrollstrukturen nicht prinzipienorientiert sind, wenngleich die Idee dieser Verarbeitungsstrategie attraktiv ist. Genau an diesem Punkt setzt die Konzeption des hier erörterten Parsers an. Auf spezielle Regeln, die bei bestimmten Konfigurationen aktiviert werden, wird vollständig verzichtet. Vielmehr soll der Parser nur abhängig von dem auftretenden lexikalischen Material arbeiten. Wie wir bereits gesehen haben, ist dieser Berechnungsmodus durchaus möglich. Die Frage ist nur noch, wie die einzelnen Rechenschritte relativ zueinander und abhängig von der Eingabekette serialisiert werden müssen.

Die Besonderheit eines deterministischen Kellerautomaten besteht in der Art der Verwaltung gespeicherter Symbole. Diese funktioniert nach dem Stapelprinzip last in - first out, d.h. daß das zuletzt auf den Stapel gelegte Element als erstes wieder entfernt werden muß, bevor tiefer liegende Elemente zugänglich sind. Wir werden im folgenden einen Kellerautomaten betrachten, dessen Stapel eine beschränkte Tiefe von sieben Einheiten aufweist. Diese Beschränkung approximiert in gewisser Weise die Kapazität des menschlichen Arbeitsgedächtnisses (vgl. Miller 1956). Tatsächlich läßt sich zeigen, daß Zentraleinbettungen bis zum Grad zwei bis drei mit einem sieben Zellen langen Arbeitsband strukturierbar sind, während höhergradige Einbettungen mehr Speicherplatz erfordern. Interessanterweise ist aber auch der menschliche Parser bei Einbettungsgraden größer als drei nicht mehr in der Lage, (adäquate) Strukturen aufzubauen.

Wir können den während der Verarbeitung sich stets neu etablierenden linken Kontext (LK) einer Äußerung mit den beschriebenen Zellen des DPDA und den Prozeßzeitpunkt (PZP) mit der Zelle, auf der der DPDA aktuell arbeitet, identifizieren. Der rechte Kontext ist im PZP unbekannt. Dies ist nun genau die in Abschnitt 1 anvisierte Zielvorstellung eines Automaten, der die Situation des menschlichen Parsers während der Verarbeitung simuliert.

5.2. LR(k)-Parser

Eine spezielle Unterklasse der deterministischen Kellerautomaten bilden die sog. LR(k)-Parser (vgl. Knuth 1965). Dies sind Erkennungsautomaten, die von links nach rechts (LR) arbeiten und k Symbole auf dem Eingabeband vorausschauen, um die Strukturierung vorzunehmen, d.h. der Parser erfüllt die Bedingung (3)(i). Er schreitet dabei wortweise von links nach rechts durch die Eingabekette, schaut k Symbole vom aktuellen Prozeßzeitpunkt nach vorne und entscheidet anhand dieser Information, welche Aktionen auszuführen sind. LR(k)-Parser besitzen ein Komplexitätsverhalten von einer Größenordnung, die linear proportional zur Länge der Eingabe ist, d.h. daß die Komplexitätsfunktion von einem Polynom ersten Grades majorisiert wird (vgl. Berwick/Weinberg 1984:192, Hopcroft/Ullman 1988:271). Im Hinblick auf diese Effizienz sind sie geeignete Modelle des menschlichen Sprachverarbeitungssystems. Unter der Zusatzbedingung, daß der Parser on-line arbeitet, ist $k = 0$ zu setzen, da keine Vorausschau über mehrere Terminalelemente notwendig ist, um Strukturen aufzubauen. Der Parser arbeitet also nur auf dem gerade gelesenen Symbol und verwendet die damit verbundene lexikalische Information, um Entscheidungen hinsichtlich seiner Aktionen zu treffen. Ein kritischer Punkt

liegt dabei in der Struktur der Kontrolltabelle, die im Regelfall durch eine Menge von PS-Regel-analogen Spezifikationen gegeben ist, die auf dem Vokabular der syntaktischen Kategorien operiert und dem Parser die Bedingungen nennt, unter denen ein Zustandswechsel stattfinden muß.

Die Struktur der Kontrolltabelle wird jedoch nicht durch eine fixierte Menge von PS-Regeln festgelegt, sondern basiert auf der Idee, daß die in der Terminalkette auftretenden lexikalischen Einheiten und die damit verbundenen Informationen selbst die Steuerung des Parsers übernehmen, d.h. der Parser erfüllt auch die Bedingung (3)(ii). Dabei operiert er auf allen Einheiten der Lautkette in gleicher Weise, d.h. er muß entscheiden, welche Aktion aus (23) durchgeführt wird. Die Entscheidung über die konkrete Berechnung liefert ihm die mit dem jeweiligen Lexem verbundene Lexikoninformation. Auf einer relativen abstrakten Ebene führt der Parser auf jedem Lexem identische Prozeduren aus, deren Berechnungsergebnis aber von der Struktur des jeweiligen Lexems festgelegt wird. Die Struktur der zugehörigen Projektionen (ableitbar aus Θ -Raster und Rektionsrichtung) legt die Domänen fest, innerhalb derer Argumente und Adjunkte realisiert sein müssen. Abhängig von den Rektionsrichtungen der jeweils projizierenden Kopfkategorie zeigt der Parser dann aber ein unterschiedliches Verhalten hinsichtlich der Integration von Argumenten. Damit reagiert der Parser aber gerade so, wie wir eingangs erörtert haben: regiert ein Kopf nach rechts, so muß er auf folgende Argumente warten; regiert ein Kopf nach links, so müssen im linken Kontext die Argumente, die die Projektion auffüllen, identifiziert und in die Struktur integriert werden.

5.3 Datenstrukturen

Um diese Berechnungsweise zu implementieren, sind mindestens zwei Datenstrukturen erforderlich. Eine wird benötigt, um nicht-lizenzierte Konstituenten zwischenspeichern. Der Zugriff auf diese Konstituenten ist jedoch nicht arbiträr, sondern erhält die Reihenfolge des Auftretens dieser Konstituenten. Insbesondere muß die Auffindbarkeit dieser Konstituenten effektiv möglich sein. Die Suchprozesse sind dabei so organisiert, daß die zuletzt abgelegten Konstituenten zuerst konsultiert werden. Diese Art der Verwaltung entspricht einem Stack (oder einer linear geordneten Liste (LGL)). Den Inhalt von LGL bilden die zum PZP nicht lizenzierten Konstituenten des linken Kontexts unter Beibehaltung ihrer Reihenfolge.

Neben der als Zwischenspeicher konzipierten LGL muß der Parser über eine Datenstruktur verfügen, die die Projektion zum aktuellen Prozeßzeitpunkt aufnimmt. In dieser Struktur werden die Projektion eines Phrasenkopfes sowie die aus LGL gelinkten Phrasen verwaltet. Sie enthält stets nur eine Struktur, die aber aufgrund von Linking-Prozessen komplexer werden kann. Diese Datenstruktur ist als erweiterbare Liste zu konzipieren, die von der Projektionsfunktion während des Verarbeitungsprozesses monoton erweitert werden kann. Diese Liste wird als projektiv erweiterbare Liste (PEL) bezeichnet. Sie ist in ihrer Kapazität durch die Anzahl der Elemente des Θ -Rasters und die Anzahl der von den morphosyntaktischen Merkmalen etablierten funktionalen Köpfe beschränkt.

Die serielle Organisation der beiden Prozeßtypen Projektion und Linking ist nun wesentlich dafür verantwortlich, wie hoch die Kapazität des Zwischenspeichers anzusetzen ist. Eine

Prozeßstrategie, die zunächst nur Projektionen aufbaut, erfordert eine hohe Speicherkapazität, da alle Teilprojektionen verwaltet werden müssen. Insbesondere zieht dies für Sätze mit mehr als 5-7 Phrasenköpfen eine Kapazitätsüberlastung nach sich, da sämtliche Teilprojektionen zu speichern sind, LGL aber auf sieben Einheiten beschränkt ist.

Die Serealisierung der Prozesse ist also so vorzunehmen, daß die Anzahl der Teilprojektionen gering bleibt. Dazu sind die Projektions- und Linking-Prozesse in eine gemischte Sequenz zu bringen, d.h. daß während der Projektion bereits Linking-Prozesse stattfinden müssen. Dies führt dazu, daß schnellstmögliche Komplexbildung bei gleichzeitiger Speicherreduktion stattfindet, d.h. der Parser erfüllt die Bedingung (3)(iv).

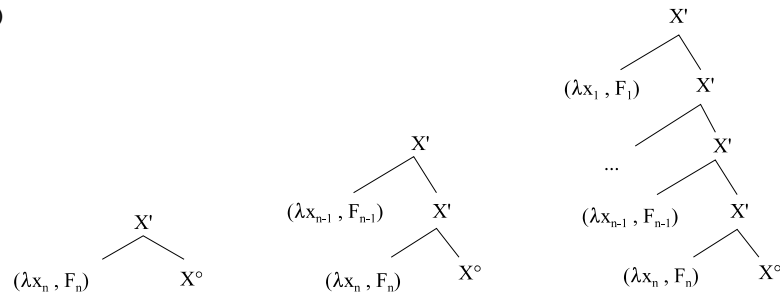
Der so konzipierte Parser arbeitet sowohl hinsichtlich der Zeitkomplexität (LR(0)-Eigenschaft) als auch hinsichtlich der Speicherkomplexität (Beschränkung von LGL auf sieben Einheiten) effizient, d.h. der Parser erfüllt auch die Bedingung (3)(iii).

6. Die Projektion der Argumentstruktur und das Linking syntaktischer Argumente

Das bisher erörterte System behandelt die Bedingungen, unter denen eine vollständige Satzstruktur einer Terminalkette mit kanonischer Realisierung zugewiesen wird. Wir betrachten nun, wie die in (27) angegebenen Funktionen mit der Projektionsfunktion interagieren.

Die Projektionsfunktion ist eine Funktion, die als Argument eine X-bar-Teilprojektion der Stufe n ($0 \leq n$) nimmt und als Wert eine X-bar-Teilstruktur der Stufe $n+1$ zurückgibt. Der neu aufgebauten Position wird eins der Linkingmerkmale aus (26) zugewiesen. Dieses spezielle Merkmal und sein zugehöriger Wert ist Teil der Argumentstruktur des projizierenden Lexems. Im Falle $n = 0$ nimmt die Projektionsfunktion als Argument ein (morphologisch möglicherweise komplexes) Element des Lexikons. In diesem Fall liegt ein unsaturierter Ausdruck vor, d.h. es wurden bis zu diesem Zeitpunkt keine Argumentstellen abgebunden. Die Argumentstruktur bildet eine Hierarchie von Zwei-Tupeln $(\lambda x_i, F_i)$, deren zweites Element jeweils eine Menge grammatischer Merkmale-Wert-Paare ist. Gemäß der durch die Hierarchie festgelegten Sequenzierung der Argumente werden diese sukzessiv von außen nach innen projiziert. Für die Sequenz der internen Argumente $(\lambda x_n, F_n) \dots (\lambda x_1, F_1)$ eines Kopfes X^0 werden die Strukturen abhängig von der Rektionsrichtung dieses Kopfes aufgebaut. Falls X^0 nach links regiert, führt dies über die folgenden Schritte zu sukzessiven Projektionen wie in (62).

(62)

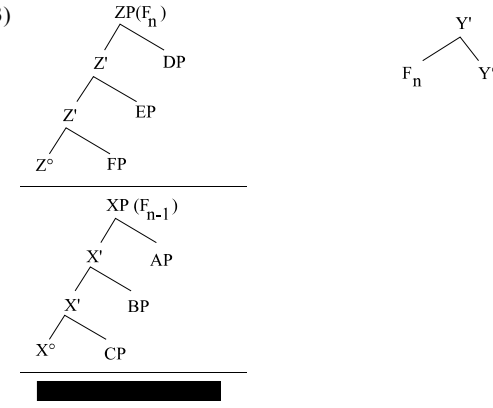


Da die Projektion dieser Argumentstellen schrittweise geschieht, hat der Parser nach jedem Projektionsschritt die Möglichkeit, zu prüfen, ob das oberste Kellersymbol die Merkmale F_i der jeweils projizierten Position aufweist. Wenn dies der Fall ist, so bindet er die Struktur in der obersten Zelle direkt in die Projektion ein. Wird also gerade die Projektion eines finiten ditransitiven Verbs aufgebaut, so befindet sich das akkusativ-markierte direkte Objekt in der obersten Zelle und wird unter der ersten Projektionsstufe V' strukturell integriert. Damit verschwindet es aus dem Kellerspeicher, und das indirekte Objekt liegt in der obersten Zelle. Wenn nun das zweite Element der Argumentstruktur projiziert wird, d.h. die Position für das indirekte Objekt, so benötigt der Parser wiederum nur einen Blick in die oberste Kellerzelle, um auch jetzt richtigerweise das indirekte Objekt dort vorzufinden. Es ist auch in diesem Fall die Kasusmarkierung, die dem Parser das Kriterium zum Linking liefert.

Diese Berechnungsart ist jedoch nur für linksorientierte Projektionen relevant, da im Falle der Rechtsorientierung die Argumente erst folgen, d.h. zum aktuellen Prozeßzeitpunkt noch unbekannt sind. Das hat zur Konsequenz, daß linksorientierte Maximalprojektionen stets vollständig sind, während rechtsorientierte Maximalprojektionen stets unvollständig sind. Vollständigkeit zu einem gewissen Prozeßzeitpunkt ist daher eine Funktion der Rektionsrichtung des Kopfes.

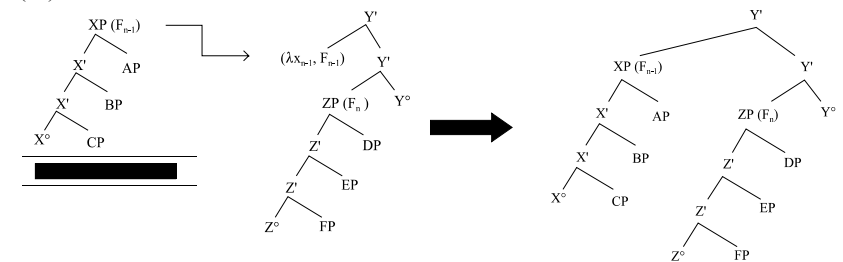
Betrachten wir nun, wie eine Projektion sukzessiv vollständig wird. Für einen linksorientierten Kopf Y^0 mit der Argumentstruktur $(\lambda x_n, F_n) \dots (\lambda x_1, F_1)$ muß der linke Kontext organisiert sein wie in (63).

(63)



Wenn die beiden Projektionen XP und ZP vollständig sind, so befinden sich die F_i aufgrund der Perkolation an den jeweiligen maximalen Projektionen. Von diesen F_i fungiert ein Merkmal als Linking-Merkmal. Die Anwesenheit dieses Merkmals und die Übereinstimmung seines Wertes an der Maximalprojektion und der projizierten Position löst nun verschiedene Prozesse aus. Erstens wird die zugehörige Projektion aus der obersten Zelle in die Struktur integriert, womit das darunter befindliche Argument das oberste Element im Speicher wird, wie in der folgenden Konfiguration (64).

(64)



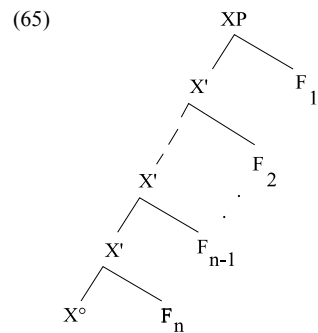
Zweitens wird mit diesem syntaktischen Linking-Prozeß zugleich eine Θ -Rolle vergeben, was drittens auf der semantischen Ebene mit der in Abschnitt 4 beschriebenen Operationen der Θ -Rollen-Vergabe einhergeht. Die Anbindung von Argumenten wird dabei zunächst durch die Merkmale der Argumentstruktur gesteuert, hat als weitere Konsequenz, daß auch eine semantische Struktur berechnet wird.

Obwohl die jeweiligen Vergabeprinzipien formal relativ klar formuliert sind, ist die aktuelle Berechnung von Adjunktionsstrukturen mit erheblichen Problemen verbunden. So scheint es, daß ohne Rekurs auf semantische Linking-Bedingungen adverbiale Modifikation

nicht zu berechnen ist, da keine syntaktischen Merkmale vorhanden sind, die einen Linking-Prozeß auslösen könnten. Andere Modifikationsarten hingegen, wie etwa im Falle pränominaler adjektivischer Adjunkte, können auf ein DP-internes Kongruenzsystem bezogen werden, innerhalb dessen referentielle Bindung stattfindet. Welche Kriterien der Parser dazu aber jeweils verwendet, bleibt zu untersuchen.

Da wir nun in der Lage sind, den Aufbau einer Projektion schrittweise für jede Argumentstelle separat zu vollziehen, können wir die beiden Prozesse Projektion und Linking in eine gemischte Sequenz bringen. D.h. wir können nun zunächst eine Argumentposition projizieren, sodann die Linking-Merkmale prüfen und das zugehörige Argument gegebenenfalls strukturell integrieren, um dann die nächste Argumentposition zu projizieren usw. Mit diesem Verfahren kann in Beispiel (63) während des Projektionsprozesses zunächst ZP aus LGL entfernt und in die Projektion Y' integriert werden, wie in (64). Zugleich wird damit XP oberstes Kellersymbol. Im folgenden Projektionsschritt wird die nächste Argumentposition aufgebaut und XP als nächstes Argument eingebunden usw. Wenn alle Positionen mit Argumenten gefüllt sind, ist YP vollständig. In PEL befindet sich sodann eine vollständige Projektion, die nun ihrerseits den Linking-Bedingungen unterliegt.

Rechtsorientierte Projektionen werden aufgebaut, ohne daß Linking-Relationen geprüft werden. Wenn Y⁰ rechtsorientiert ist, ergibt sich aus sukzessiver Projektion die Struktur in (65).



Diese wird oberstes Kellersymbol und wartet auf Maximalprojektionen, die die gleichen F_i aufweisen. Wenn diese auftreten, ist lastLGL unvollständig, und es wird geprüft, ob eine der Positionen in YP gleiche Merkmale F_i aufweist wie die in PEL auftretende vollständige Projektion, so daß jede in PEL vollständige Projektion in lastLGL integriert wird. Dieser Prozeß wird fortgesetzt, bis lastLGL vollständig ist. Sodann wird lastLGL nach PEL versetzt, und der Algorithmus iteriert auf den weiteren Lexikoneinheiten im rechten Kontext.

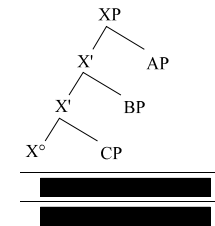
Eine interessante Konfiguration liegt vor, wenn in der obersten Zelle eine rechtsorientierte unvollständige Projektion auftritt, während gerade eine linksorientierte Projektion aufgebaut wird. Ein solcher Fall liegt etwa bei Extraktion aus Präpositionalphrasen vor, d.h. bei sog. preposition stranding wie in (66).

(66) *[Diesem Haus]_i hat er [in t_i] gewohnt.

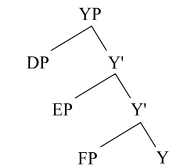
Diese Konstruktion ist im Deutschen ungrammatisch, nicht aber im Englischen. Wie läßt sich dieser Effekt aus der Architektur des Parsers und des Parameters für die Rektionsrichtung ableiten? Insbesondere ist ja interessant, an welcher Stelle die Ungrammatikalität des Satzes (66) diagnostiziert werden kann. Dies ist offensichtlich die Position, an der sie Spur eingesetzt wird, d.h. zwischen der Präposition und dem Verb.

Setzt man X := P und Y := V so tritt zum Zeitpunkt der Projektion von gewohnt eine Konstellation wie in (67) auf. In diesem Fall sind beide Projektionen unvollständig.

(67) LGL (unvollständig):

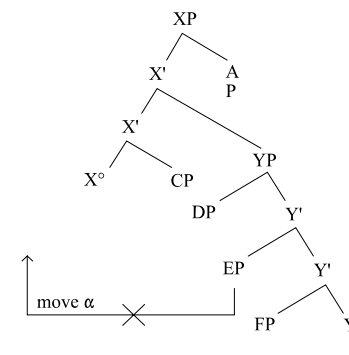


PEL (unvollständig):

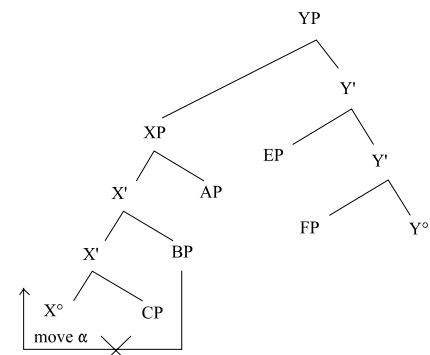


In dieser Situation scheint der Parser zunächst zwei Operationen ausführen zu können. Entweder wird YP in XP integriert oder XP in YP. Daraus ergibt sich dann entweder die Struktur in (68)a) oder die Struktur in (68)b).

(68) a)



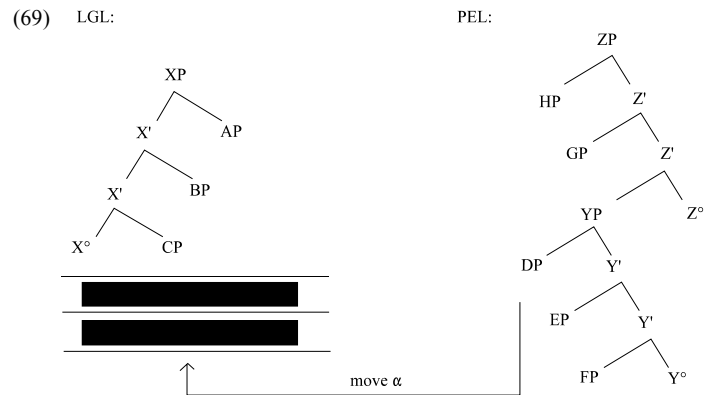
b)



Da nun aber sowohl YP in (68)a) als auch XP in (68)b) unvollständig sind, muß in beiden Fällen Extraktion aus der jeweils eingebetteten Phrase stattgefunden haben. Es liegen mit (68)a) und (68)b) aber zwei Konfigurationen vor, die gegen Kaynes (1984) Connectedness Condition (CC) bzw. Kusters (1987) Bedingung der Globalen Harmonie (BGH) verstoßen, da aus einer Domäne

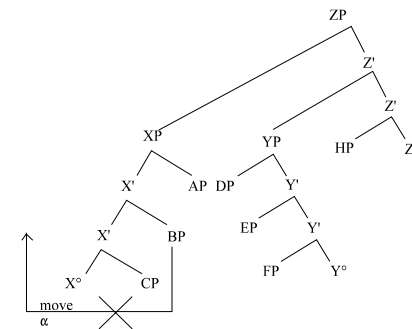
extrahiert wurde, die keinen Teilbaum aus gleich orientierten Projektionen bildet. Dies liegt daran, daß XP links- und YP rechtsorientiert ist, beide Projektionen aber zu einer zusammenhängenden Struktur verrechnet wurden. Dies zeigt zum einen, daß Vollständigkeit zum Linking notwendig ist, und es zeigt zum anderen, daß die extrahierte DP nicht on-line rekonstruiert werden kann, da die Präposition nach rechts das Verb aber nach links regiert.

Nun könnte jedoch der Fall eintreten, daß die Projektionen XP und YP gar keinen Teilbaum bilden. Dann nämlich, wenn ein folgender linksorientierter Kopf beide Projektionen integriert. Dies entspräche der Konfiguration in (69).

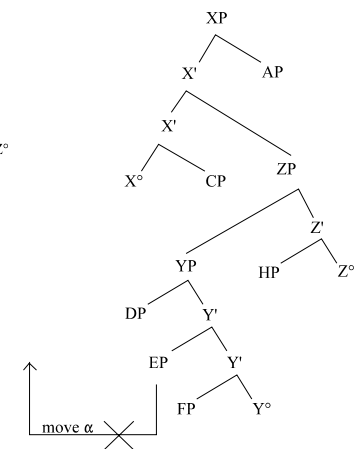


Damit bildet ZP einen Teilbaum gleichorientierter Projektionen und Extraktion aus YP ist nach CC bzw. BGH möglich. Jetzt stellt sich aber erneut die Frage, ob ZP in XP integriert wird oder XP in ZP, d.h. es tritt im Prinzip die gleiche Situation auf, die auch schon in (67) vorlag, daß nämlich eine unvollständige rechtsorientierte Projektion XP einer unvollständigen linksorientierten Projektion ZP gegenübersteht. Als resultierende Strukturen kommen daher nur (70)a) und (70)b) in Frage.

(70) a)



b)



Beide Male ergeben sich aber wieder Teilbäume mit nicht gleich orientierten Projektionen, so daß Extraktion aus den Einbettungen nicht möglich ist. Aus dieser Konfiguration ist also prinzipiell kein Entkommen mehr möglich, ganz unabhängig davon, wie der rechte Kontext aussieht.

Da als Entscheidungskriterien für die syntaktische Strukturbildung die beiden Kriterien Vollständigkeit und Orientierung verwendet werden, kann der Parser also unter strikt lokalen Bedingungen die Unzulässigkeit solcher Extraktionen feststellen bzw. er kann derartige Strukturen gar nicht berechnen. Es läßt sich daher wie folgt argumentieren. CC bzw. BGH sind keine unabhängigen Prinzipien des grammatischen Systems, sondern sie sind aus der funktionalen Architektur des Performanzsystems und den Kriterien, die der Parser stets zur Strukturbildung benötigt, abzuleiten. M.a.W. liefert der so konzipierte Parser eine funktionale Erklärung für ein vermeintlich grammatisches Prinzip. CC bzw. BGH sind daher aus diesen beiden Kriterien und der funktionalen Architektur des Performanzsystems unter strikt lokalen Bedingungen berechenbar.

Man beachte, daß die Berechnung von preposition stranding im Englischen problemlos möglich ist, da dort P und V die gleiche Orientierung haben. Für den Parser entsteht also kein Konflikt, wenn die Rektionsrichtung für V als R(echts) parametrisiert ist. Er hat damit offensichtlich die Möglichkeit, die extrahierte DP entlang der gleich orientierten Projektionen in die Basisposition zu transferieren, vgl. (71).

(71) What are we ending this chapter with?

Der bemerkenswerteste Effekt des LR(0)-Parsers besteht darin, daß die Entscheidungen über die Strukturbildung stets in benachbarten Domänen stattfinden. Das Performanzsystem greift dabei zeitlinear auf das Lexikon zu, und es werden voll flektierte Lexikoneinheiten für den syntaktischen Prozessor geliefert. Die Lexikoneinheiten liefern mit dem Θ -Raster die Information über die Struktur des Komplementbereichs in Abhängigkeit von der Rektionsrichtung der in Frage stehenden Kategorie. Nach links regierende Kategorien finden ihre Komplemente im linken Kontext, d.h. diese sind in S_i bereits bekannt und befinden sich im Speicher; nach rechts regierende Kategorien erwarten ihre Komplemente im rechten Kontext, d.h. diese sind in S_i unbekannt und das Performanzsystem muß die Kategorie speichern, um auf folgende Komplemente zu warten. Das Speichersystem selbst ist organisiert wie ein Keller-automat. Damit wird dem Faktum Rechnung getragen, daß syntaktische Konstruktionen kontextfrei berechenbar sind. Zugleich wird damit aber erreicht, daß (linear) lange Abhängigkeiten in einem bestimmten Zustand S_i wieder strikt adjazent auftreten, indem sich die abhängigen Elemente in PEL und lastLGL befinden. Die relevanten Bereiche von PEL und lastLGL sind dabei jeweils die m-Kommando-Domänen der projizierenden Köpfe. Zwischen diesen und nur diesen werden syntaktische Linkingbedingungen geprüft. Das Performanzsystem ist damit auf die Bedingungen minimaler Rektion festgelegt, d.h. nur minimale Regenten können überhaupt als Regenten auftreten.

Die Konzeption des Linkings von Argumenten in bestimmte Positionen erfordert notwendigerweise Prüfprozeduren in bestimmten Domänen. Diese Domänen sind in dem hier erörterten Performanzsystem die m-Kommando-Domänen von PEL und lastLGL. Wenn in diesen Domänen keine Merkmale geprüft werden können, brechen die Berechnungen zusammen bzw. bleiben bestimmte Konstituenten syntaktisch isoliert. Linking-Prozesse weisen dabei wesentliche Züge von generalisierten Transformationen auf. Ein Linking-Prozeß integriert eine vollständige Projektion P in die Position K einer Projektion P', wenn an P und K gleiche Merkmale auftreten, wobei diese Merkmale aus Spezifikator-Kopf-Kongruenz und Kopf-Komplement-Relation abgeleitet sind. Zugleich hängt Linking von den flexionsmorphologischen und kategorialen Charakterisierungen der beteiligten Köpfe ab.

Diese Eigenschaften des Performanzsystems zeigen interessante Parallelen zum Minimalistischen Programm (Chomsky 1992), so daß es scheint, daß Begriffe der Kompetenz- und Performanztheorie kompatibel werden.

Literatur

- Barton, G. E./Berwick, R. C./Ristad, E. S. (1987): Computational Complexity and Natural Language. Cambridge, Mass.: MIT Press.
- Bech, G. (1955/57) Studien über das deutsche verbum infinitum. Historisk-filologiske Meddelelser udgivet af Det kongelige Danske Videnskabernes Selskab. Band 35,2 (1955) und 36,6 (1957).
- Berwick, R./Abney, St. P./Tenny, C. (eds.) (1991): Principle-Based Parsing: Computation and Psycholinguistics. Dordrecht: Kluwer Academic Publishers.
- Berwick, R. C./Weinberg, A. (1984): The Grammatical Basis of Linguistic Performance: Language Use and Acquisition. Cambridge, Mass.: MIT Press.
- Bierwisch, M. (1983a): Semantische und konzeptuelle Repräsentation lexikalischer Einheiten. In: Motsch, W./ Růžicka, R. (eds.): Untersuchungen zur Semantik. (= Studia grammatica 22). Berlin: Akademie-Verlag, S. 61-99.
- Bierwisch, M. (1983b): How On-line is Language Processing? In: Bierwisch, M. (1983): Essays in the Psychology of Language. (= Linguistische Studien A 114). Berlin: Akademie der Wissenschaften der DDR.
- Bierwisch, M. (1986): Cognitive Linguistics: Framework and Topics. Ms. Max Planck Institut für Psycholinguistik, Nijmegen.
- Chomsky, N. (1981): Lectures on Government and Binding. Dordrecht: Foris.
- Chomsky, N. (1986): Barriers. (= Linguistic Inquiry Monograph 13). Cambridge, Mass.: MIT Press.
- Chomsky, N. (1992): A Minimalist Program for Linguistic Theory. MIT Occasional Papers in Linguistics, Number 1. Cambridge, Mass.: MIT Press.
- Chomsky, N. (1994): Bare Phrase Structure. MIT Occasional Papers in Linguistics, Number 5. Cambridge, Mass.: MIT Press.
- Davidson, D. (1966): The Logical Form of Action Sentences. In: Rescher, N. (ed.): The Logic of Decision and Action. Pittsburgh, PA: University of Pittsburgh Press, S. 81-95.
- Fodor, J. (1983): The Modularity of Mind. Cambridge, Mass.: MIT Press.
- Fong, S. (1991): The Computational Implementation of Principle-Based Parsers. In: Berwick, R./Abney, St. P./Tenny, C. (eds.): Principle-Based Parsing. Dordrecht: Kluwer Academic Publishers, S. 65-83.
- Fukui, N./Speas, M. (1986): Specifiers and Projections. MIT Working Papers in Linguistics, Vol. 8, 128-172.
- Grewendorf, G. (1988): Aspekte der deutschen Syntax: Eine Rektions-Bindungs-Analyse. (= Studien zur deutschen Grammatik 33). Tübingen: Niemeyer.
- Haider, H. (1985): The Case of German. In: J. Toman (ed.): Studies in German Grammar. Dordrecht: Foris, S. 65-101.
- Haider, H./Bierwisch, M. (1989): Steuerung kompositionaler Strukturen durch thematische Informationen. Finanzierungsantrag, Sonderforschungsbereich 340: Sprachtheoretische Grundlagen für die Computerlinguistik. Universität Stuttgart, 65-90.
- Higginbotham, J. (1985): On Semantics. Linguistic Inquiry 16.4, 547-593.
- Kayne, R. (1984): Connectedness and Binary Branching. Dordrecht: Foris.
- Knuth, D. E. (1965): On the Translation of Languages From Left to Right. Information and Control 8, 607-639.
- Kolb, H.-P. (1991): Monotonicity and Locality: Problems of Principle Based Parsing. Zeitschrift für Sprachwissenschaft 10.2, 161-190.
- Koster, J. (1987): Domains and Dynasties. Dordrecht: Foris.
- Lohnstein, H. (1993): Projektion und Linking. Ein prinzipienbasierter Parser fürs Deutsche. (= Linguistische Arbeiten 287). Tübingen: Niemeyer.
- Marcus, M. (1980): A Theory of Syntactic Recognition for Natural Language. Cambridge, Mass.: MIT-Press.

- Marslen-Wilson W./Tyler, L. K. (1980): The temporal structure of spoken language understanding. In: Cognition 8, 1-71
- Miller, G. A. (1956): The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. Psychological Review 63, 81-97.
- Reis, M./Rosengren, I. (1988): Wh-Imperatives?! Sprache und Pragmatik 10, 1-49.
- Stechow, A. v./Sternefeld, W. (1988): Bausteine syntaktischen Wissens. Ein Lehrbuch der generativen Grammatik. Opladen: Westdeutscher Verlag.
- Frauenfelder, U./Tyler, L. K. (1987): Spoken Word Recognition. Cambridge, Mass.: MIT Press.
- Williams, E. (1980): Predication. Linguistic Inquiry 11.1, 203-238.