Hanno Wulz

Formalismen einer Übersetzungsgrammatik

FORSCHUNGSBERICHTE DES

INSTITUTS FÜR DEUTSCHE SPRACHE

MANNHEIM

herausgegeben von

Ulrich Engel und Gerhard Stickel

Schriftleitung: Eva Teubert

Band 46

HANNO WULZ

Formalismen einer Übersetzungsgrammatik

gnv Gunter Narr Verlag · Tübingen

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Wulz, Hanno:
Formalismen einer Übersetzungsgrammatik / Hanno Wulz. — Tübingen: Narr, 1979.
(Forschungsberichte des Instituts für Deutsche Sprache; Bd. 46)
ISBN 3 - 87808 - 646 - 6

Druck: Müller+Bass, Tübingen
Printed in Germany

ISBN 3 - 87808 - 646 - 6

I n h a l t s v e r z e i c h n i s

0. VORBEMERKUNG

0.1 Zum Thema der Arbeit

In den letzten Jahren mehren sich in der Sprachforschung die Versuche, formallogische Mittel zur Explizierung der Semantik natürlichsprachlicher Äußerungen zu verwenden. Vereinfacht geht es dabei darum, eine formale Kunstsprache zu konstruieren, zu der eine (modelltheoretische) Semantik angebbar ist. Da dadurch die Bedeutung der kunstsprachlichen Formeln festgelegt ist, kann die Bedeutung einer natürlichsprachlichen Formulierung durch die Angabe ihres kunstsprachlichen Äquivalents expliziert werden.

Von besonderer praktischer Relevanz sind solche Arbeiten zur formalen Semantik in denjenigen Bereichen der linguistischen Datenverarbeitung, in denen nicht nur Buchstabenketten verarbeitet werden, sondern in denen es vor allem auf die mit natürlichsprachlichen Formulierungen vermittelten Informationen ankommt. Die von der Computerverarbeitung bedingte formale Darstellung solcher Informationen führt zu ähnlichen Problemen wie im Bereich der formalen Semantik: die Entwicklung formaler Sprachen, die idealiter die gleiche Ausdrucksmächtigkeit haben wie eine natürliche Sprache.

Die Notwendigkeit der Darstellung von Informationen, die in natürlichsprachlichen Äußerungen formuliert sind, in einer formalen Sprache zum Zwecke der maschinellen Informationserschließung, der automatischen Problemlösung oder der Computersimulation von Sprachverstehen bedingt jedoch nicht nur die Frage nach den erforderlichen Eigenschaften und der geeigneten Form einer solchen formalen Sprache. Ist eine solche formale Sprache gefunden, dann erfordert eine Operationalisierung auf dem Computer darüberhinaus auch eine Explizierung des r e g e l h a f t e n Zusammenhangs zwischen der natürlichen Sprache und dieser formalen Repräsentationssprache. Dies bedeutet, daß Regeln anzugeben sind, die zum Ausdruck bringen, wie man von natürlichsprachlichen Formulierungen zu den ihnen in der formalen Sprache entsprechenden Formeln gelangt. Gerade diese Fragestellung scheint im Bereich der formalen Semantik noch wenig erforscht, sei es, daß der genannte Zusammenhang der menschlichen Intuition überlassen wird, sei es, daß nur aufgezeigt wird, wie aus einer formalsprachlichen Formel eine natürlichsprachliche Äußerung erzeugt werden kann, nicht jedoch vice versa.

Betrachtet man das Verhältnis zwischen einem natürlichsprachlichen Satz und der entsprechenden Formel einer formalen Sprache als Übersetzungsbeziehung, dann kann unter einem theoretisch-deskriptiven Aspekt diese Beziehung durch die Aufzählung all derjeniger Satz-Formel-Paare festgelegt werden, die Übersetzungen zueinander sind. Dazu stellen wir einen Regelformalismus vor, mit dem die Elemente einer solchen Übersetzungsrelation generiert werden können, weshalb wir ein nach diesem Formalismus dargestelltes Regelwerk aufgrund der Ähnlichkeiten zu Transformationsgrammatiken als Ü b e r s e t z u n g s g r a m m a t i k (ÜG) bezeichnen.

Unter einem praktisch-operationalen Aspekt entspricht einer Übersetzungsgrammatik ein Verfahren, mit dem aus gegebenen natürlichsprachlichen Formulierungen die entsprechenden formalsprachlichen Formeln hergestellt werden (und vice versa), wobei wir diesen Vorgang in Anlehnung an die Literatur zur Theorie formaler Sprachen und Automaten als T r a n s d u k t i o n bezeichnen.
Die von uns umrissene Übersetzungsgrammatik berücksichtigt diesen zweiten Aspekt der Operationalisierung, so daß die Regeln dieser Grammatik für Transduktionsverfah-

ren verwendet werden können. Im Vordergrund unserer Arbeit steht jedoch der F o r m a l i s m u s einer ÜG, d.h. die Gestalt der Regeln, die zur Verfügung stehen müssen, um den angedeuteten Übersetzungszusammenhang ausdrücken zu können. Wir illustrieren den Formalismus an Beispielen des Deutschen und einer prädikatenlogischen Sprache, die wir als vorgegeben betrachten und nicht weiter diskutieren, so daß unsere Vorschläge nicht daran zu messen sind, inwieweit die Übersetzungsbeispiele adäquat erscheinen, sondern daran, mit welchem Aufwand die Regeln erlauben, den Weg zu den vorgegebenen Übersetzungen zu beschreiben.

Geht man davon aus, daß die Übersetzung von einer natürlichen Sprache in eine andere natürliche Sprache in zwei Schritten beschreibbar ist, d.h. durch eine Übersetzung aus der Ausgangssprache in eine formale Zwischensprache und von der Zwischensprache in die Zielsprache, dann müssten unsere Überlegungen auch auf Paare von natürlichen Sprachen übertragbar sein.

Sicherlich sind unsere Überlegungen an zahlreichen Stellen noch nicht ausgereift oder unvollständig. Wir glauben jedoch, daß diese Überlegungen so weit fortgeschritten sind, daß ihre Dokumentation im vorliegenden Rahmen nicht nur gerechtfertigt, sondern auch geboten erscheint, um eine breitere interdisziplinäre Diskussion zur fundierteren Klärung der involvierten linguistischen, mathematischen und automatentheoretischen Aspekte zu ermöglichen.

0.2 Anmerkung zum Aufbau

Beim Aufbau der vorliegenden Arbeit gingen wir von einem Adressatenkreis aus, der heterogen ist bezüglich seines Informationsstandes und Informationsinteresses. Was deshalb individuell geurteilt als unnötige Wiederholung von Bekanntem erscheinen mag, kann aus einer anderen Sicht notwendige Grundlage für das Verständnis von Einzelproblemen sein. Um eine selektive oder zumindest unterschiedlich vertiefende Lektüre zu ermöglichen, haben wir versucht, die einzelnen Kapitel so zu gestalten, daß sie auch möglichst ohne detaillierte Kenntnis vorangehender Kapitel informativ bleiben. Im Interesse eines im Umgang mit mathematischen Formeln weniger geübten Lesers haben wir auf rigorose formale Darstellungen wenn möglich verzichtet und bedienen uns einer scheinbar umgangssprachlichen Formulierungsweise, die jedoch durch die im Anhang gegebenen Definitionen normiert ist. Für ein präzises Verständnis der Arbeit ist der Anhang deshalb unverzichtbar.

0.3 Zur Entstehung der Arbeit

Die vorliegende Arbeit entstand in der Abteilung Linguistische Datenverarbeitung des Instituts für deutsche Sprache im Rahmen des vom Bundesminister für Forschung und Technologie finanzierten Projekts PLIDIS. Die Arbeit verdankt den Diskussionen mit den Projektmitarbeitern zahlreiche Anregungen. Wertvoll waren vor allem die Verbesserungsvorschläge von D. Kolb, der für das natürlichsprachliche Frage-Antwort-System PLIDIS einen Interpreter für die Regeln der hier vorgeschlagenen Übersetzungsgrammatik programmiert hat. Besonderer Dank kommt W. Dilger zu, der sich in unermüdlicher Geduld als Diskussionspartner und Berater für Probleme der Formalisierung zur Verfügung gestellt hat. Der umsichtigen und einschränkungslosen Förderung von Prof. Dr. D. Krallmann verdankt diese Arbeit ihr Zustandekommen in der vorliegenden Form.

Die Arbeit wurde vom Fachbereich 3, Sprach- und Literaturwissenschaften, der Universität Essen als Dissertation angenommen.

1. VERWENDUNGSZUSAMMENHANG EINER ÜBERSETZUNGSGRAMMATIK

1.1 Übersetzungsrelation

Will man Überlegungen zum Problem der Übersetzung nic auf ein umgangssprachliches Verständnis von "übersetz aufbauen, so wird man bereits hier mit ersten Schwierigkeiten konfrontiert. Definiert man etwa, daß eine sprachliche Formulierung β aus einer Sprache L_2 eine Übersetzung der Formulierung α aus einer Sprache L_1 ist, wenn β die gleiche Bedeutung wie α hat, so ist diese Definition dann unbefriedigend, wenn nicht nähe erklärt ist, was die Bedeutung eines Satzes sein soll Je nachdem, welcher Bedeutungsbegriff zugrundegelegt wird, kann die Definition zudem in einer Vielzahl von Fällen unbrauchbar sein, wenn die Bedeutungsgleichhei total verstanden wird, denn besonders bei der Übersetzung natürlichsprachlicher Formulierungen tritt meistens ein mehr oder minder großer Informationsverlust auf - man denke hier allein an die Schwierigkeit oder Unmöglichkeit der Übersetzung stilistischer Merk male von sprachlichen Formulierungen -. Dieser Informationsverlust muß aber in einer Definition von Übersetzung explizit oder implizit angesprochen sein. Betrachtet man eine sprachliche Formulierung im Rahmen einer Interaktion, könnte man die für die Übersetzung "unschädliche" Differenz zwischen α und β beziehen auf die für die Interaktion relevanten Informationen und festlegen:

> Eine sprachliche Formulierung β aus einer Sprache L_2 ist in einer Interaktion I eine Übersetzung der Formulierung α aus einer Sprache L_1, wenn β die gleichen für I relevanten Informationen enthält wie α.

Die Bezugnahme auf gleiche für eine Interaktion relevante Informationen anstatt auf Bedeutungsgleichheit bringt die Erklärung dessen, was eine Übersetzung sein soll, lediglich insofern weiter, als ein Zusammenhang der Übersetzungsproblematik mit zentralen Fragestellungen der Kommunikationswissenschaft angedeutet ist, der auch charakteristisch ist für die Fundierung neuerer Richtungen der Übersetzungswissenschaft (vgl. etwa KADE (1968) sowie die ausführliche Gesamtdarstellung von WILSS (1977)). Eine Diskussion des Übersetzungsbegriffs auf kommunikationswissenschaftlicher Grundlage erscheint uns im Rahmen unserer Arbeit weder leistbar noch erforderlich, so daß wir es bei einem vorwissenschaftlichen Verständnis belassen wollen und lediglich festhalten, daß wir von Übersetzungen ausgehen, bei denen ein (unschädlicher) Informationsverlust auftreten kann.

Unabhängig davon, wie nun im einzelnen ein Übersetzungsbegriff definiert wird, kann unter einem theoretischen Gesichtspunkt die Übersetzungsbeziehung zwischen Formulierungen einer Sprache L_{AS} und einer Sprache L_{ZS} beschrieben werden durch eine Relation[1)]

$$ÜR \subseteq L_{AS} \times L_{ZS}.$$

d.h. also durch eine Menge von Paaren (α,β), wobei α eine Formulierung aus L_{AS} und β eine Formulierung aus L_{ZS}. Wir betrachten also die beiden Sprachen als Mengen von Formulierungen und nennen ·

---

1) Zur Erläuterung der verwendeten mathematischen Zeichen und Begriffe vgl. die Definitionen und Erläuterungen im Anhang!

ÜR eine *Übersetzungsrelation*, L_{AS} *Ausgangssprache* (bezüglich ÜR) und L_{ZS} *Zielsprache* (bezüglich ÜR). Gilt $(\alpha,\beta) \in$ ÜR, dann bezeichnen wir β auch als eine *Übersetzung* von α.

Geht man davon aus, daß L_{AS} und L_{ZS} insofern zwei völlig verschiedene Sprachen sind, als keine Formulierung von L_{AS} auch gleichzeitig eine Formulierung von L_{ZS} ist, d.h. sind L_{AS} und L_{ZS} disjunkte Mengen, dann sind die formalen Eigenschaften der Relation ÜR festgelegt. Daraus folgt insbesondere, daß ÜR keine Äquivalenzrelation im mathematischen Sinn ist, denn wenn $(\alpha,\beta) \in$ ÜR gilt, kann nicht auch $(\beta,\alpha) \in$ ÜR gelten.

Zur Vermeidung von Mißverständnissen ist zu beachten, daß wir hier eine "mathematisierte" Terminologie einführen. Wenn wir sagen, daß die Relation ÜR keine Äquivalenzrelation ist, dann bedeutet dies nicht, daß bei $(\alpha,\beta) \in$ ÜR α nicht zu β "äquivalent" ist im nicht-mathematischen Sinne von "gleichwertig" oder "gleichbedeutend" oder im Sinne des übersetzungswissenschaftlichen Terminus von "Übersetzungsäquivalenz" (vgl. WILSS (1977), pp 156). Gerade mit Hilfe der Relation ÜR kann "Übersetzungsäquivalenz" für zwei Sprachen L_1, L_2 festgelegt werden als (mathematische) Äquivalenzrelation

$$ÜÄ \subseteq (L_1 \cup L_2) \times (L_1 \cup L_2)$$

mit der Festlegung:

> Ein Paar (α,β) mit $\alpha \in L_1 \cup L_2$ und $\beta \in L_1 \cup L_2$ ist in ÜÄ, wenn $(\alpha,\beta) \in$ ÜR oder $(\beta,\alpha) \in$ ÜR oder $\alpha = \beta$.

Bindet man den Übersetzungsbegriff an den Handlungszusammenhang einer sprachlichen Interaktion, dann folgt daraus, daß zur Beschreibung aller denkbaren Übersetzungsbeziehungen für ein Sprachenpaar (L_{AS}, L_{ZS}) eine Menge von Übersetzungsrelationen $ÜR_1, \ldots, ÜR_n$ anzugeben ist, wobei $ÜR_i$ die Übersetzungsbeziehung für den Handlungszusammenhang i beschreibt. In der Folge vernachlässigen wir jedoch diese Differenzierung und reden lediglich von (irgend)einer Übersetzungsrelation ÜR.

Bei unseren Überlegungen klammern wir eine primäre, als methodisch und empirisch charakterisierbare Fragestellung aus, nämlich:

> Aufgrund welcher Faktoren läßt sich entscheiden, ob ein Paar (α, β) in Übersetzungsrelation steht?

Vielmehr konzentrieren wir uns auf eine damit verbundene sekundäre, mehr praktische oder technische Frage:

> Mit welchen Darstellungsmitteln läßt sich der Sachverhalt darstellen, daß ein Paar (α, β) in einer Übersetzungsrelation ÜR steht und wie kann ausgehend von einer dargestellten Übersetzungsrelation ermittelt werden, ob ein Paar (α, β) in ÜR ist?

Dies bedeutet, daß wir in der Folge davon ausgehen, daß eine Übersetzungsrelation *gesetzt* sei. M.a.W. wir interessieren uns nicht dafür, aufgrund welcher (empirischer) Kriterien und welcher methodischer Hilfsmittel festgelegt wird, daß eine Formulierung α zu einer Formulierung β in Übersetzungsrelation stehen soll. Uns interessiert vielmehr, mit welchen formalen Mitteln das Resultat einer solchen Festlegung ausgedrückt werden kann.

1.2 Aufzählung einer Übersetzungsrelation

Das einfachste Mittel zur Festlegung einer Übersetzungsrelation ÜR für ein Sprachenpaar (L_{AS}, L_{ZS}) ist die extensionale Aufzählung, d.h. die Angabe einer Liste all derjenigen Paare aus $L_{AS} \times L_{ZS}$, die in ÜR liegen. Dieses Verfahren ist jedoch für umfangreichere Sprachmengen wegen seines Aufwandes nicht praktikabel. Wir suchen deshalb nach einem Verfahren, das auf Regeln aufbaut, mit denen für Teile einer ausgangssprachlichen Formulierung die zielsprachlichen Entsprechungen festgelegt werden können, sowie nach Regeln, die angeben, wie aus solchen Teilformulierungen in Abhängigkeit von einer ausgangssprachlichen Gesamtformulierung eine zielsprachliche Gesamtformulierung zu konstruieren ist. Dies bedeutet, daß wir von der grundlegenden Prämisse ausgehen, daß sich das Phänomen "übersetzen" in einzelne Schritte zerlegen läßt, die regelhaft beschreibbar sind.
Wir verdeutlichen diese Zielsetzung durch eine Analogie zur Beschreibung von Einzelsprachen mit Hilfe von Regelgrammatiken (auch als generative Grammatiken, Phrasenstrukturgrammatiken, CHOMSKY-Grammatiken etc. bezeichnet) und führen dabei einige Begriffe ein, die im Anhang in einem größeren Kontext präzisiert sind.

Anstatt eine Sprache L durch Angabe einer Liste aller Sätze von L festzulegen, wird eine Regelgrammatik G = (VN, VT, S, P) angegeben, wobei

VN eine Menge von Kategorien-Symbolen, das n i c h t - t e r m i n a l e V o k a b u l a r (von G), auch als 'syntaktische Kategoriensymbole' bezeichnet;

VT eine von VN disjunkte Menge von Symbolen, das terminale Vokabular (von G), in linguistischer Umgangssprache auch als 'Wörter' oder 'Lemmata' bezeichnet;

S ein ausgezeichnetes Symbol aus VN, das Startsymbol (von G);

P eine Menge von Paaren der Form (a,α), geschrieben $a \rightarrow \alpha$, die (Produktions-) Regeln (von G), wobei a aus VN und α eine aus Symbolen von $VN \cup VT$ gebildete Kette[1)] ist.

Das Vokabular $V = VN \cup VT$ nennen wir dann auch das Alphabet von G.
Um mittels G eine Sprache zu definieren, wird folgender Ableitungsbegriff festgelegt:

- sind u und v Ketten über V, dann ist eine Kette $\beta = u\alpha v$ aus einer Kette uav direkt ableitbar, wenn es in P eine Regel $a \longrightarrow \alpha$ gibt;

- eine Folge von Ketten $(\alpha_1, \ldots, \alpha_n)$ ist eine Ableitung von α_n aus α_1 (in G), wenn für alle i, $i \leq (n-1)$, gilt: α_{i+1} ist direkt ableitbar aus α_i. Wir sagen dann auch, α_n ist aus α_i ableitbar. Besteht α_n nur aus terminalen Symbolen, d.h. $\alpha_n \in VT^*$, ist α_n terminal abgeleitet (in G).

- die Menge L(G) aller aus dem Startsymbol S (in G) terminal abgeleiteten Ketten ist die

---

1) Verkettet man Symbole eines Vokabulars V (z.B.: 'das ', 'Mädchen ', 'lacht '), dann bezeichnen wir das daraus entstehende Gebilde (z.B.: 'das Mädchen lacht ') als Kette über V. Die Menge aller Ketten, die sich mit Symbolen aus V bilden lassen, notieren wir mit V^*. Entsprechendes gilt für VN und VT.

von G *e r z e u g t e S p r a c h e*. Die Ketten von L(G) bezeichnen wir dann als 'Sätze' oder 'Formeln' von L(G).

Zur realen Aufzählung oder Erzeugung einer (endlichen) Sprache L(G) kann ein Verfahren angegeben werden, das auf der Interpretation einer Produktionsregel $a \rightarrow \alpha$ als einer Anweisung beruht, nach der in beliebigen Ketten ein Symbol a durch α zu ersetzen ist, wodurch eine neue Kette 'produziert' wird. Bei geeigneter Steuerung des Verfahrens können dann alle aus S in G terminal abgeleiteten Ketten, i.e. die Sätze von L(G), erzeugt werden.

Soll nun mittels einer Regelgrammatik G eine vorgegebene Sprache L beschrieben werden, ist es eine Frage der Adäquatheit bzw. Vollständigkeit oder Korrektheit von G, ob L = L(G) bzw. ob L(G) nur eine Teilmenge von L oder ob L(G) Ketten enthält, die nicht in L liegen. Die Beurteilung der Adäquatheit von G setzt voraus, daß L irgendwie unabhängig von G gesetzt ist und daß es Kriterien gibt, die wir in aller Vagheit als 'Kompetenz für L' bezeichnen wollen, mit deren Hilfe entscheidbar ist, ob eine Kette zu L gehört oder nicht.

Der Formalismus einer Regelgrammatik ist dann lediglich ein Darstellungshilfsmittel, mit dem idealiter die Leistungen der Kompetenzkriterien nachgebildet werden. Der Formalismus selbst stellt jedoch keine Entdeckungsprozeduren für die Auffindung solcher Kriterien zur Verfügung.
Wenn wir nun nach einem Regelwerk suchen, mit dem die Kompetenz für eine gegebene Übersetzungsrelation expliziert werden kann, dann bedeutet dies, daß wir nach einem ähnlichen axiomatischen System wie dem

einer Regelgrammatik suchen. Durch Verallgemeinerung des für Regelgrammatiken verwendeten Ableitungsbegriffs soll eine Festlegung darüber ermöglicht werden, welche Paare (α,β) aus $L_{AS} \times L_{ZS}$ in einer Übersetzungsrelation ÜR liegen. Ein solches axiomatisches System nennen wir dann auch Ü b e r s e t z u n g s - g r a m m a t i k .

Obwohl wir unseren Kompetenzbegriff vage belassen und nicht in die Dichotomie Kompetenz-Performanz (CHOMSKY (1969), pp 13) gestellt haben, ist uns klar, daß eine Übersetzungsgrammatik auch in diesem Kontext zu diskutieren wäre, sowie ihr Beitrag zu der von WILSS angeführten Zielsetzung,

> "daß zur modellorientierten Beschreibung der sprachpaarübergreifenden übersetzerischen Kompetenz (als interlingualer Sprachverstehens- und Sprachreproduktionskompetenz) ergänzend die problemorientierte, beobachtungsadäquate Beschreibung der sprachpaarbezogenen übersetzerischen Performanz (als interlingualer Textanalyse- und Textsyntheseperformanz) zu treten hat" (WILSS (1977),p.12).

1.3 Erzeugung einer Übersetzung

Wir haben bisher noch nicht erläutert, in welcher Weise die von einer Übersetzungsgrammatik ÜG zu einem Sprachenpaar L_1, L_2 zu erzeugenden Formeln, die Sprache von ÜG, mit einer Übersetzungsrelation $ÜR \subset L_1 \times L_2$ zusammenhängen soll. Dazu gibt es mindestens zwei Möglichkeiten:

(1) Die Formeln der Sprache von ÜG haben die Form $\alpha\beta$, so daß $(\alpha,\beta) \in ÜR$. Es gibt dann nur ein einziges Axiom, aus dem parallel sowie in gegenseitiger Abhängigkeit die Teilketten α und β abgeleitet

werden. ÜG kann dann als Parallel-grammatik beschrieben werden (vgl. PRATT (1971)).

(2) Die Sprache von ÜG ist die Menge aller Formeln β, für die es ein α gibt, so daß $(\alpha,\beta) \in$ ÜR und β ist in ÜG aus α ableitbar. D.h. die Axiome von ÜG, aus denen Formeln abgeleitet werden, wären dann gleich der Menge $\{\alpha \mid$ es gibt ein x, so daß $(\alpha,x) \in$ ÜR $\}$.

Wie wir zeigen werden, ist die Möglichkeit (2) modifiziert realisierbar durch ein formales System, in dem zwar nicht β aus α ableitbar ist, jedoch eine Baumstruktur t_β aus einer Baumstruktur t_α, so daß β die Endkette von t_β und α die Endkette von t_α ist, wobei die Endkette eines Baumes durch Verkettung derjenigen Symbole entsteht, mit denen die Endknoten des Baumes markiert sind. ÜG kann dann mit einem ähnlichen Formalismus wie demjenigen einer Transformationsgrammatik (vgl. (GINSBURG/PARTEE (1969), PAUSE (1976)) beschrieben werden.

Um zwischen den beiden Möglichkeiten entscheiden zu können, müßten folgende Fragen bezüglich der resultierenden Systeme beantwortet werden:

- Welche formale Eigenschaften können für die beiden Systeme bewiesen werden und ergibt sich daraus eine Bewertung?
- Welches der beiden Systeme ist ökonomischer bezüglich der Regelmenge und der Komplexität der Einzelregeln?

- Welche Verfahren lassen sich, wenn überhaupt, für die beiden Systeme angeben zur
  (i) Entscheidung, ob für ein Paar (α,β) gilt: $(\alpha,\beta) \in$ ÜR
  (ii) Erzeugung einer Formel β aus einer Formel α, so daß gilt: $(\alpha,\beta) \in$ ÜR
  (iii) Erzeugung einer Formel α aus einer Formel β, so daß gilt: $(\alpha,\beta) \in$ ÜR

- Nach welchen Kriterien sind diese Verfahren zu bewerten?

Die Untersuchung dieser Fragen ist uns weder im Rahmen dieser Arbeit noch - soweit überschaubar - aufgrund des derzeitigen Forschungsstandes möglich. Wir treffen deshalb eine heuristische Entscheidung auf der Grundlage des Verwendungszusammenhangs, in dem wir eine Übersetzungsgrammatik anwenden wollen.

Die Verwendung einer Übersetzungsgrammatik als Grundlage von Entscheidungsverfahren für (i) oder für Verfahren zur Aufzählung einer Übersetzungsrelation, scheint uns vor allem von systematischem, deskriptivem Interesse zu sein. Demgegenüber wird die praktische Anwendung einer Übersetzungsgrammatik vornehmlich im Bereich des Übersetzens als einem Prozeß liegen, für den Verfahren zu (ii) und (iii) gesucht werden. Sind diese Verfahren algorithmisierbar, d.h. in einer Folge von Einzelanweisungen so präzisierbar, daß diese Anweisungen von einer Maschine ausführbar sind, dann müßte auf der Basis einer Übersetzungsgrammatik eine (abstrakte) Maschine T konstruierbar sein. Diese Maschine T würde als Eingabe Formulierungen einer Ausgangssprache akzeptieren und zu jeder Eingabe eine Ausgabeformulierung erzeugen, die zur

Eingabeformulierung in Übersetzungsrelation steht (vgl. Abb. 1.1).

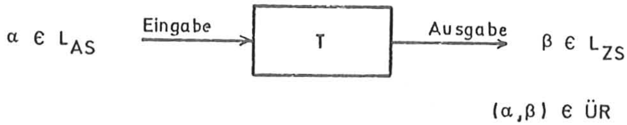


Abbildung 1.1: Schema einer Übersetzungsmaschine

Die Maschine T nennen wir auch *Transduktor* und die Erzeugung einer Ausgabe durch M *Transduktion*.

Als Realisierung von T ist ein Computerprogramm vorstellbar, das die Regeln einer Übersetzungsgrammatik interpretiert als Anweisungen für die Durchführung von Ableitungsschritten entsprechend dem für eine Übersetzungsgrammatik festgelegten Ableitungsbegriff. Den Idealfall, daß die Regeln einer Übersetzungsgrammatik umkehrbar sind und somit ein zu T analoger Transduktor T' angebbar ist, der Transduktionen in umgekehrter Richtung ausführt, betrachten wir in der Folge nicht näher (vgl. hierzu jedoch die Überlegungen von PAUSE (1976) zur Umkehrbarkeit von Transformationsgrammatiken).

Der skizzierten Anwendung einer Übersetzungsgrammatik zur Konstruktion eines Transduktors kommt der oben unter (2) angedeutete formale Rahmen für eine Übersetzungsgrammatik am nächsten, so daß wir die Möglichkeiten einer Parallelgrammatik nicht weiter verfolgen werden.

1.4 Anwendung einer Übersetzungsgrammatik

Zum Verständnis und auch zur Rechtfertigung des Nachfolgenden ist es erforderlich, die Praxis, für die wir eine Übersetzungsgrammatik konzipieren, weiter zu präzisieren. Dies ist auch notwendig, um zu vermeiden, daß unsere Vorschläge an Ansprüchen gemessen werden, die wir nicht erheben, insbesondere den Anspruch, eine Übersetzungsgrammatik für ein Paar von natürlichen Sprachen vorschlagen zu wollen. Es wird jedoch unter Bezugnahme auf den Begriff einer Semantik-Sprache aufgezeigt werden, wie eine Übersetzungsgrammatik im Hinblick auf natürliche Sprachen verwendet werden kann.

1.4.1 Semantische Interpretation als Übersetzung

Bei seinen Überlegungen über den Zusammenhang zwischen Semantik und Übersetzungswissenschaft stellt NEUBERT "eine frappierende Gemeinsamkeit zwischen dem Gegenstand der Semantik und dem zentralen Problem der Übersetzungswissenschaft" fest, und er fährt fort:

> "Man kann doch offenbar behaupten, daß das zielsprachliche (= ZS) Translat eines Satzes der Ausgangssprache (= AS) in einem durchaus nicht tautologischen Sinn als eine Angabe der Bedeutung eines Textsegments der AS mit Nicht-AS-Mitteln angesehen werden kann. Auf eine knappe Formel gebracht: Das ZS-Translat stellt eine semantische Interpretation der AS-Version dar." (NEUBERT (1968), p. 200)

NEUBERT verfolgt den Gedanken jedoch nicht weiter, die Bedeutung einer Formulierung einer Sprache L_1 durch Formulierungen einer Sprache L_2 darzustellen. Dies ist allerdings auch nicht unbedingt sinnvoll, wenn die

Bedeutungen von Formulierungen der Sprache L_2 ihrerseits nicht expliziert sind.

In neuerer Zeit mangelt es jedoch nicht an Versuchen, ausgehend von Arbeiten wie etwa MONTAGUE (1970), LEWIS (1970), THOMASON (1972) und CRESSWELL (1973) formale Sprachen zu formulieren, für die einerseits eine semantische Interpretation angebbar ist und die andererseits so ausdrucksmächtig sind, daß die Inhalte natürlichsprachlicher Formulierungen in ihnen wiedergebbar sind. Für welchen Umfang von natürlichsprachlichen Formulierungen dies möglich ist oder möglich sein wird, kann nur spekulativ beantwortet werden und soll hier keine weitere Berücksichtigung finden, zumal wir uns in der Folge ohnedies nur mit Ausschnitten natürlicher Sprachen beschäftigen.

Eine solche Sprache, mit der die Bedeutungen von Formulierungen einer anderen Sprache dargestellt werden, nennt ZIFONUN eine "Semantiksprache" und definiert:

> "Eine Sprache L_2 ist Semantiksprache für eine Sprache L_1, wenn
>
> (1) mindestens ein Teil aller in L_1 ausdrückbaren Inhalte in L_2 ausdrückbar sind;
>
> (2) für die beiden Sprachen eine Übersetzungsrelation definiert ist, so daß mindestens ein Teil aller Ausdrücke von L_1 eine Übersetzung in L_2 haben.
> Diese Übersetzungsrelation kann bezogen sein auf einen Handlungszusammenhang C. Dann ist in C ein Ausdruck $a_2 \in L_2$ eine Übersetzung eines Ausdrucks $a_1 \in L_1$ gdw. die Inhalte, die nach der Intention derjenigen, die mit a_1 in C interagieren, mit a_1 in C ausgedrückt werden sollen, auch mit a_2 ausgedrückt werden.
>
> (3) für die Sprache L_2 eine semantische Interpretation angebbar ist, die aufgrund der Übersetzungsrelation auf L_1 induzierbar ist .

'In C in Übersetzungsrelation stehen' heißt daher nicht interlingual synonym sein. Der Begriff der Synonymie würde voraussetzen, daß alle Inhalte, die beliebige Sprecher von L_1 mit a_1 vermitteln, auch mit a_2 vermittelt werden können. Die beiden Ausdrücke müßten in allen Gebräuchen, die von ihnen regelhaft gemacht werden können, übereinstimmen. Dies ist jedoch im Verhältnis von L_1 und L_2 zumindest immer dann kaum der Fall, wenn L_2 eine formale Kunstsprache ist. Denn L_2 ist in der Regel ärmer als L_1, da sie ja zu bestimmten Zwecken und für bestimmte Handlungszusammenhänge entworfen ist, etwa zum formalen Schließen o.ä." (ZIFONUN (1978), Kap. 1.2.1)

Neben dem primären Problem, eine formale Sprache etwa auf der Basis der Prädikatenlogik so ausdrucksmächtig zu definieren, daß sie als Semantiksprache für Ausschnitte einer natürlichen Sprache fungieren kann, stellt sich demnach die Frage nach der Übersetzung von natürlichsprachlichen Formulierungen in die Semantiksprache, die SUPPES so charakterisiert:

"The central difficulty with this approach is, that now as before how the semantics of the surface grammar is to be formulated is still unclear. In other words, how can explicit formal relations be established between first-order logic and the structure of natural languages? Without the outlines of a formal theory, this line of approach has moved no further than the classical stance of introductory teaching in logic, which for many years has concentrated on the translation of English sentences into first-order logical notation. The method of translation, of course, is left at an intuitive and ill-defined level." (SUPPES (1973), p. 370)

Auf diesem Hintergrund fassen wir unsere Überlegungen zu einer Übersetzungsgrammatik auf als einen Beitrag zur Klärung, mit welchen Methoden die Übersetzungsrelation zwischen einer natürlichen Sprache und einer formalen Sprache erklärt werden kann. D.h. unsere Vorschläge werden daran zu messen sein, inwieweit es möglich ist, mit den erarbeiteten Formalismen eine solche Übersetzungsrelation darzustellen.

1.4.2 Semantiksprache und interne Repräsentation in einem Informationssystem

Das theoretisch linguistische Interesse an einer Semantiksprache erhält eine praktische Relevanz durch die Notwendigkeit, bei der Konstruktion von Informationssystemen für natürlichsprachliche Ein- und Ausgabe über formale Repräsentationsmittel zu verfügen, die geeignet sind zur Darstellung der natürlichsprachlich formulierten Informationen. Unabhängig davon, ob solche Systeme im Erkenntnisinteresse künstlicher Intelligenzforschung (vgl. WULZ (1976)) oder mit Schwerpunkt auf einem praktischen Einsatz (vgl. z.B. das System PLIDIS, BERRY-ROGGHE et al. (1978)) realisiert werden zur maschinellen Informationserschließung, zur automatischen Problemlösung oder zur Computersimulation von Sprachverstehen (vgl. v.HAHN (1978), WULZ/ZIFONUN (1978)), erweist sich - thesenhaft zusammengefaßt - die Verwendung einer Semantiksprache zur systeminternen Repräsentation mindestens unter drei Gesichtspunkten als sinnvoll:

(i) Das DV-System muß gleiche Inhalte erkennen, auch wenn sie natürlichsprachlich unterschiedlich ausformuliert sind. Aus ökonomischen Gründen ist es deshalb angezeigt, gleiche Informationen gleich darzustellen mit Hilfe einer formalen Sprache und so natürlichsprachliche Paraphrasen aufzulösen.

(ii) Ein "intelligentes" System soll in der Lage sein, mit Hilfe seines "Wissens" Informationen zu erschließen, die nicht explizit abgespeichert sind. Bezeichnet man die im System gespeicherten Fakten (= Aussagen) und Regeln bzw. Gesetzmäßigkeiten (= Axiome) als "Wissen", beruht die "Intelligenz" des Systems u.a. auf der Fähigkeit, logische Schlüsse zu ziehen, d.h. Ableitungen durchzuführen. Eine formale Repräsentationssprache ermöglicht die Definition formaler Ableitungsregeln, die auf der syntaktischen Form von Einheiten der Repräsentationssprache aufbauen nach dem Vorbild logischer Ableitungsregeln wie etwa 'modus ponens' u.a. (vgl. (ZIFONUN (1978)).

(iii) Die häufige Ambiguität natürlichsprachlicher Formulierungen erfordert bei der maschinellen Verarbeitung eine Disambiguierung oder zumindest ein Erkennen der Ambiguität. Werden natürlichsprachliche Formulierungen übersetzt in eine Semantiksprache, erübrigt sich eine forcierte Disambiguierung auf der syntaktischen Ebene der natürlichen Sprache. Trotzdem bestehende Ambiguitäten können dann zumindest teilweise mit den für (ii) erforderlichen Verfahren aufgelöst werden.

Eine (abstrakte) Maschine M, die in der Lage sein soll, solche Verfahren auszuführen, kann in einer ersten Annäherung skizziert werden als bestehend aus drei Teilmaschinen T_1, IS, T_2 (vgl. Abb. 1.2). Während IS die eigentlichen Verfahren der Informationsverarbeitung oder Problemlösung ausführt, sind T_1 und T_2 Transduktoren, wobei T_1 natürlichsprachliche Formulierungen aus L_1 übersetzt in Formulierungen der formalen Sprache L_2 und T_2 die formalsprachlichen Ergebnisse der Informationsverarbeitung wieder zurückübersetzt in L_1.

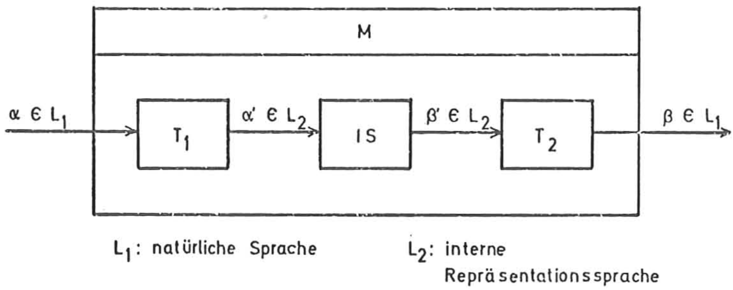


Abbildung 1.2

Damit ist die Praxis angedeutet, für die wir eine Übersetzungsgrammatik nutzbar machen wollen. D.h. mit der von uns konzipierten Übersetzungsgrammatik soll nicht nur die Übersetzungsrelation zwischen Ausschnitten einer natürlichen Sprache und einer formalen Semantiksprache darstellbar sein, sondern auf der Basis der Übersetzungsgrammatik soll auch ein Transduktor für ein Informationssystem zur Übersetzung natürlichsprachlicher Formulierungen in Formeln einer Semantiksprache konstruierbar sein.

Konkreter Anlaß für unsere Überlegungen war die von ZIFONUN entwickelte "Konstruktsprache" (KS), die auf prädikatenlogischer Grundlage sowohl als Semantiksprache als auch als Mittel zur internen Repräsentation in einem Informationssystem konzipiert wurde (vgl. ZIFONUN (1978)). Zur Verwendung dieser Konstruktsprache im System PLIDIS (Problemlösendes Informationssystem mit Deutsch als Interaktionssprache) war ein Transduktor erforderlich, der auf der Grundlage der hier vorgeschlagenen Übersetzungsgrammatik programmiert wurde (zum Gesamtsystem PLIDIS vgl.

BERRY-ROGGHE/WULZ (1977) bzw, neuerdings auch BERRY-ROGGHE et al. (1978)). Die Grobstruktur des in PLIDIS verwendeten Transduktors ist in Abb. 1.3 schematisch dargestellt. Die in PLIDIS vorgenommene Aufteilung der Übersetzung in drei Schritten, morphologische Analyse - syntaktische Analyse - Übersetzung im engeren Sinn, wird verständlich, wenn wir unsere Konzeption einer Übersetzungsgrammatik näher erläutert haben.

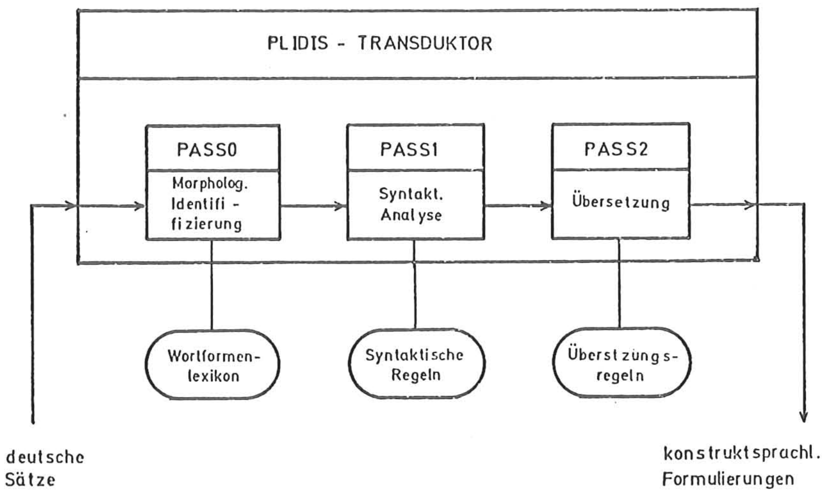


Abbildung 1.3: Schema der Übersetzungskomponente von PLIDIS

1.4.3 Interlingua und Mehrschrittübersetzung bei natürlichen Sprachen

Unsere Überlegungen sind nicht notwendigerweise auf die angedeutete Praxis der automatischen Problemlösung und Informationserschließung beschränkt. Wenn es möglich ist, eine Übersetzungsgrammatik $ÜG_1$ für ein Sprachenpaar L_1, L_2 zu definieren, wobei L_1 eine natürliche Sprache und L_2 eine Semantiksprache, dann müßte es auch möglich sein, eine Übersetzungsgrammatik $ÜG_2$ für ein Paar L_2, L_3 zu definieren, so daß L_3 eine natürliche Sprache ist. Konstruiert man den Transduktor T_1 von Abb. 1.2 aufgrund der Regeln von $ÜG_1$ und T_2 aufgrund der Regeln von $ÜG_2$, erzeugt die oben skizzierte Maschine M aus Formulierungen der Sprache L_1 Formulierungen der Sprache L_3. In einem solchen Modell wird die Semantiksprache L_2 als nicht-einzelsprachliche Interlingua (vgl. PAUSE (1974)) verwendet, über der die Komponente IS insbesondere zur Disambiguierung operiert.

Eine Diskussion der Frage, inwieweit ein solches Modell realisierbar sein dürfte, geht über den Rahmen dieser Arbeit hinaus. Auf dem Hintergrund der Künstlichen-Intelligenz-Forschung erscheint es uns jedoch plausibel, daß für eingeschränkte Anwendungsbereiche und Sprachmengen Lösungen gefunden werden können, die in solchen Bereichen zu sinnvollen und korrekten Übersetzungen führen, in denen stilistische Ähnlichkeiten zwischen Ausgangssprache und Zielsprache keine Rolle spielen.

2. INFORMALE SKIZZE EINER ÜBERSETZUNGSGRAMMATIK AM BEISPIEL DEUTSCH/KONSTRUKTSPRACHE

Zum besseren Verständnis der formalen Definitionen in Kapitel 3 und der Regelbeispiele in Kapitel 4 entwickeln wir zuerst informal das Konzept einer Übersetzungsgrammatik. Anhand von Beispielen wollen wir vor allem aufzeigen, über welchen sprachlichen Einheiten einer Ausgangs- und einer Zielsprache Regularitäten für eine Übersetzung festgelegt werden können und welche Operationen die Regeln beschreiben sollen, mit denen diese Regularitäten dargestellt werden. Wir verwenden dazu Beispiele aus dem System PLIDIS und der dort eingesetzten Konstruktsprache KS. Da das hier so weit als möglich informal Dargestellte später formal definiert wird, sind Wiederholungen unvermeidlich, die wir jedoch im Interesse einer besseren Durchschaubarkeit unserer Vorschläge in Kauf nehmen.

2.1 Zur Zielsprache

Die Konstruktsprache KS, die wir bei unseren Übersetzungsbeispielen als Zielsprache zu ausgangssprachlichen Formulierungen des Deutschen verwenden, ist eine prädikatenlogische Sprache mit einer Sortenstruktur, die wir hier nur mit starken Vereinfachungen skizzieren (zu einer Gesamtdarstellung vgl. ZIFONUN (1976), (1978); DILGER/ZIFONUN (1978); zum hier verwendeten Sprachumfang von KS vgl. Abschnitt 4.1).

Den S o r t e n von KS entspricht auf semantischer Ebene eine Klassifizierung von Objekten aufgrund ihrer Eigenschaften. Beispiele für Sorten in einem

KS-Sprachausschnitt, mit dem Sachverhalte über einen Weltausschnitt "Industrieabwasserüberwachung" formulierbar sein sollen, sind: 'ort', 'stoff' (wie z.B. Cadmium, Zink), 'stoffkoll' (= Stoffkollektiv als Ansammlung von Stoffen wie z.B. eine Abwasserprobe), 'per' (= Person), 'firma' bzw. 'betrieb' (als Untersorten von 'per'), 'num' (numerischer Wert), 'dimzahl' (= dimensionierter numerischer Wert wie 5 mg/l, 3 Gramm), 'int' (= Intervall, Zeitbezeichnung), 'abstrobj' (= abstraktes Objekt wie z.B. ein Schriftstück, ein Laborbericht, ein Erlaß).

Das A l p h a b e t von KS umfaßt:

- KS-Variablen, i.e. Variablen mit einer zugeordneten Sorte wie z.B. x.per, x.stoff etc.
- Individuenkonstanten, i.e. Namen wie 'Stuttgart', 'Günter Lauxmann', 'Zink' etc.
- logische Junktoren: NICHT, UND, ODER, IMPLIK, AEQUIV
- logische Quantoren: FUERALL, EXIST
- Operationszeichen
- Relationszeichen.

Im nachfolgenden informalen Teil halten wir uns nicht an die Unterscheidung von Operationszeichen und Relationszeichen und ihre unterschiedliche syntaktische Verwendung, sondern reden nur von 'Prädikatszeichen' oder kurz von 'Prädikaten'.

Während den KS-Variablen und Individuenkonstanten eine Sorte zugeordnet wird, wird den Prädikaten ein Sortentupel zugeordnet. D.h. sind s_i, $i \leq n$, Sorten und ist P ein n-stelliges Prädikat, wird P das Tupel $(s_1,\ldots,s_n)$ zugeordnet, wodurch festgelegt ist, daß das i-te Argument von P die Sorte s_i haben muß. Damit skizzieren wir eine vereinfachte S y n t a x von KS rekursiv:

1. KS-Variablen und Individuenkonstanten sind Terme von KS

2. Ist P ein n-stelliges Prädikat mit dem Sortentupel $(s_1,\ldots,s_n)$ und sind $t_1,\ldots,t_{n-1}$ Terme, wobei t_i ein Term der Sorte s'_i, $i \leq (n-1)$, dann ist $(P\ t_1\ldots t_{n-1})$ ein Term der Sorte s_n, wenn für alle i, $i \leq (n-1)$, gilt: $s_i \geq s'_i$.

3. Ist P ein n-stelliges Prädikat mit dem Sortentupel $(s_1,\ldots,s_n)$ und sind $t_1,\ldots,t_n$ Terme, wobei t_i ein Term der Sorte s'_i, $i \leq n$, dann ist $(P\ t_1\ldots t_n)$ eine atomare Formel, wenn für alle i, $i \leq n$, gilt: $s_i \geq s'_i$.

4. Atomare Formeln sind Formeln.

5. Sind p und q Formeln, dann auch (NICHT p), (UND p q), (ODER p q), (IMPLIK p q), (AEQUIV p q).

6. Ist p eine Formel und x eine KS-Variable, dann sind auch (EXIST x p) und (FUERALL x p) Formeln.

7. Sonst ist nichts ein Term oder eine Formel von KS.

Bei unseren Beispielen gehen wir davon aus, daß KS mit einer Regelgrammatik beschrieben sei, zu deren nichtterminalem Vokabular u.a. die Symbole FORMEL, AFOR (atomare Formel), TERM und PRAED (Prädikat) gehören. Dazu seien die Merkmale 'sorttup' (Sortentupel) und 'sorte' definiert, so daß komplexe Symbole wie PRAED/sorttup=($s_1 \ldots s_n$) und TERM/sorte=s bildbar seien. Aufgrund der Produktionsregeln der KS-Grammatik seien dann Syntaxbäume bildbar wie etwa die in Abb. 2.1 dargestellte Struktur.

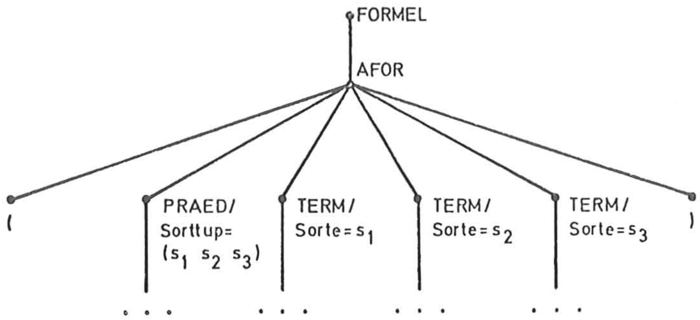


Abbildung 2.1: Ausschnitt aus einem KS-Syntaxbaum

Mit einem für den Gegenstandsbereich "Industrieabwasserüberwachung" definierten KS-Vokabular kann aufgrund der umrissenen Syntax und der Bedeutung dieses Vokabulars ein Satz wie

(1) Die Wasserproben bei Fa. Joos in Stuttgart vom 21.11.77 enthielten 0,1 mg/l Zink.

reformuliert werden durch die folgende F o r m e l :

```
(2) (EXIST x.abstrobj                               (i)
     (ANTEIL ZN                                     (ii)
            (PROBE (BETRIEB JOOS STUTTGART)
             77/11/21)                              (iii)
            x.abstrobj                              (iv)
            0,1-mg/l))                              (v)
```

Die Formel wird gebildet durch das 4-stellige Prädikatsymbol ANTEIL mit den im Sortentupel (stoff, stoffkoll, abstrobj, dimzahl) festgelegten Restriktionen für die Argumente. Zu den Argumenten:

1. Argument (ii): 1-stelliger Term der Sorte stoff (in Anlehnung an das chem. Zeichen für 'Zink')

2. Argument (iii): 2-stelliger Term der Sorte stoffkoll(ektiv) mit dem Prädikat PROBE, dessen 1. Argument ein Term der Sorte betrieb und dessen 2. Argument ein Term der Sorte int(ervall)

3. Argument (iv): sortierte Variable der Sorte abstrobj (abstraktes Objekt wie z.B. ein Laborbericht, in dem die Untersuchungsergebnisse für eine Probe festgehalten sind), die durch einen Existenzquantor gebunden ist (i).

4. Argument (v): 1-stelliger Term der Sorte dimzahl (dimensionierte Zahl).

Betrachtet man (2) als eine Übersetzung des Satzes (1) in die formale Sprache KS, dann soll es möglich sein, mittels der Regeln einer entsprechenden Übersetzungsgrammatik die Formel (2) aus dem Satz (1) herzuleiten.

2.2 Ersetzung durch Kontextpattern

Eine auf der Hand liegende Regularität des Zusammenhangs von zwei Sprachen sind die Entsprechungen ihrer terminalen Symbole, wie sie zum Beispiel in zweisprachigen Wörterbüchern aufgelistet sind. Meistens werden jedoch in solchen Wörterbüchern nicht nur Lexempaare angegeben, deren erste Komponente in der einen Sprache der zweiten Komponente in der anderen Sprache entspricht. Vielmehr wird auch notiert, wie zumindest das zweite Lexem in der andern Sprache zu verwenden ist, etwa durch Anführung von Beispielen oder durch Hinweise auf den syntaktischen Kontext.
In ähnlicher Weise verfahren wir bei der Beschreibung des Zusammenhangs zwischen deutschen Lexemen und Symbolen der Konstruktsprache KS. D.h. in einer Übersetzungsregel für das Verb "enthalten" wollen wir nicht nur festlegen, daß diesem Verb in KS das Prädikat ANTEIL entspricht, sondern auch, daß ANTEIL als vierstelliges Prädikat mit dem Sortentupel (stoff, stoffkoll, abstrobj, dimzahl) zur Bildung einer Formel verwendet wird. Wir stellen diese Information durch den Klammerausdruck

(3) (FORMEL [ANTEIL A1 A2 A3 A4])

dar, wobei die Symbole A1 bis A4 Hilfssymbole sind, die nicht zum Vokabular von KS gehören. Um objektsprachliche Klammern, d.h. Klammern die zum Vokabular von KS gehören, von metasprachlichen Klammern, die zur Darstellung von Listen gehören, zu unterscheiden, schreiben wir für die Klammern von KS auch [bzw.]. Paraphrasiert man den mit [ ] geklammerten Ausdruck von (3) mit der Formulierung

"Der ANTEIL an <u>stoff</u> A1 in einem <u>stoffkoll</u>ektiv A2 hat gemäß einem abstrakten objekt A3 einen dimensionierten zahlenwert von A4."

dann kann man dieser Paraphrase deutsche Sätze gegenüberstellen wie etwa (1) oben oder auch

(4) Die Abwasserproben in der Firma Müller enthielten gemäß Laborbericht vom 23.11.77 0,5 mg/l Cadmium.

(5) Am 21.11.77 enthielten die Proben bei der DMV 0,8 mg/l Zink.

(6) Enthielten die Proben von Lauxmann Cadmium?

Bei der Gegenüberstellung sieht man, daß die deutsche Entsprechung zu

- <u>stoff</u> A1 in Sätzen mit dem Verb "enthalten" als Nomen einer Nominalphrase formuliert ist, die Akkusativergänzung zum Verb ist,
- <u>stoffkoll</u> A2 als Verbergänzung im Nominativ,
- <u>abstrobj</u> A3 als Praepositionalphrase, wobei Praepositionen möglich sind wie "gemäß", "laut", "entsprechend", "aufgrund" etc.,
- <u>dimzahl</u> A4 als Maßangabe der Ergänzung im Akkusativ.

Dies bedeutet für die Übersetzung von deutschen Sätzen mit "enthalten", daß zumindest ein Teil der KS-Entsprechung aus einer mit ANTEIL gebildeten Formel besteht, die man aus (3) dadurch erhält, indem man für A1 die Übersetzung des Nomens der Ergänzung im Akkusativ einsetzt, für A2 die Übersetzung der Verbergänzung im Nominativ usw.

Aufgrund dieser naheliegenden Beobachtung kommt man zu ersten Regeltypen für eine Übersetzungsgrammatik: Bezeichnet man einen Klammerausdruck wie (3) als Kontextpattern (CP) und faßt man die Hilfssymbole A1, A2 usw. als Kennzeichnung von Leerstellen (SLOT) auf, dann können K o n t e x t p a t t e r n r e g e l n definiert werden, die einem ausgangssprachlichen Symbol ein Kontextpattern zuordnen, sowie L e e r - s t e l l e n r e g e l n , die angeben, wie die Leerstellen des CP mit Teilübersetzungen aus dem Kontext des ausgangssprachlichen Symbols auszufüllen sind.

Für das Verb "enthalten" sind also Regeln zu formulieren, in denen die o.a. Beobachtungen festgehalten werden:

- Kontextpatternregel: Zur Konstruktion der Übersetzung eines Satzes mit "enthalten" ist das Pattern (FORMEL ANTEIL A1 A2 A3 A4) zu verwenden;

- Leerstellenregel für A1: In die Leerstelle A1 ist ein Term der Sorte stoff einzusetzen, den man durch Übersetzung des Nomens der Akkusativergänzung von "enthalten" gewinnt (z.B. die Übersetzung von "Cadmium" in (4));

- Leerstellenregel für A2: In die Leerstelle A2 ist ein Term der Sorte stoffkoll einzusetzten, der aus der Übersetzung der Nominativergänzung von "enthalten" resultiert (z.B. die Übersetzung von "die Abwasserproben in der Firma Müller" in (4)).

usw.

Aus den Überlegungen zur Übersetzung von "enthalten" geht hervor, daß die in Leerstellen einzusetzenden Teilübersetzungen charakterisierbar sind durch:

- zielsprachensyntaktische Bedingungen wie Sortenrestriktionen oder einer Festlegung, daß Argumente eines Prädikats keine Formeln sein dürfen somdern Terme;

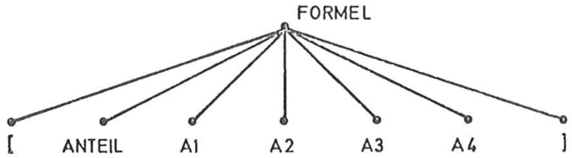
- syntaktische Kategorien und Relationen der Ausgangssprache wie Nomen, Nominalgruppe, Ergänzung etc.

Die Anwendung von Leerstellenregeln setzt demnach voraus, daß sowohl für Teile eines ausgangssprachlichen Satzes als auch für die Teile der daraus erzeugten Übersetzung die entsprechenden syntaktischen Informationen greifbar sind.

Um über solche syntaktischen Informationen zur Ausgangssprache zu verfügen, definieren wir eine Übersetzungsgrammatik so, daß ihre Regeln nicht auf Sätze s einer Ausgangssprache L_{AS} angewendet werden, sondern auf syntaktische Strukturbäume t_s, die die Ableitung von s in einer Regelgrammatik G_{AS} darstellen.

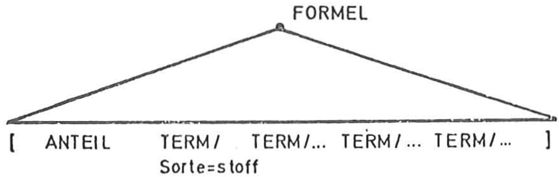
Analog zu den syntaktischen Informationen der Ausgangssprache werden die zielsprachlichen Informationen verfügbar gemacht, indem bei der Erzeugung einer Übersetzung nicht nur Ketten von terminalen Symbolen der Zielsprache aufgebaut werden, sondern ebenfalls Baumstrukturen, aus denen die syntaktische Struktur einer zielsprachlichen Kette ersichtlich ist. Wir haben deshalb im o.a. Beispiel mit dem ANTEIL-Prädikat dem Verb "enthalten" nicht nur den mit eckigen Klammern eingegrenzten Ausdruck zugeordnet, sondern den gesamten Klammerausdruck von (3), der sich auch als Baum (vgl. (7)) darstellen läßt. Diese Struktur ist allerdings stark vereinfacht und stellt keine echte Struktur von

(7)



KS, da die Endkette von (7) in einer Grammatik von KS nicht direkt aus FORMEL ableitbar ist. Der Vereinfachung liegt die in (8) dargestellte Baumkontur zugrunde, in der die Binnenstruktur des Baumes gemäß einer

(8)



Grammatik von KS noch näher zu spezifizieren wäre (vgl. eine solche Spezifizierung für ein dreistelliges Prädikat in Abb. 2.1!). Da jedoch eine solche detaillierte Binnenstruktur für die folgenden Überlegungen ohne Belang ist, verwenden wir einfachheitshalber Ausdrücke wie (3) für Kontextpattern und (8) für ihre Darstellung.

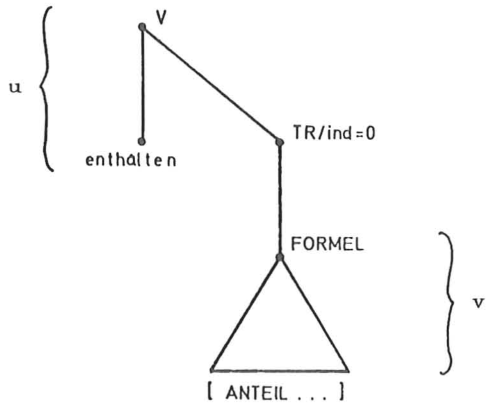
Was bedeutet es nun für die Anwendung einer Kontextpatternregel, daß dem Verb "enthalten", d.h. einer Teilstruktur wie

(9)

die Struktur (8) 'zugeordnet' wird? Legt man diese

Zuordnung so fest, daß in einem Baum mit einer Teilstruktur (9) diese Teilstruktur durch (8) ersetzt wird, dann geht die Information verloren, daß in dem dadurch resultierenden Baum die Teilstruktur (8) aus (9) entstanden ist. Da wir jedoch festgestellt haben, daß wir für die Ausfüllung der Leerstellen eines Kontextpatterns auch syntaktische Information der Ausgangssprache benötigen, diese jedoch bei der Ersetzung verloren gehen würden, ersetzen wir (9) durch die Struktur (10).

(10)



Das bedeutet, daß eine AS-Teilstruktur u nicht durch ihr Kontextpattern v ersetzt wird. Vielmehr wird, prozedural betrachtet, die AS-Teilstruktur u zuerst um einen Knoten erweitert, der mit dem Hilfssymbol TR/ind=∅ markiert ist. Erst dann wird unter diesen neuen Knoten das Kontextpattern 'angehängt'.

Das komplex notierte Hilfssymbol TR wird immer so verwendet, daß gilt: dominiert ein Symbol a ein Symbol TR, dann ist die von TR dominierte Struktur v dem Symbol a bzw. der vom Symbol a dominierten Struktur u als Translat zugeordnet. Wir nennen dann auch v das T r a n s - l a t von a. Durch die Verwendung des Symbols TR kann sichergestellt werden, daß die durch Anwendung von Regeln in einen Baum eingebrachten Translate immer eindeutig identifizierbar und wiederauffindbar sind.

Soll einer Struktur u ein Kontextpattern v zugeordnet werden, dann verwenden wir für diese Festlegung den Formalismus

(11) $u \xrightarrow{CP} v$

oder gleichbedeutend

(12) (CP u v)

so daß wir als Kontextpatternregel für "enthalten" formulieren können:

(13) (V enthalten) $\xrightarrow{CP}$ (FORMEL [ANTEIL A1 A2 A3 A4])

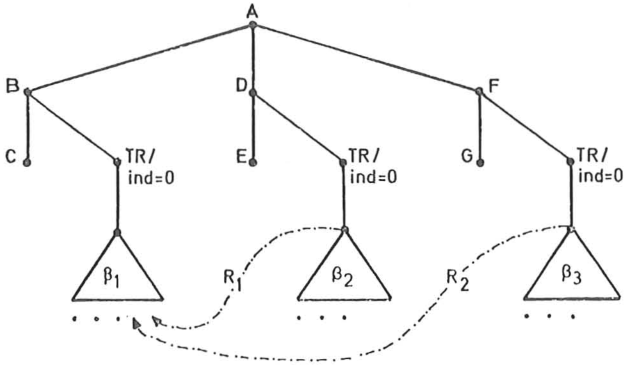
Damit sind bereits wesentliche Konturen einer Übersetzungsgrammatik angedeutet:

- Die Regeln einer Übersetzungsgrammatik werden nicht auf Symbolketten angewendet, sondern auf Bäume. Eine Übersetzungsregel auf einen Baum anwenden heißt dann, durch Ersetzung/Veränderung von Teilbäumen einen neuen Baum erzeugen. Übersetzungsregeln können deshalb als Baumoperationen interpretiert werden.
- Ausgangspunkt einer Übersetzung sind die syntaktischen Strukturbäume der ausgangssprachlichen Formulierungen. Diese Baummenge, die B a s i s einer Übersetzungsgrammatik, ist definierbar mit Hilfe einer Regelgrammatik der Ausgangssprache.

2.3 Translathebung

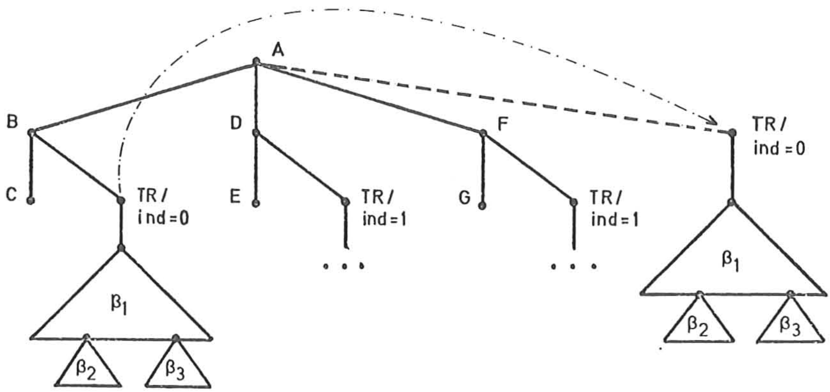
Bevor wir detaillierter auf den Formalismus von Leerstellenregeln eingehen, beschäftigen wir uns zuerst mit der Frage, wie aus den Translaten von Teilstrukturen die Translate einer Gesamtstruktur zu gewinnen sind. Dazu betrachten wir das abstrakte Beispiel (14). Die gestrichelten Pfeile in (14) sollen andeuten, daß die aus ausgefüllten Kontextpattern entstandenen Translate von D und F in Leerstellen des Kontextpatterns β_1 von (B C) aufgrund der Leerstellenregeln R_1 und R_2 für β_1 einzusetzen sind. D.h. zusammen mit der CP-Regel (CP (B C) β_1) formulieren die Regeln R_1 und R_2, wie die Translate von Kontextteilen von (B C) mittels dem Pattern β_1 zu einem Gesamttranslat zusammenzufassen sind.

(14)



Da dadurch bereits die Translate der von A dominierten Teilstrukturen im Translat von B zusammengefaßt sind, kann grundsätzlich angenommen werden, daß das so entstandene Translat von B auch gleichzeitig das Translat von A ist. Um generell feststellen zu können, welches Translat die Translate des Kontexts enthält, verwenden wir das Attribut <u>ind</u> von TR und legen fest: wird ein Translat in eine Leerstelle eingesetzt, dann wird das TR-Symbol, das dieses Translat dominiert, ersetzt durch das Symbol TR/ind=1. Wenn nun ein Symbol wie z.B. A in (15) nur ein Symbol dominiert, das seinerseits TR/ind=Ø dominiert, alle anderen von A dominierten Symbole (wie D und F) das Symbol TR/ind=1 dominieren, dann kann die vor TR/ind=Ø dominierte Struktur dem Symbol A als Translat zugeordnet werden, so daß aus (14) die Struktur (15) entsteht.

(15)



Dieses Prinzip, das wir T r a n s l a t h e b u n g nennen, ist nach unserer Übersicht über die Übersetzung deutsch/KS weitgehend anwendbar und nur für wenige Symbole, die nicht-terminale Symbole dominieren, müssen eigene CP-Regeln definiert werden.

2.4 Ausfüllen von Leerstellen

Wird ein Kontextpattern als Baum dargestellt, sind die Leerstellen Teilbäume dises Baumes und zwar solche Teilbäume, die nur aus einem Knoten bestehen (vgl. A1, A2 usw. in (7)!). Soll nun festgehalten werden, daß z.B. die Leerstelle A2 im Kontextpattern von "enthalten" durch einen Term der Sorte stoffkoll (komplex notiert TERM/Sorte=stoffkoll) auszufüllen ist, dann kann dies durch eine Regel "Ersetze Struktur" (ER.ST) formuliert werden,

(16) (ER.ST A2 TERM/Sorte=stoffkoll)

durch die zum Ausdruck gebracht wird, daß A2 durch einen Teilbaum zu ersetzen ist, dessen Wurzelknoten mit TERM/Sorte=stoffkoll markiert ist. Entsprechend unserer Überlegungen zur zweiten Leerstelle des CP von "enthalten" (vgl. die Erläuterungen zu (3)) muß jedoch noch präzisiert werden, daß

(i) die von TERM/Sorte=stoffkoll dominierte Struktur Translat einer Nominalgruppe im Nominativ (NG/K=nom),

(ii) die Nominalgruppe im Nominativ Ergänzung zum Verb "enthalten"

sein muß.

Unter Verwendung der über den Knoten eines Baumes definierbaren Relation wie Dominanzrelation (DOM), mittelbare Dominanzrelation (DOM*), Linksrelation (LFT) und mittelbare Linksrelation (LFT*) sowie der Junktoren 'und' ($\wedge$) und 'oder' ($\vee$) läßt sich (i) darstellen durch eine Formulierung wie

(17) (DOM NG/K=nom TR/ind=Ø) $\wedge$ (DOM TR/ind=Ø TERM /Sorte=stoffkoll)

wofür wir dann auch schreiben

(18) (TRANSLAT NG/K=nom TERM/Sorte=stoffkoll)

und (18) als Äquivalent zu (17) betrachten.

Da durch (17) bzw. (18) Bedingungen festgelegt werden, die eine Baumstruktur erfüllen muß, damit bestimmte Regeln angewendet werden dürfen, nennen wir solche Bedingungen S t r u k t u r b e d i n g u n g e n.

Nach einem ähnlichen Prinzip wie in (17) bzw. (18) können Strukturbedingungen für nicht-attributive Satzglieder, die wir hier summarisch als Ergänzungen (ERG) bezeichnen, sowie für Attribute (ATTR) definiert werden, wobei die Details von der ausgangssprachlichen Grammatik abhängen, die für die Konstruktion der Basis der Übersetzungsgrammatik zugrundegelegt wird. Es ist allerdings fraglich, inwieweit Strukturbedingungen für Ergänzungen und Attribute so definiert werden können, daß für einen Teilbaum nicht gleichzeitig die Bedingung ERG als auch die Bedingung ATTR zutrifft. In der Praxis hat sich allerdings herausgestellt, daß diese strukturelle Ambiguität im Laufe einer Übersetzung unter Zuhilfenahme der Sorten von KS aufgelöst werden kann.

Gehen wir davon aus, daß die Bedingung ERG definiert sei, kann die Leerstellenregel für die Leerstelle A2 des CP von "enthalten" in Form einer b e d i n g t e n Regel, d.h. als eine Regel, die zusätzlich Strukturbedingungen enthält, geschrieben werden:

(19) ((ER.ST A2 TERM/Sorte=stoffkoll)
((TRANSLAT NG/K=nom TERM/Sorte=stoffkoll)
$\wedge$ (DOM V A2) $\wedge$ (ERG V NG/K=nom)))

Zusätzlich zu (19) ist eine Regel erforderlich, mit der im Hinblick auf die Translathebung das Symbol TR/ind=Ø, das TERM/Sorte=stoffkoll dominiert, ersetzt wird durch das Symbol TR/ind=1. Wir verwenden dazu eine Regel "Ersetzung eines Symbols" (ER.S), geschrieben

(20) $a \xrightarrow{ER.S} b$ bzw. (ER.S a b)

mit der ein Symbol a durch ein Symbol b ersetzt wird. Im Fall der o.a. Leerstellenersetzung wird die Regel als bedingte Regel verwendet:

(21) ((ER.S TR/ind=Ø TR/ind=1)
(DOM TR/ind=Ø TERM/Sorte=stoffkoll))

2.5 Aufpfropfung

Die bisher beschriebene Überführungsstrategie baut auf dem Grundprinzip von Kontextpattern zu ausgangssprachlichen Konstituenten einer Satzstruktur auf, in denen Leerstellen ausgewiesen sind, die durch Translate von anderen Konstituenten der Satzstruktur ausgefüllt werden können. Ein zweites Grundprinzip ist die Translathebung, das für Translate von Strukturen $u_1,\ldots,u_n$, deren Wurzelknoten Geschwister sind, regelt, welches dieser Translate angehoben wird zum Translat der Struktur, deren Wurzelknoten die Strukturen $u_1,\ldots,u_n$ dominiert. Dabei wird dasjenige Translat angehoben, das noch nicht in ein anderes Kontextpattern eingesetzt wurde. Dies kann jedoch nur dann erfolgen, wenn die anderen Translate bereits in Leerstellen eingesetzt wurden. Nun gibt es allerdings Fälle, für die vorhersagbar ist, daß das entstehende Translat in keine Leerstelle eines anderen Kontextpatterns einsetzbar sein wird, so daß das Prinzip der Translathebung versagen würde, da mit Sicherheit mehrere nichteingesetzte Translate vorliegen.

Ein Beispiel für einen solchen Fall ist die Übersetzung von "kein" wie etwa in dem Fragesatz:

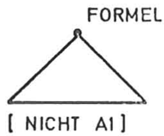
(22) Enthielt keine Probe Zink?

Wir gehen davon aus, daß die entsprechende KS-Formulierung für (22) die negierte Formel

```
(23)  (?(NICHT
         (EXIST x.betrieb (EXIST x.int (EXIST x.abstrobj
                                       (EXIST x.dimzahl
         (ANTEIL ZN (PROBE x.betrieb x.int) x.abstrobj
                                        x.dimzahl))))))
```

ist. D.h. daß das Kontextpattern für "kein" darstellbar ist als

(24)



wobei in die Leerstelle A1 eine FORMEL einzusetzen ist, die aus der Übersetzung des nicht negierten Fragesatzes entsteht, genauer genommen: das Translat des Satzes mit Ausnahme von "kein". Dies läßt sich jedoch nicht auf dem Weg einer Kontextpatternregel für "kein" darstellen (vgl. Abb. 2.2), da in diesem Fall das Translat für NP nicht ermittelbar ist, das jedoch für die Einsetzung in das CP von "enthalten" erforderlich ist, da sowohl NEGDET als auch N Knoten mit der Markierung TR/ind=Ø dominieren.

Um das Prinzip der Translathebung anwenden zu können, müßte die von NEGDET dominierte Struktur aus dem Baum entfernt und irgendwie aufbewahrt werden, bis der Satz ohne NEGDET übersetzt ist, um die dann entstandene Formel in die Leerstelle A1 einzusetzen. Da wir

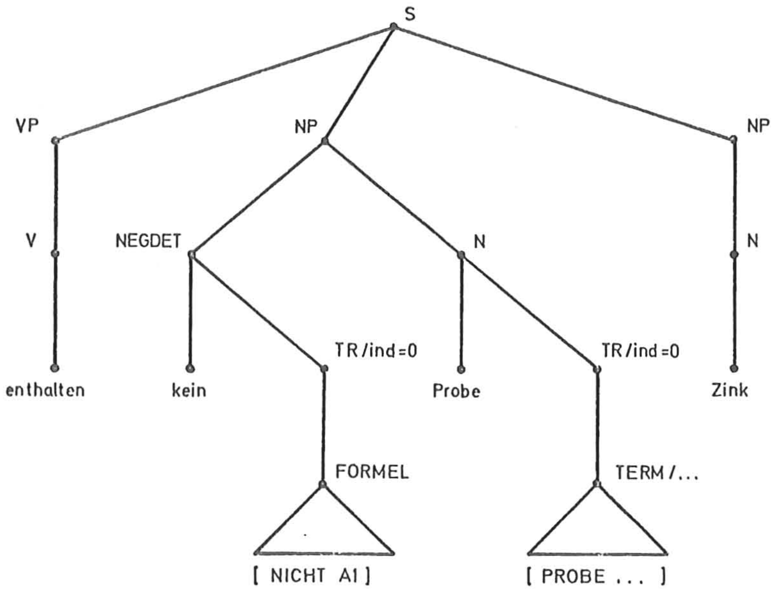


Abbildung 2.2

bei unserem Formalismus keine Möglichkeit haben, Teilstrukturen "zwischenzuspeichern", erhält "kein" als Kontextpattern nur den Baum mit der Wurzelknotenmarkierung TR/ind=1 und (24) wird auf den Gesamtbaum so "aufgepfropft", daß der S-Knoten an der Stelle A1 steht (vgl. Abb. 2.3). Nach der Aufpfropfung kann die von S dominierte Struktur nach den bisher erläuterten Prinzipien weiter überführt werden, bis das Translat von S ermittelt ist. Durch eine besondere Regel (vgl. 2.6) kann dann die von S dominierte Struktur durch das Translat von S ersetzt werden, so daß die angestrebte negierte Formel entsteht.

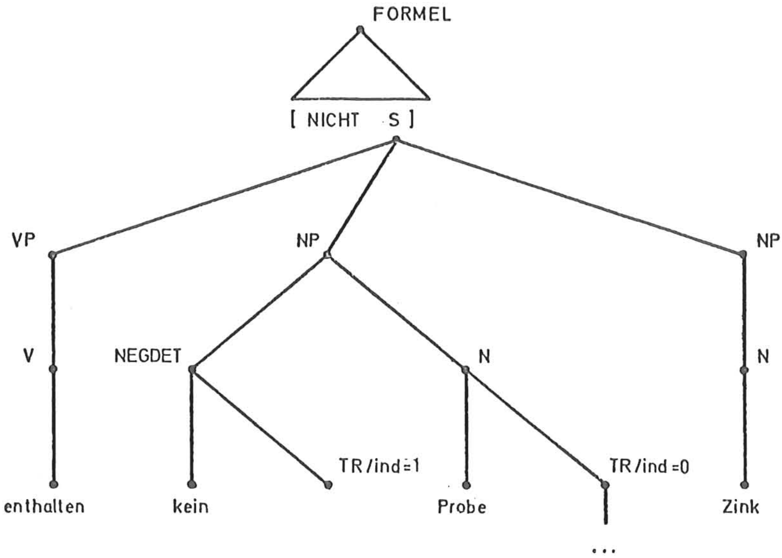


Abbildung 2.3: Aufpfropfung

Zur Beschreibung der Aufpfropfung führen wir eine Regel zur Ersetzung einer Struktur durch ein 'Literal' (ER.LIT) ein, geschrieben

(25) $a \xrightarrow{\text{ER.LIT}} \alpha$ bzw. (ER.LIT a α)

wobei a ein Symbol und α ein Klammerausdruck. Die Regel ist so anzuwenden, daß ein von a dominierter Baum durch den α entsprechenden Baum ersetzt wird.

Für eine zu definierende S t r u k t u r v a r i a b l e & legen wir fest, daß & diejenige Struktur bezeichnet, die in einer Regel von der linken Regelseite bezeichnet wird, so daß die Aufpfropfung der Struktur (24) mit der Regel zu beschreiben ist:

(26) $S \xrightarrow{\text{ER.LIT}}$ (FORMEL [NICHT &])

2.6 Terminierung und Transformation

Unabhängig davon, ob auf einen Baum Teilstrukturen aufgepfropft wurden oder nicht, müssen nach der Translathebung bis zum S-Knoten die ausgangssprachlichen Strukturteile aus einem abgeleiteten Baum entfernt werden, damit eine reine zielsprachliche Struktur entsteht. Dies kann ganz einfach dadurch erfolgen, daß die von S dominierte Struktur durch das Translat von S ersetzt wird, wozu wir die T e r m i n i e r u n g s r e g e l

(27) ((ER.ST S X1) (TRANSLAT S X1))

formulieren.

Es kann durchaus möglich sein, daß die durch Anwendun der Regel (27) entstandene Struktur noch weiter trans formiert werden muß. Dies ist bei einer Übersetzung nach KS dann erforderlich, wenn Leerstellenregeln angegeben werden für den Fall, daß in einem deutschen Satz eine Konstituente gar nicht ausformuliert ist, deren Translat in eine Leerstelle eingesetzt werden sollte. Das die Leerstelle bezeichnende Hilfssymbol wird dann ersetzt durch eine Variable von KS. Es dürf sich als zweckmäßig erweisen, die Bindung solcher Variablen durch Quantoren erst nach Anwendung der Terminierungsregel durchzuführen.

3. ALLGEMEINE FORM EINER ÜBERSETZUNGSGRAMMATIK

Bei der Definition der allgemeinen Form einer Übersetzungsgrammatik gehen wir so vor, daß wir zuerst eine globale Definition einer Übersetzungsgrammatik geben und den dazugehörigen Ableitungsbegriff festlegen. Erst dann werden wir die detaillierten Definitionen zu den Einzelkomponenten vornehmen. Dies hat zur Folge, daß im Definiens einzelner Definitionen bisweilen Bedingungen auftreten, die erst später definiert werden. Wir nehmen jedoch an, daß dieses Vorgehen trotzdem zu einem leichteren Gesamtverständnis beiträgt, da dadurch der Gesamtzusammenhang, in dem eine Definition steht, schon beim ersten Lesen deutlicher wird.

3.1 Definition einer Übersetzungsgrammatik

<u>Def.:</u> (Übersetzungsgrammatik) (Ü1)

Eine <u>Übersetzungsgrammatik</u> ist ein Viertupel

ÜG = (A,F,B,R)

mit folgenden Komponenten:

A ist ein Alphabet, das <u>Alphabet von</u> ÜG,

F ist die Baummenge über dem Alphabet A, die <u>Baummenge von</u> ÜG,

B ist eine Teilmenge von F, die <u>Basis von</u> ÜG,

R ist eine Menge von Regeln über A, die (<u>Überführungs-</u>) <u>Regeln</u> von ÜG.

<u>Erläuterung zur Definition</u>

Das Alphabet A umfaßt die Symbole, mit denen die Bäume von F und B markiert sind. Wie in der Folge näher zu definieren ist, enthält die Basis B diejenigen Bäume, die die Strukturen der Ausgangssprache der Übersetzungsgrammatik beschreiben und die mittels der Regeln von ÜG abgeleitet werden zu Strukturen der Zielsprache. Die Baummenge F enthält somit neben den Bäumen der Basis auch diejenigen Bäume, die durch die Anwendung von Überführungsregeln entstehen.

Bei der Festlegung eines Ableitungsbegriffs für eine Übersetzungsgrammatik unterscheiden wir zwischen einer Überführung und einer Ableitung und zwar in Abhängigkeit davon, ob Bäume mit Hilfe von Überführungsregeln aus einem Baum der Basis B oder aus der Baummenge F erzeugt werden. Wir verwenden dazu den Begriff der Anwendung einer Regel, der im einzelnen jedoch erst bei der Definition der Regeln selbst festgelegt wird.

<u>Def.:</u> (Überführbarkeit) (Ü2)

Sei ÜG = (A,F,B,R) eine Übersetzungsgrammatik.

Ein Baum $u \in F$ heißt <u>(vermöge ÜG) direkt überführbar in</u> $v \in F$, gdw.:

> es gibt eine Regel $r \in R$, so daß v aus u durch Anwendung der Regel r entsteht.

Ein Baum $u \in F$ heißt <u>(vermöge ÜG) überführbar in</u> einen Baum $v \in F$, gdw.:

> es gibt Bäume $u_1, \ldots, u_n \in F$
> mit $n \geq 0$, so daß
> (1) $u_1 = u$
> (2) $u_n = v$
> (3) u_{i-1} ist direkt überführbar in u_i, wobei $2 \leq i \leq n$.

Def.: (Ableitung, Ableitbarkeit, abbrechende Ableitung) (Ü3)

Sei ÜG = (A,F,B,R) eine Übersetzungsgrammatik; sei $n \geq 1$.

Eine Ableitung in ÜG ist eine endliche Folge $(u_1,\ldots,u_n)$, $u_i \in F$, mit

(1) $u_1 \in B$

(2) u_{i-1} ist direkt überführbar in u_i, wobei $2 \leq i \leq n$

$u \in F$ heißt ableitbar in ÜG, gdw.:

es gibt eine Ableitung $(u_1,\ldots,u_n)$ mit $u_n = u$.

Eine Ableitung $(u_1,\ldots,u_n)$ heißt abbrechend, gdw.:

für alle $v \in F$ gilt:
u_n ist nicht überführbar in v.

Ein Baum $u \in F$ heißt terminal abgeleitet (bezüglich ÜG), gdw.:

(1) es gibt eine abbrechende Ableitung $(u_1,\ldots,u_n)$ mit $u_n = u$.

(2) für alle Knoten $k \in Kn(u)$ gilt:
ist k ein terminaler Knoten, dann ist k mit einem terminalen Symbol $a \in VT_{ZS}$ markiert (vgl. Def. Ü4 zu VT_{ZS})

3.2 Das Alphabet einer Übersetzungsgrammatik

Bei der Festlegung des Alphabets einer Übersetzungsgrammatik werden drei Teilalphabete berücksichtigt:

- das Alphabet V_{AS}, mit dessen Symbolen die Bäume der Basis markiert sind (<u>ausgangssprachliches Vokabular</u>);
- das Alphabet V_{ZS}, mit dessen Symbolen die terminal abgeleiteten Bäume aus F markiert sind (<u>zielsprachliches Vokabular</u>);
- das Alphabet V_{AUX}, ein <u>Hilfsalphabet</u> von Symbolen, die in den Bäumen einer Überführung auftreten können, später jedoch wieder getilgt werden.

<u>Def.:</u> (Vokabulare von ÜG) (Ü4)

Sei ÜG = (A,F,B,R) eine Übersetzungsgrammatik.

Ein Vokabular V_{AS} mit $V_{AS} = VN_{AS} \cup VT_{AS}$ ist das <u>ausgangssprachliche Vokabular (AS-Vokabular) von</u> ÜG, wobei:

$$V_{AS} \subseteq A$$

Wir nennen VN_{AS} das <u>nicht-terminale AS-Vokabular von</u> ÜG VT_{AS} das <u>terminale AS-Vokabular von</u> ÜG.

Ein Vokabular V_{ZS} mit $V_{ZS} = VN_{ZS} \cup VT_{ZS}$ ist das <u>zielsprachliche Vokabular (ZS-Vokabular) von</u> ÜG, wobei:

$$V_{ZS} \subseteq A$$

Wir nennen VN_{ZS} das <u>nicht-terminale ZS-Vokabular von</u> ÜG VT_{ZS} das <u>terminale ZS-Vokabular</u> von ÜG.

Ein Vokabular H ist das <u>Hilfsvokabular von</u> ÜG, wobei:

$$H \subseteq A$$

Ein Symbol a $\in V_{AUX}$ nennen wir <u>Hilfssymbol</u>, wobei:

$$V_{AUX} \subseteq H$$

Das Symbol & $\in V_{AUX}$ nennen wir <u>Strukturvariable</u>.

Ein Symbol b $\in V_{VAR}$ nennen wir <u>Symbolvariable</u>, wobei:

$$V_{VAR} \subseteq H, \text{ wobei } V_{VAR} = \{X1,\ldots,Xn\}$$

<u>Def.:</u> (Alphabet von ÜG) (Ü5)

Sei ÜG eine Übersetzungsgrammatik, V_{AS} das AS-Vokabular von ÜG, V_{ZS} das ZS-Vokabular von ÜG und H das Hilfsvokabular von ÜG.

Ein Alphabet A ist das <u>Alphabet von</u> ÜG, gdw.:

$A = V_{ÜG} \cup V_{ÜG}^{(\mathbb{N})}$ wobei (1) $V_{ÜG} = V_{AS} \cup V_{ZS} \cup H$

(2) $V_{ÜG}^{(\mathbb{N})}$ ist das indizier-Alphabet von $V_{ÜG}$

<u>Erläuterung</u>

Die Definition sieht vor, daß es im Alphabet einer Übersetzungsgrammatik nicht nur einfache Symbole gibt wie etwa zum Beispiel NP, VP u.ä., sondern dazu auch indizierte Symbole wie NP_1, NP_2,...,NP_n ($n \in \mathbb{N}$). Solche indizierten Symbole sind erforderlich, wenn es unter den Bäumen von ÜG Bäume gibt, in denen mehrere Knoten mit dem gleichen Symbol markiert sein können, was bei natürlichen Sprachen meistens der Fall ist. Will man nun in einer Regel Festlegungen über verschiedene Knoten treffen, die jedoch gleich markiert sind, dann kann man auf die Nennung der Knoten verzichten, wenn man festlegt, daß z.B. NP_1 und NP_2 in einem Baum zwei verschiedene Knoten zuzuordnen sind, die beide mit dem Symbol NP markiert sind.

3.3 Die Basis einer Übersetzungsgrammatik

<u>Def.:</u> (Basis einer Übersetzunsggrammatik) (Ü6)

Sei ÜG eine Übersetzungsgrammatik; sei $V_{AS} = VN_{AS} \cup VT_{AS}$ das ausgangssprachliche Vokabular von ÜG.

Eine Baummenge B ist die <u>Basis von</u> ÜG, gdw.:

(1) B = { u | (1) alle nichtterminalen Knoten von u sind mit Symbolen aus VN_{AS} markiert;
(2) alle terminalen Knoten von u sind mit Symbolen aus VT_{AS} markiert. }

(2) $B \subset F$ ist entscheidbar

In der Folge gehen wir davon aus, daß die Basis B einer Übersetzungsgrammatik ÜG gegeben ist durch die terminalen Bäume einer kontextfreien Grammatik $G = (VN_{AS}, VT_{AS}, S, P)$, wobei VN_{AS} gleich dem nichtterminalen AS-Vokabular von ÜG, VT_{AS} gleich dem terminalen AS-Vokabular von ÜG mit der Einschränkung $(VN_{AS} \cup VT_{AS}) \not\subseteq V_{ÜG}^{(\mathbb{N})}$, so daß gilt:

$$B = \{ u \mid u \text{ ist ein terminaler Baum von } G \}$$

Dies bedeutet jedoch nicht, daß wir generell die Basis einer Übersetzungsgrammatik auf die Bäume einer kontextfreien Grammatik einschränken. Für die Gesamtdefinition ist es ohne Belang, wie die Basis einer Übersetzungsgrammatik aufgezählt bzw. beschrieben wird.

3.4 Basissprache und erzeugte Sprache

Die Baummenge F einer Übersetzungsgrammatik enthält u.a. drei Teilmengen B, F_I und F_T, wobei:

- B die Bäume der Basis von ÜG
- F_I die Menge der Bäume, die aus den Bäumen der Basis ableitbar sind, jedoch nicht terminal abgeleitet sind
- F_T die Menge der Bäume, die terminal abgeleitet sind.

Faßt man die Wörter der Bäume von B, d.h. die Verkettungen der Markierungen der terminalen Symbole eines Baumes, zusammen in einer Menge L_B, dann kann man L_B als die <u>Basissprache</u> der Übersetzungsgrammatik bezeichnen. Entsprechend kann man eine Menge L_T als die Menge der Wörter der Bäume von F_T festlegen und L_T als die von einer Übersetzungsgrammatik <u>erzeugte Sprache</u> bezeichnen.

<u>Def.:</u> (Basissprache, von ÜG erzeugte Sprache) (Ü7)

Sei ÜG eine Übersetzungsgrammatik; sei V_{AS} das AS-Vokabular von ÜG, V_{ZS} das ZS-Vokabular von ÜG; sei B die Basis und F die Baummenge von ÜG.

Eine Menge $L_B \subseteq V_{AS}^*$ ist die <u>Basissprache von</u> ÜG, gdw.:

$$L_B = \{ \alpha \mid \alpha \text{ ist das Wort von } u, u \in B \}$$

Eine Menge $L_T \subseteq V_{ZS}^*$ ist die <u>von</u> ÜG <u>erzeugte Sprache</u>, gdw.:

$$L_T = \{ \alpha \mid \alpha \text{ ist das Wort von } u, u \in F, \text{ und } u \text{ ist terminal abgeleitet bzgl. ÜG} \}$$

Bemerkung:

Die Basissprache von ÜG bzw. die von ÜG erzeugte Sprache ist nicht ohne weiteres mit der Ausgangssprache bzw. der Zielsprache einer Übersetzung zu identifizieren. Inwieweit eine Basissprache identisch ist mit einer Ausgangssprache und die von ÜG erzeugte Sprache mit einer Zielsprache, ist eine Frage der Adäquatheit von ÜG, d.h. wie adäquat von ÜG eine Übersetzungsbeziehung beschrieben wird.

3.5 Regeln einer Übersetzungsgrammatik

Bei der Festlegung der Regeln einer Übersetzungsgrammatik gehen wir von einem Regelbegriff aus, der drei Aspekte unterscheidet. Zum einen kann eine Regel betrachtet werden als ein formales Objekt, wie etwa wenn wir sagen, eine Regel sei ein Paar, ein Tripel o.ä. und Eigenschaften der Komponenten dieses Objekts angeben.

Ein anderer Aspekt ist die Darstellung einer Regel, d.h. die Regelschreibweise, die systematisch in einer Regelsyntax beschrieben werden kann.

Ein dritter Aspekt ist die Semantik einer Regel, die die Regelverwendung expliziert durch Festlegung des <u>Anwendungsbereichs</u> einer Regel als Menge der Objekte, auf die eine Regel anwendbar ist, sowie durch eine <u>Anwendungsvorschrift</u> für die Regel, die beschreibt, wie aus Objekten des Anwendungsbereichs andere Objekte durch Regelanwendung entstehen. Dadurch ist dann auch der <u>Wertebereich</u> einer Regel als Menge der Objekte bestimmt, die durch die Anwendung der Regel entstehen können.

Ist eine Regel ein Paar $r = (a,b)$, dann bezeichnet man r als zweistellige Regel, die erste Stelle des Paares auch als linke Seite der Regel, die zweite Stelle als rechte Seite der Regel. Üblicherweise charakterisiert dann die linke Regelseite die Objekte des Anwendungsbereichs der Regel in einer durch die Anwendungsvorschrift zu spezifizierenden Weise, während die rechte Regelseite die Objekte des Wertebereichs charakterisiert.

<u>Beispiel:</u>

Sei $G = (VN,VT,S,P)$ eine kontextfreie Grammatik. Als <u>formales Objekt</u> beschrieben ist eine Regel $r \in P$ ein Paar $r = (a,\alpha)$, wobei $a \in VN$ und $\alpha \in A^+$ mit $A = VN \cup VT$. Die <u>Schreibweise</u> der Regel ist $a \longrightarrow \alpha$. Der <u>Anwendungsbereich</u> von r ist diejenige Teilmenge von A^+, die Wörter der Form $\beta = (x_1, \ldots, a, \ldots, x_n)$ enthält. Die Anwendungsvorschrift legt fest, daß ein Wort γ aus einem Wort β durch Anwendung der Regel r dadurch entsteht, daß das Symbol a in β durch α ersetzt wird.

Die Regeln einer Übersetzungsgrammatik bauen auf zweistelligen Regeln auf, deren Anwendungsbereiche Teilmengen der Baummenge einer Übersetzungsgrammatik sind. Diese einfachen Regeln beschreiben entweder die Ersetzung von Knotenmarkierungen durch andere Symbole oder die Ersetzung bzw. Umstrukturierung von Teilbäumen.

Da das zu ersetzende Symbol bzw. die zu ersetzende Teilstruktur als Charakterisierung des Anwendungsbereichs einer Regel nicht immer ausreichend ist, wird neben solchen einfachen zweistelligen Regeln ein weiterer Regeltyp eingeführt, der es erlaubt, den Anwendungsbereich einer Regel durch Strukturbedingungen einzuschränken.

Im Hinblick auf die Verwendung von Symbolvariablen und indizierten Symbolen sind darüberhinaus aus einfachen Regeln zusammengesetzte komplexere Regeln erforderlich, in deren Anwendungsvorschrift festgelegt ist, wie die in den einfachen Regeln einer komplexen Regel enthaltenen Variablen und indizierten Symbole einheitlich zu interpretieren sind.

Bevor wir jedoch auf die Regeln im einzelnen eingehen, definieren wir zuerst den Begriff der Knotenzuordnung. Die Knotenzuordnung spielt insofern eine wichtige Rolle für die Festlegung der Anwendung von Regeln, als in den Regeln die Symbole des Alphabets einer Übersetzungsgrammatik verwendet werden zur Bezeichnung der Knoten von Bäumen aus der Baummenge F einer Übersetzungsgrammatik. Aufgabe der Knotenzuordnung ist es, diese Bezeichnungsrelation zu explizieren. Dabei ist folgendes zu beachten:

1. Wir gehen davon aus, daß die Bäume aus F weder mit indizierten Symbolen noch mit Symbolvariablen markiert sind.

2. Verschiedene Knoten eines Baumes können mit dem gleichen Symbol indiziert sein.

3. Symbolvariablen werden in Regeln verwendet, um einen beliebigen Knoten eines Baumes zu bezeichnen bzw. präziser formuliert: um einen beliebigen Knoten zu bezeichnen, der mit einem beliebigen Symbol markiert ist.

4. Mehrere verschieden indizierte Symbole werden in einer Regel verwendet, um verschiedene Knoten zu bezeichnen, die jedoch alle mit dem entsprechenden nichtindizierten Symbol markiert sind.

Daraus folgt, daß ein Symbol einer Regel nicht unbedingt eindeutig einen Knoten in einem Baum bezeichnet. M. a. W. einem Symbol einer Regel können in einem Baum möglicherweise mehrere Knoten zugeordnet werden.

Wir unterscheiden in der Folge zwischen der grundsätzlichen Zuordenbarkeit von Knoten eines Baumes zu Symbolen einer Regel und einer konkreten Zuordnung eines Knotens zu einem Symbol.

Def.: (Knotenzuordnung) (Ü8)

Sei A das Alphabet, F die Baummenge einer Übersetzungsgrammatik ÜG; seien $V_{AUX} \subseteq A$ die Symbolvariablen von ÜG; sei $u \in F$ ein Baum; sei φ eine Indexbeseitigung.

Ein Knoten $k \in Kn(u)$ ist einem Symbol $a \in A$ in einem Baum u zuordenbar, gdw.:

$(\varphi(a),k) \in M(u) \vee a \in V_{AUX}$

Ist $\alpha = a_1,\ldots,a_n$ eine Folge von Symbolen, $K = k_1,\ldots,k_n$ eine Folge von Knoten, dann nennen wir das Paar $\omega = (\alpha,K)$ eine Knotenzuordnung für α (in u), gdw.:

> $(\forall i,\ i \leq n)$ gilt:
> k_i ist ein dem Symbol a_i in u zuordenbarer Knoten.

Wir sagen dann auch, k_i ist a_i vermöge ω zugeordnet.

Erläuterung:

Während den Symbolvariablen aus V_{AUX} beliebige Knoten eines Baumes zuordenbar sind, ist einem nichtindizierten Symbol nur dann ein Knoten zuordenbar, wenn dieser Knoten auch mit diesem Symbol markiert ist. Einem indizierten Symbol $a^{(i)}$ kann ein Knoten zugeordnet werden, der mit dem entsprechenden nichtindizierten Symbol a markiert ist.

Schreibkonvention für Knotenzuordnungen

Zur Bezeichnung von Knotenzuordnungen reservieren wir das Zeichen ω und schreiben für ein Paar (α,K) auch $\omega_u(\alpha)$ bzw. $\omega(\alpha)$, wenn $\omega = (\alpha,K)$ eine Knotenzuordnung für α in u ist.

Wollen wir ausdrücken, daß eine Knotenzuordnung durch Verknüpfung einzelner Knotenzuordnungen entstanden ist, d.h. gilt

$$\omega(\alpha) = ((a_1,\ldots,a_n)(k_1,\ldots,k_n))$$
$$\omega(\beta) = ((b_1,\ldots,b_m)(t_1,\ldots,t_m))$$
$$\omega(\gamma) = ((a_1,\ldots,a_n,b_1,\ldots,b_m)(k_1,\ldots,k_n,t_1,\ldots,t_m))$$

dann schreiben wir auch

$$\omega(\gamma) = \omega(\alpha) \circ \omega(\beta)$$

Def.: (Knotenzuordnung für Listen) (Ü9)

Sei A das Alphabet, F die Baummenge einer Übersetzungsgrammatik; sei l eine Liste über A; sei a ein Symbol aus A; sei u ein Baum aus F.

Eine Knotenzuordnung für l (in u) definieren wir rekursiv:

1. Ist l = a oder l = (a) eine Liste und ist k ein dem Symbol a in u zuordenbarer Knoten, dann ist das Paar (a,k) eine Knotenzuordnung für l in u.

2. Ist $l = (a, l_1, \ldots, l_n)$, $n > 0$, mit der Symbolfolge $\lambda = (a, \lambda_1, \ldots, \lambda_n)$, ist k ein dem Symbol a in u zuordenbarer Knoten und ist K_i eine Knotenfolge, so daß das Paar (λ_i, K_i) eine Knotenzuordnung für die Liste l_i in u ist mit $i \leq n$, dann ist ein Paar (λ, K) mit der Knotenfolge $K = k, K_1, \ldots, K_n$ eine Knotenzuordnung für l in u, gdw.:

   $(\forall\, i,\ i \leq n)\ (\forall\, j,\ j \leq (n-1))$

   (1) $(\exists\, m,\ m > 0)$

   $l_i = (a_i, l_{i_1}, \ldots, l_{i_m}) \longrightarrow K_i = k_i, K_{i_1}, \ldots, K_{i_m}$

   wobei gilt:

   k_i ist ein dem Symbol a_i in u zuordenbarer Knoten und

   (λ_{i_x}, K_{i_x}) ist eine Knotenzuordnung für l_{i_x} in u mit $x \leq m$.

   (2) $l_i = a_i \longrightarrow K_i = k_i$

   wobei gilt:

   k_i ist ein dem Symbol a_i in u zuordenbarer Knoten.

   (3) $(k, k_i) \in D(u)$

   (4) $(k_j, k_{j+1}) \in L(u)$

3. Andere Knotenzuordnungen sind keine Knotenzuordnungen für 1.

In Anlehnung an Definition Ü8 bezeichnen wir eine Knotenzuordnung für eine Liste 1 mit $\omega(1)$. (Zur näheren Erklärung der Definition vgl. die Erläuterung zu Definition Ü10!)

<u>Def.:</u> (zugeordneter Baum, nichtindizierter Baum) (Ü10)

Sei 1 eine Liste; seien u, v Bäume; sei (λ,K) eine Knotenzuordnung für 1 in u.

v ist der der Liste 1 <u>vermöge</u> (λ,K) <u>in</u> u <u>zugeordnete</u> Baum, gdw.:

$K = Kn(v)$
d.h. wenn K und Kn(v) mengentheoretisch gleich sind.

Ist v ein Baum über einem nichtindizierten Alphabet, dann nennen wir v auch den <u>nichtindizierten</u> <u>Baum</u> von 1.

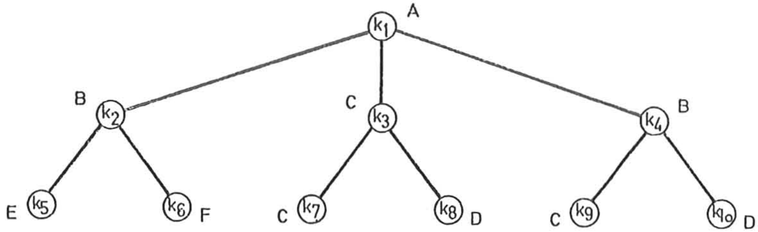
<u>Erläuterung:</u>

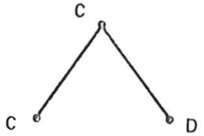
Zur Verdeutlichung der Definition Ü9 und Ü10 betrachten wir die Liste $1 = (X1\ C_1\ D_1)$, wobei X1 eine Symbolvariable sei. Sei u der in Abb. 3.1 dargestellte Baum, dann gibt es für 1 die Knotenzuordnungen

$$\omega_1(1) = ((X1,C_1,D_1)(k_3,k_7,k_8))$$
$$\omega_2(1) = ((X1,C_1,D_1)(k_4,k_9,k_{10}))$$

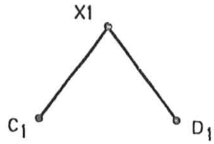
so daß der Baum v_1 von Abb. 3.1 der der Liste 1 vermöge $\omega_1(1)$ in u <u>zugeordnete</u> <u>Baum</u> ist. Davon zu unterscheiden ist der als Baum v_2 abgebildete <u>Baum</u> <u>von</u> 1, der kein Baumausschnitt von u ist.

Baum u





Baum v_1.



Baum v_2.

Abbildung 3.1

Def.: (einheitliche Knotenzuordnung) (Ü11)

Sei $V^{\mathbb{N}}$ das indizierte Alphabet einer Übersetzungsgrammatik ÜG; sei V_{AUX} die Menge der Symbolvariablen von ÜG; seien $\omega(\alpha)$, $\omega(\beta)$, $\omega_1,\ldots,\omega_n$ Knotenzuordnungen.

Eine Knotenzuordnung $\omega(\alpha) = ((a_1,\ldots,a_n),(k_1,\ldots,k_n))$ heißt zuordnungsgleich, gdw.:

$(\forall i \leq n)(\forall j \leq n)$

(1) $a_i = a_j \longrightarrow k_i = k_j$

(2) $a_i, a_j \in (V_{AUX} \cup V^{\mathbb{N}}) \wedge a_i \neq a_j \longrightarrow k_i \neq k_j$

Zwei Knotenzuordnungen $\omega(\alpha) = ((a_1,\ldots,a_n),(k_1,\ldots,k_n))$ und $\omega(\beta) = ((b_1,\ldots,b_m),(t_1,\ldots,t_m))$ heißen *einheitlich*, gdw.:

$(\forall i \leq n)(\forall j \in m)$

(1) $\omega(\alpha)$ ist zuordnungsgleich

(2) $\omega(\beta)$ ist zuordnungsgleich

(3) $a_i = b_j \longrightarrow k_i = t_j$

(4) $a_i, b_j \in (V_{AUX} \cup V^{\mathbb{N}}) \wedge a_i \neq b_j \longrightarrow k_i \neq t_j$

Eine Folge von Knotenzuordnungen $\omega_1,\ldots,\omega_n$ heißt einheitlich, gdw.:

$(\forall i \leq n)(\forall j \leq n)$

ω_i und ω_j sind einheitliche Knotenzuordnungen.

Erläuterung:

Für eine *zuordnungsgleiche Knotenzuordnung* wird festgelegt, daß gleichen Symbolen gleiche Knoten und unterschiedlich indizierten Symbolen bzw. verschiedenen Symbolvariablen verschiedene Knoten zugeordnet werden. Dies hat zur Folge, daß Knotenzuordnungen für Listen nicht unbedingt zuordnungsgleich sein müssen. So ist z.B. die Knotenzuordnung für eine Liste 1 = (A B C B) bezüglich des Baumes u von Abb. 3.1 mit

$$\omega(1) = ((A,B,C,B),(k_1,k_2,k_3,k_4))$$

nicht zuordnungsgleich, da dem Symbol B einmal der Knoten k_2 und einmal der Knoten k_4 zugeordnet wird.

Mit dem Begriff der *einheitlichen Knotenzuordnung* können verschiedene Knotenzuordnungen aufeinander bezogen werden, und zwar in der Weise, daß erforderliche gleiche bzw. unterschiedliche Zuordnungen von einzelnen Knoten zu Symbolen einheitlich erfolgen.

3.5.1 Strukturbedingungen

Für die Definitionen von Strukturbedingungen gehen wir von folgenden Festlegungen aus:

> Sei ÜG = (A,F,B,R) eine Übersetzungsgrammatik; sei $u \in F$ ein Baum; seien $a,b \in A$ Symbole.

Def.: (Dominanzbedingung) (Ü12)

Eine Dominanzbedingung ist ein Paar sb = (a,b), geschrieben (DOM a b).

Ein Baum u erfüllt die Dominanzbedingung (DOM a b) unter einer Knotenzuordnung ω, gdw.:

> es gibt Knoten k_1,k_2, so daß gilt:
> (1) k_1 ist a vermöge ω zugeordnet
> (2) k_2 ist b vermöge ω zugeordnet
> (3) $(k_1,k_2) \in D(u)$

Erläuterung:

Aufgrund der Festlegung für die Knotenzuordnung erfüllt ein Baum u die Dominanzbedingung (DOM a b), wenn ein Knoten k_1 einen Knoten k_2 in u dominiert (3), wobei k_1 entweder mit dem Symbol a markiert ist oder mit dem entsprechenden nichtindizierten Symbol zu a oder aber mit einem beliebigen Symbol, wenn a eine Symbolvariable ist. Entsprechendes gilt für k_2.

Def.: (mittelbare Dominanzbedingung) (Ü13)

Eine mittelbare Dominanzbedingung ist ein Paar sb = (a,b), geschrieben (DOM* a b).

Ein Baum u erfüllt die mittelbare Dominanzbedingung (DOM* a b) unter einer Knotenzuordnung ω, gdw.:

es gibt Knoten k_1, k_2, so daß gilt:
(1) k_1 ist a vermöge ω zugeordnet
(2) k_2 ist b vermöge ω zugeordnet
(3) $(k_1, k_2) \in D^*(u)$

<u>Def.:</u> (Linksbedingung) (Ü14)

Eine <u>Linksbedingung</u> ist ein Paar sb = (a,b), geschrieben (LFT a b).

Ein Baum u <u>erfüllt</u> die Linksbedingung (LFT a b) <u>unter einer Knotenzuordnung</u> ω, gdw.:

es gibt Knoten k_1, k_2, so daß gilt:
(1) k_1 ist a vermöge ω zugeordnet
(2) k_2 ist b vermöge ω zugeordnet
(3) $(k_1, k_2) \in L(u)$

<u>Def.:</u> (mittelbare Linksbedingung) (Ü14a)

Eine <u>mittelbare Linksbedingung</u> ist ein Paar sb = (a,b), geschrieben (LFT* a b).

Ein Baum u <u>erfüllt</u> die mittelbare Linksbedingung <u>unter einer Knotenzuordnung</u> ω, gdw.:

es gibt Knoten k_1, k_2, so daß gilt:
(1) k_1 ist a vermöge ω zugeordnet
(2) k_2 ist b vermöge ω zugeordnet
(3) $(k_1, k_2) \in L^*(u)$

<u>Def.:</u> (Patternbedingung) (Ü15)

Eine <u>Patternbedingung für Teilbäume</u> bzw. <u>Baumausschnitte</u> ist eine Liste l über dem Alphabet A, geschrieben (PATTERN.TB l) bzw. (PATTERN.BA l).

Ein Baum u erfüllt die Patternbedingung (PATTERN.TB 1) bzw. (PATTERN.BA 1) unter einer Knotenzuordnung ω, gdw.:

es gibt Knoten $k_1, \ldots, k_n \in Kn(u)$,
es gibt Symbole $a_1, \ldots, a_n$, so daß gilt:
(1) $a_1 \ldots a_n$ ist die Symbolfolge von 1
(2) $(\forall i \leq n)$
k_i ist vermöge ω dem Symbol a_i zugeordnet
(3) $\omega'(1) = ((a_1, \ldots, a_n), (k_1, \ldots, k_n))$ ist eine Knotenzuordnung für 1
(4) es gibt einen Baumausschnitt v von u, so daß gilt:
v ist der der Liste 1 in u vermöge $\omega'(1)$ zugeordnete Baum.

<u>Erläuterung:</u>

Die Patternbedingung ermöglicht es zu formulieren, daß ein Baum v als Teilbaum oder Baumausschnitt in einem Baum u vorhanden sein muß. Dies kann zwar auch durch eine entsprechende Konjunktion von Dominanz- und Linksbedingungen dargestellt werden, ist jedoch von der Darstellung her aufwendig. Die Ausdrucksmächtigkeit der Patternbedingung könnte noch wesentlich gesteigert werden, wenn im Hilfsalphabet einer Übersetzungsgrammatik Symbole vorgesehen würden, die als obligatorische und fakultative Baumvariablen interpretiert werden. D.h. an einem Beispiel erklärt: sei Y eine Baumvariable und (PATTERN.TB (A B Y C)) eine Patternbedingung, dann erfüllt ein Baum u diese Patternbedingung, wenn es den in Abb. 3.2 dargestellten Baum v gibt mit den (in der Abbildung nicht näher markierten) Knoten $k_1, \ldots, k_n$ mit $n > 0$ bei Y als obligatorischer Baumvariablen bzw. $n \geq 0$ mit Y als fakultativer Baumvariablen, so daß v ein Teilbaum von u ist.

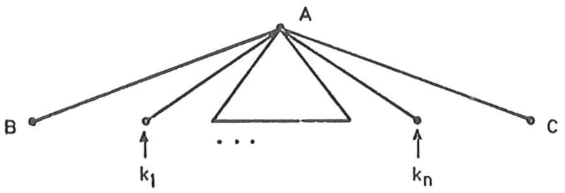


Abbildung 3.2

Def.: (Gleichheitsbedingung) (Ü16)

Eine Gleichheitsbedingung ist ein Paar sb = (a,b), geschrieben (EQ a b).

Ein Baum u erfüllt die Gleichheitsbedingung (EQ a b) unter einer Knotenzuordnung ω, gdw.:

es gibt Knoten $k_1, k_2 \in Kn(u)$, so daß gilt:
(1) k_1 ist a vermöge ω zugeordnet
(2) k_2 ist b vermöge ω zugeordnet
(3) $k_1 = k_2$

Erläuterung:

Die Gleichheitsbedingung weicht insofern von den bisher definierten Bedingungen ab, als sie in der Hauptsache keine Bedingung über die Struktur eines Baumes formuliert, sondern vielmehr Bedingungen für die Knotenzuordnung festlegt. Eine solche Bedingung wird dazu verwendet, um die Menge der einer Symbolvariablen zuordenbaren Knoten einzuschränken wie z.B. durch die Gleichheitsbedingung (EQ X NP). Allerdings wird eine solche Gleichheitsbedingung erst sinnvoll bei konjunktiven oder disjunktiven Verküpfungen von Bedingungen, die wir weiter unten definieren werden.

Def.: (Strukturbedingung) (Ü17)

Rekursive Definition von Strukturbedingungen:

1. Dominanzbedingungen, mittelbare Dominanzbedingungen, Linksbedingungen, mittelbare Linksbedingungen, Patternbedingungen und Gleichheitsbedingungen sind Strukturbedingungen.

2. Sind $sb_1, \ldots, sb_n$ Strukturbedingungen, dann auch (UND $sb_1 \ldots sb_n$) und (ODER $sb_1 \ldots sb_n$).

3. Sind sb_1 und sb_2 Strukturbedingungen, dann auch (IMPLIK sb_1 sb_2).

4. Ist sb eine Strukturbedingung, dann auch (NON sb).

5. Ist sb eine Strukturbedingung und a ein Symbol des Alphabets von ÜG, dann sind auch (FÜRALL a sb) und (EXIST a sb) Strukturbedingungen.

6. Andere Strukturbedingungen gibt es nicht.

Ein Baum u erfüllt die Strukturbedingung (UND $sb_1 \ldots sb_n$) unter einer Knotenzuordnung ω, gdw.:

$(\forall i \leq n)(\forall j \leq n)$

(1) u erfüllt die Strukturbedingung sb_i unter der Knotenzuordnung ω.

(2) ω ist eine zuordnungsgleiche Knotenzuordnung.

Ein Baum u erfüllt die Strukturbedingung (ODER $sb_1 \ldots sb_n$) unter einer Knotezuordnung ω, gdw.:

$(\exists i \leq n)$

u erfüllt die Strukturbedingung sb_i unter der Knotenzuordnung ω.

Ein Baum u <u>erfüllt</u> die Strukturbedingung (IMPLIK sb_1 sb_2) <u>unter</u> <u>einer</u> <u>Knotenzuordnung</u> ω, gdw.:

<u>wenn</u> u die Strukturbedingung sb_1 unter der Knotenzuordnung ω erfüllt,

<u>dann</u> erfüllt u auch die Strukturbedingung sb_2 unter der Knotenzuordnung ω.

Ein Baum u <u>erfüllt</u> die Strukturbedingnung (NON sb) <u>unter</u> <u>einer</u> <u>Knotenzuordnung</u> ω, gdw.:

u erfüllt die Strukturbedingung sb unter einer Knotenzuordnung ω nicht.

Ein Baum u <u>erfüllt</u> die Strukturbedingung (FÜRALL a sb) <u>unter</u> <u>einer</u> <u>Knotenzuordnung</u> ω, gdw.:

für alle Knoten $k \in Kn(u)$ gilt:

wenn k ein dem Symbol a zuordenbarer Knoten ist und wenn $\omega' = \omega \circ (a,k)$ eine zuordnungsgleiche Knotenzuordnung ist,

dann erfüllt u die Strukturbedingung sb unter der Knotenzuordnung ω'.

Ein Baum u <u>erfüllt</u> die Strukturbedingung (EXIST a sb) <u>unter</u> <u>einer</u> <u>Knotenzuordnung</u> ω, gdw.:

es gibt einen Knoten $k \in Kn(u)$, so daß gilt:

(1) k ist ein dem Symbol a zuordenbarer Knoten
(2) $\omega \circ (a,k)$ ist eine zuordnungsgleiche Knotenzuordnung
(3) u erfüllt die Strukturbedingung sb unter der Knotenzuordnung $\omega \circ (a,k)$.

S c h r e i b k o n v e n t i o n für Strukturbedingungen:

1. Ist (P a a_i) bzw. (P a_i a), $i \leq n$, eine Strukturbedingnung mit $P \in \{$ DOM, DOM*, LFT, LFT*, EQ $\}$, dann schreiben wir für eine Strukturbedingnung

   (ODER (P a a_1) ... (P a a_n)) auch (P a (ODER a_1 ... a_n))
   bzw. für
   (ODER (P a_1 a) ... (P a_n a)) auch (P (ODER a_1 ... a_n) a)

   Entsprechendes gilt für Strukturbedingungen mit UND.

2. Ist sb eine Strukturbedingung, dann schreiben wir statt einer Strukturbedingung

   sb_1 = (FÜRALL a (IMPLIK sb (NON (EQ a b)))

   eine Strukturbedingung sb_2, wobei sb_2 aus sb entsteht durch Ersetzung aller b durch $\neq$b.

   Beispiel:

   Statt (FÜRALL X (IMPLIK (DOM A X) (NON (EQ X D))))
   schreiben wir (DOM A $\neq$D)

Erläuterung:

Ausdrücke, für die unter 2. eine Schreibkonvention eingeführt wird, sind erforderlich, da aufgrund des Zusammenhangs zwischen Knotenzuordnung und der Erfüllung von Strukturbedingungen Restriktionen bezüglich der Negation einer Strukturbedingung mit NON entstehen. Dies sei an folgendem Beispiel erklärt:

Sei u der in Abb. 3.3 dargestellte Baum und es soll mit Hilfe von Strukturbedingungen für einen Knoten k von u folgende Festlegung gemacht werden:

1. k soll mit dem Symbol A markiert sein;
2. k soll keinen Knoten mit der Markierung D dominieren.

Formuliert man diese Bestimmungen mit der Strukturbedingung sb_1 = (NON (DOM A D)), dann erfüllt u diese Strukturbedingung unter den Knotenzuordnungen

$$\omega_1 = ((A,D)(k_1,k_5))$$
$$\omega_2 = ((A,D)(k_1,k_7)),$$

was der obigen Festlegung entspricht. Definitionsgemäß erfüllt u die Bedingung sb_1 jedoch auch unter den Knotenzuordnungen

$$\omega_3 = ((A,D)(k_4,k_5))$$
$$\omega_4 = ((A,D)(k_4,k_6)),$$

da bei ω_3 der A zugewiesene Knoten k_4 den D zugewiesenen Knoten k_5 nicht dominiert und ähnlich bei ω_4, so daß ebenfalls die Bedingung (DOM A D) nicht erfüllt, bzw. ihre Negation erfüllt ist. Demnach ist sb_1 keine adäquate Formulierung für die unter 1. und 2. genannten Bedingungen.

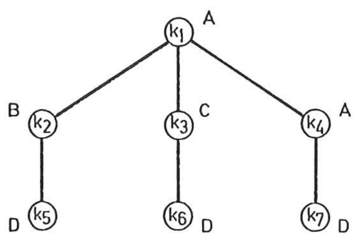


Abbildung 3.3

3.5.2 Ersetzungsregeln

Für die Definitionen der einfachen Regeln einer Übersetzungsgrammatik gehen wir von folgenden Festlegungen aus:

Sei ÜG = (A,F,B,R) eine Übersetzungsgrammatik; seien a,b $\in$ A Symbole; seien s_1,s_2 Listen über A; seien u,v $\in$ F Bäume; sei φ eine Indexbeseitigung; sei ω eine Knotenzuordnung.

Def.: (Regel ER.S - "Ersetze Symbol") (Ü18)

Eine Regel "Ersetze Symbol (durch Symbol)" (ER.S)ist ein Paar von Symbolen, r = (a,b), geschrieben (ER.S a b).

Ein Baum v entsteht aus einem Baum u durch Anwendung der Regel (ER.S a b) unter einer Knotenzuordnung ω, gdw.:

es gibt einen Knoten k $\in$ Kn(u), so daß gilt:
(1) k ist dem Symbol a vermöge ω zugeordnet
(2) v entsteht aus u durch Ersetzung der Markierung von k durch die Markierung φ(b).

Def.: (Regel ER.ST - "Ersetze Struktur") (Ü19)

Eine Regel "Ersetze Struktur (durch Struktur)" (ER.ST) ist ein Paar von Listen, r = (s_1,s_2), geschrieben (ER.ST s_1 s_2).

Ein Baum v entsteht aus einem Baum u durch Anwendung der Regel (ER.ST s_1 s_2) unter einer Knotenzuordnung ω, gdw.:

es gibt Bäume u_1,u_2, so daß gilt:
(1) u_1,u_2 sind terminale Teilbäume von u
(2) der vermöge ω der Liste s_1 in u zugeordnete Baum

ist koradikaler Teilbaum von u_1

(3) der vermöge ω der Liste s_2 in u zugeordnete Baum ist koradikaler Teilbaum von u_2

(4) v entsteht aus u durch Ersetzung von u_1 durch u_2.

Erläuterung:

Eine Regel "Ersetze Struktur (durch Struktur)" formuliert das "Umhängen" von terminalen Teilbäumen eines Baumes, d.h. die Ersetzung eines Teilbaumes u_1 durch einen im Baum vorhandenen Teilbaum u_2. Die Regel ist so definiert, daß die den Listen s_1 und s_2 zugeordneten Bäume nicht identisch mit u_1 und u_2, sondern lediglich koradikale Teilbäume von u_1 und u_2 sein müssen. Sofern es zur Identifizierung der Teilbäume ausreicht, können s_1 und s_2 auch nur Listen von Bäumen sein, die nur aus einem Knoten bestehen.

Beispiel:

Abb. 3.4 illustriert die Anwendung der folgenden Regeln auf einen Baum u mit dem Resultat v:

Regel 1: (ER.ST (B D E) (C F G))

Regel 2: (ER.ST B C)

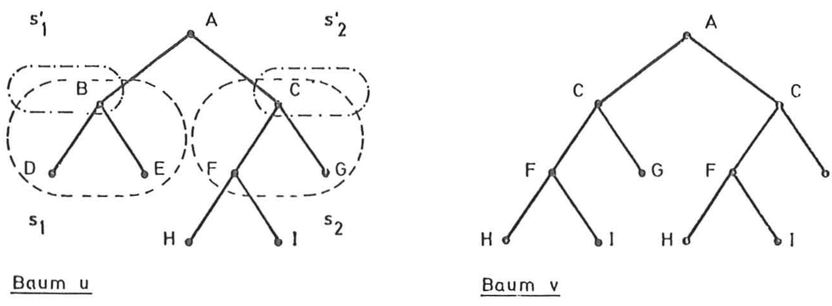


Abbildung 3.4: Anwendung der Regeln 1 und 2

Def.: (Regel ER.LIT - "Ersetze Struktur durch Literal) (Ü20)

Eine Regel "Ersetze Struktur durch Literal" (ER.LIT) ist ein Paar von Listen, $r = (s_1, s_2)$, geschrieben (ER.LIT s_1 s_2).

Ein Baum v entsteht aus einem Baum u durch Anwendung der Regel (ER.LIT s_1 s_2) unter einer Knotenzuordnung ω, gdw.:

Fall 1: die Strukturvariable & ist kein Symbol der Symbolfolge von s_2 und es gilt:

(1) es gibt einen Baum u_1, der terminaler Teilbaum von u ist

(2) der vermöge ω der Liste s_1 in u zugeordnete Baum ist koradikaler Teilbaum von u_1

(3) v entsteht aus u durch Ersetzung von u_1 durch den nichtindizierten Baum von s_2.

Fall 2: die Strukturvariable & ist ein Symbol der Symbolfolge von s_2 und es gilt:

(1) es gibt einen Baum u_1, der terminaler Teilbaum von u ist

(2) der vermöge ω der Liste s_1 zugeordnete Baum ist koradikaler Teilbaum von u_1

(3) es gibt einen Baum u_2, der der nicht indizierte Baum von s_2 ist

(4) es gibt einen terminalen Teilbaum q von u_2, so daß $Kn(q) = \{k\} \wedge (\&, k) \in M(q)$, d.h. q hat nur einen Knoten, der mit der Strukturvariablen & markiert ist

(5) es gibt einen Baum u_3, der aus u_2 durch Ersetzung des Teilbaums q von u_2 durch u_1 entsteht

(6) v entsteht aus u durch Ersetzung von u_1 durch u_3.

Erläuterung zu Fall 1:

Bei der Regel "Ersetze Struktur durch Literal" wird ein Teilbaum u_1 eines Baumes u nicht durch einen im Baum u bereits vorhandenen Teilbaum ersetzt, sondern durch einen neuen Baum, dessen Liste vollständig als zweites Argument der Regel formuliert ist (vgl. Regel 3 im folgenden Beispiel).

Ist das zweite Argument gleich der leeren Liste, ist die Regel eine Tilgungsregel, da der Teilbaum u_1 durch den leeren Baum ersetzt wird (vgl. Regel 4 im folgenden Beispiel).

Beispiel:

Abb. 3.5 illustriert die Anwendung der Regeln 3 und 4 auf einen Baum u mit dem Resultat v:

Regel 3: (ER.LIT B (A E D))

Regel 4: (ER.LIT F ())

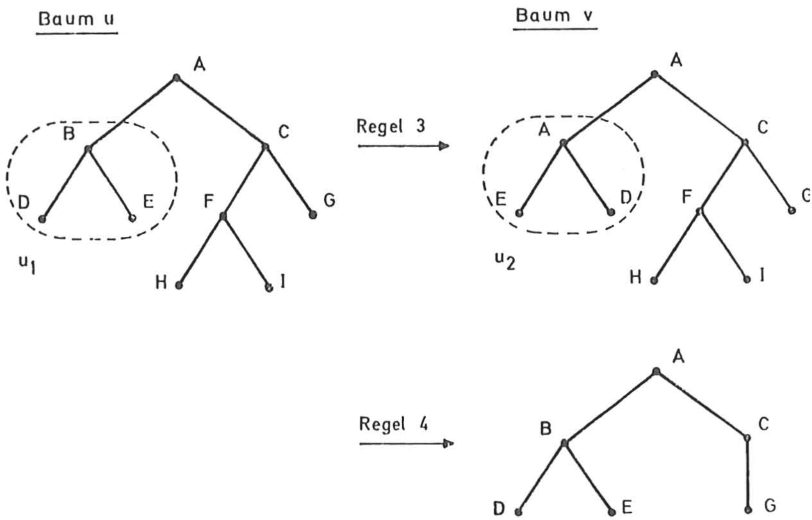


Abbildung 3.5: Anwendung der Regeln 3 und 4

Erläuterung zu Fall 2:

Durch die Verwendung der Strukturvariablen & kann formuliert werden, daß ein Teilbaum u_1 durch einen Baum u_3 ersetzt wird, wobei u_3 den Baum u_1 als Teilbaum enthält. Der Wurzelknoten von u_1 steht in u_3 an der Stelle, an der in u_2 der mit der Strukturvariablen & markierte Knoten steht. Eine solche Verwendung der Regel ist u.a. dann erforderlich, wenn ein Baum oberhalb seines Wurzelknotens erweitert werden soll, d.h. wenn eine "Aufpfropfung" formuliert werden soll.

Beispiel:

Abb. 3.6 illustriert die Anwendung der folgenden Regel auf einen Baum u mit dem Resultat v:

Regel 5: (ER.LIT (A) (KONJ B &))

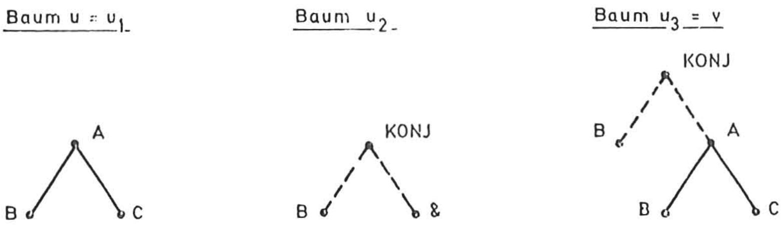


Abbildung 3.6: Anwendung der Regel 5

Def.: (einfache Ersetzungsregel) (Ü21)

Die Regeln

"Ersetze Symbol (durch Symbol)" (ER.S),
"Ersetze Struktur (durch Struktur)" (ER.ST),
"Ersetze Struktur durch Literal" (ER.LIT)

sind einfache Ersetzungsregeln.

Def.: (bedingte Ersetzungsregel) (Ü22)

Ist r eine einfache Ersetzungsregel, sb eine Strukturbedingung, dann ist das Paar (r,sb), geschrieben (r sb), eine bedingte Ersetzungsregel.

Ein Baum v entsteht aus einem Baum u durch Anwendung einer bedingten Ersetzungsregel (r sb) unter einer Knotenzuordnung ω, gdw.:

(1) u erfüllt die Strukturbedingung sb unter der Knotenzuordnung ω
(2) v ensteht aus u durch Anwendung der Regel r unter der Knotenzuordnung ω
(3) ω ist eine zuordnungsgleiche Knotenzuordnung.

Erläuterung:

Mit Ausnahme der Regeln zur Ersetzung von Symbolen erlauben die übrigen einfachen Ersetzungsregeln zwar eine beliebig detaillierte Beschreibung eines zu ersetzenden Teilbaumes, ausgehend von der Charakterisierung durch einen Teilbaum, der nur aus einem Knoten besteht, bis zur Gesamtdarstellung des Teilbaums. Entsprechendes gilt für das zweite Argument der Regel ER.ST.

In den Fällen, in denen eine solche Gesamtdarstellung den Anwendungsbereich einer Regel zu stark einschränkt, ein weniger in die Tiefe gehender Teilbaum jedoch einen zu großen Anwendungsbereich zulassen würde, kann durch

Strukturbedingungen im Interesse einer Regelökonomie die Zahl der notwendigen Regeln eingeschränkt werden.

Beispiel:

Es sei eine Regel (ER.LIT s_1 s_2) zu schreiben, die festlegt, daß die von s_1 zu beschreibende praepositionale Nominalgruppe (PNG) durch den Baum von s_2 zu ersetzen ist. Die Regel soll allerdings nur anwendbar sein, wenn eine der Praepositionen mit, mittels, durch in der PNG vorkommt. Mit Hilfe der Strukturbedingungen kann die Regel formuliert werden als bedingte Regel

```
((ER.LIT (PNG) s2) (UND
                    (DOM PNG PRAEP)
                    (DOM PRAEP (ODER
                           mit mittels durch)))).
```

Ohne Strukturbedingungen wären drei einfache Ersetzungsregeln erforderlich:

(ER.LIT (PNG (PRAEP mit) NG) s_2)
(ER.LIT (PNG (PRAEP mittels) NG) s_2)
(ER.LIT (PNG (PRAEP durch) NG) s_2).

Die Zahl der erforderlichen Regeln würde noch größer, wenn PNG in einer zugrundegelegten Grammatik gar nicht mit Hilfe der Kategorie NG definiert wird, so daß Teilbäume der in Abb. 3.7 dargestellten Form möglich wären, wobei u in Abb. 3.7 eine beliebige Struktur sei.

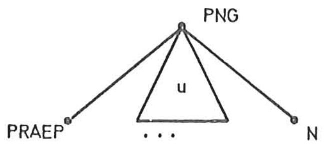


Abbildung 3.7

Ohne die Möglichkeit von Strukturbedingungen müssten für den beschriebenen Fall in Einzelregeln jeweils die für u möglichen Strukturen expliziert werden, es sei denn, es würde eine Baumvariable Y eingeführt. Y müsste dann als Bezeichnung eines obligatorischen oder fakultativen Baumausschnitts interpretiert werden. Insgesamt würde zwar eine solche Baumvariable in Einzelfällen auch eine ökonomischere Darstellung von Strukturbedingungen ermöglichen, würde aber die in 3.5.1 definierten Strukturbedingungen nicht überflüssig machen, wie schon das o.a. Beispiel mit den Praepositionen zeigt. Wir sehen jedoch bei den vielfältigen Formulierungsmöglichkeiten mit Strukturbedingungen keine Notwendigkeit für die Einführung von Baumvariablen, zumal dann die Festlegung der Regelanwendungsbedingungen wesentlich komplizierter würde.

Mit der bisher erläuterten Verwendung von Strukturbedingungen wurde nur die interne Struktur eines zu ersetzenden Teilbaums präzisiert. Eine andere wichtige Möglichkeit ist die Spezifizierung des Kontextes eines Teilbaumes mit Hilfe von Strukturbedingungen. Dies bedeutet, daß bei einer solchen Verwendung von Strukturbedingungen die Regeln zu kontextsensitiven Regeln werden.

<u>Beispiel:</u>

Die bedingte Regel

((ER.LIT (PNG) s_2) (LFT VK PNG))

zur Ersetzung eines Teilbaumes u_1, dessen Wurzelknoten k_1 mit dem Symbol PNG markiert ist, durch den Baum einer Liste s_2 darf nur dann angewendet werden, wenn es unmittelbar links (LFT) von k_1 einen Knoten gibt, der mit dem Symbol VK markiert ist (vgl. Abb. 3.8).

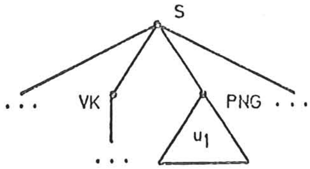


Abbildung 3.8

Def.: (Ersetzungsregel) (Ü23)

Einfache Ersetzungsregeln und bedingte Ersetzungsregeln sind Ersetzungsregeln.

3.5.3 Erweiterungsregeln

Die definierten Ersetzungsregeln ermöglichen die Veränderung von Bäumen lediglich an den Stellen, an denen bereits Teilbäume vorhanden sind, wobei dann diese Teilbäume durch andere Teilbäume ersetzt werden können, deren Knotenmenge größer oder kleiner sein kann als die des zu ersetzenden Teilbaums. Die Erweiterung eines Baumes um einen neuen Zweig wie etwa die Einführung eines Bruderknotens zu einem gegebenen Knoten ist damit nicht möglich. Dazu definieren wir deshalb die Erweiterungsregeln.

Bei den Definitionen von Erweiterungsregeln gehen wir von folgenden Festlegungen aus:

Sei ÜG = (A,F,B,R) eine Übersetzungsgrammatik;
seien a,b $\in$ A Symbole; seien s_1, s_2 Listen über A;
seien u,v $\in$ F Bäume.

Def.: (Regel EW.RSO / EW.LSO - "Erweitere um rechten Sohn / linken Sohn") (Ü24)

Eine Regel "Erweitere um rechten Sohn" (EW.RSO) bzw. "Erweitere um linken Sohn" (EW.LSO) ist ein Paar von Listen, $r = (s_1, s_2)$, geschrieben (EW.RSO s_1 s_2) bzw. (EW.LSO s_1 s_2).

Ein Baum v entsteht aus einem Baum u durch Anwendung der Regel (EW.RSO s_1 s_2) bzw. (EW.LSO s_1 s_2) unter einer Knotenzuordnung ω, gdw.:

es gibt Bäume u_1, u_2, u_3, so daß gilt:
(1) u_1 ist terminaler Teilbaum von u
(2) der vermöge ω der Liste s_1 zugeordnete Baum ist koradikaler Teilbaum von u_1
(3) u_2 ist der nichtindizierte Baum der Liste s_2
(4) u_3 entsteht aus u_1 durch Rechtserweiterung (bei EW.RSO) bzw. durch Linkserweiterung (bei EW.LSO) von u_1 durch u_2
(5) v entsteht aus u durch Ersetzung von u_1 durch u_3.

Beispiel:

Abb. 3.9 zeigt die der Definition entsprechenden Bäume u_1, u_2, u_3 sowie u,v für die folgende Regel:

Regel 6: (EW.RSO B (F G H))

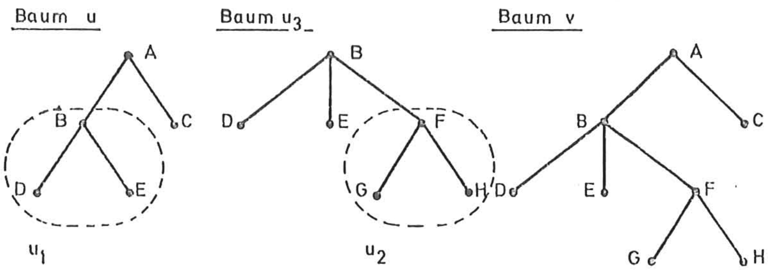


Abbildung 3.9: Anwendung der Regel 6 auf u mit Resultat v

Def.: (Regel EW.RBR / EW.LBR - "Erweitere um rechten Bruder / linken Bruder") (Ü25)

Eine Regel "Erweitere um rechten Bruder" (EW.RBR) bzw. "Erweitere um linken Bruder" (EW.LBR) ist ein Paar von Listen, $r = (s_1, s_2)$, geschrieben (EW.RBR s_1 s_2) bzw. (EW.LBR s_1 s_2).

Ein Baum v entsteht aus einem Baum u durch Anwendung der Regel (EW.RBR s_1 s_2) bzw. (EW.LBR s_1 s_2) unter der Knotenzuordnung ω, gdw.:

es gibt Bäume u_1, u_2, u_3, so daß gilt:

(1) u_1 ist terminaler Teilbaum von u

(2) der vermöge ω der Liste s_1 zugeordnete Baum mit dem Wurzelknoten k ist Teilbaum von u_1

(3) k wird direkt dominiert vom Wurzelknoten von u_1

(4) u_2 ist der nichtindizierte Baum der Liste s_2

(5) u_3 entsteht aus u_1 durch Rechtserweiterung (bei EW.RBR) bzw. durch Linkserweiterung (bei EW.LBR) von u_1 um u_2 neben dem Knoten k

(6) v entsteht aus u durch Ersetzung von u_1 durch u_3.

Beispiel:

Abb. 3.10 zeigt die der Definition entsprechenden Bäume u_1, u_2, u_3 und u,v für die folgende Regel:

Regel 7: (EW.RBR D (F G H))

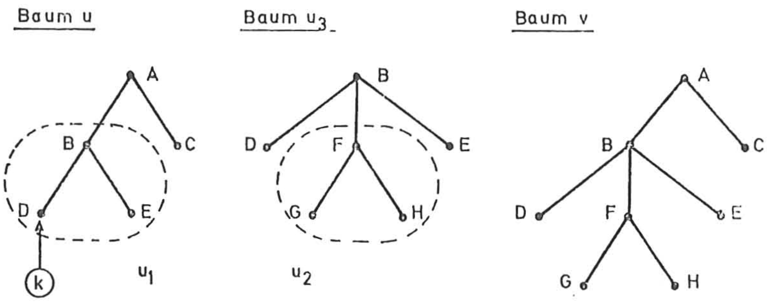


Abbildung 3.10: Anwendung der Regel 7 auf u mit Resultat v

Def.: (einfache Erweiterungsregel) (Ü26)

Die Regeln

"Erweitere um rechten Sohn" (EW.RSO)

"Erweitere um linken Sohn" (EW.LSO)

"Erweitere um rechten Bruder"(EW.RBR)

"Erweitere um linken Bruder" (EW.LBR)

sind einfache Erweiterungsregeln.

Def.: (bedingte Erweiterungsregeln) (Ü27)

Ist r eine einfache Erweiterungsregel und sb eine Strukturbedingung, dann ist das Paar (r,sb), geschrieben (r sb), eine bedingte Erweiterungsregel.

Ein Baum v entsteht aus einem Baum u durch Anwendung einer bedingten Erweiterungsregel (r,sb) unter einer Knotenzuordnung ω, gdw.:

(1) u erfüllt die Strukturbedingung unter der Knotenzuordnung

(2) v entsteht aus u durch Anwendung der Regel r unter der Knotenzuordnung

(3) ω ist eine zuordnungsgleiche Knotenzuordnung.

Def.: (Erweiterungsregel) (Ü28)

Einfache Erweiterungsregeln und bedingte Erweiterungsregeln sind Erweiterungsregeln.

3.5.4 Komplexe Regeln

Die bisher definierten Regeln legen die kleinsten Schritte fest, in denen Bäume durch Umstrukturierung ihrer Teile verändert werden können. Dabei ist die Reihenfolge, in der die Veränderungen vorzunehmen sind, nicht festgelegt mit Ausnahme der Einschränkungen, die sich aus den Anwendungsbedingungen der einzelnen Regeln ergeben. Da die Anwendungsbedingungen jedoch isoliert für einzelne Regeln festgelegt sind, kann in einer einzelnen Regel nicht Bezug genommen werden auf andere Regeln. Genau dies kann jedoch erforderlich sein, wenn eine Strukturveränderung nur in mehreren zusammenhängenden Schritten vollziehbar ist wie etwa, wenn an einem Teilbaum sukzessive mehrere Veränderungen durchzuführen sind. Dazu muß aber durch entsprechende restriktive Strukturbedingungen sichergestellt werden, daß die sukzessiven Regeln auch auf den 'gleichen' Teilbaum angewendet werden, was in der Praxis zu aufwendigen Regeln führen kann. Zusätzlich kann eine Regelordnung (vgl. 3.6) definiert werden, die die Reihenfolge der Anwendung einzelner Regeln vorgibt. Allerdings würde dadurch die Verwendungsmöglichkeit von indizierten Symbolen und Symbolvariablen stark gemindert, da die Knotenzuordnung über der einzelnen Regel definiert ist und damit nicht festgelegt ist, daß bei Regelfolgen aufgrund bestimmter Regelordnungen die entsprechende Folge von Knotenzuordnungen einheitlich sein muß. Wir definieren deshalb unabhängig von einer Regelordnung den Typ einer komplexen Regel, die sich aus mehreren, sukzessiv anzuwendenden Regeln zusammen-setzt. Für die Anwendung einer komplexen Regel wird eine einheitliche Knotenzuordnung zur Bedingung gemacht. Dadurch können in den Einzelregeln einer komplexen Regel gleiche indizierte Symbole oder Symbolvariable zur Bezeichnung eines gleichen Knotens verwendet werden, so daß auch ein Bezug auf Symbole vorangehender Einzelregeln innerhalb einer komplexen Regel möglich ist.

Bei den Definitionen der komplexen Regeln gehen wir von folgenden Festlegungen aus:

> Sei ÜG = (A,F,B,R) eine Übersetzungsgrammatik; seien $u,v \in F$ Bäume; seien $r_1,\ldots,r_n \in R$ Regeln, wobei $n \geq 2$.

Def.: (konjunktive Regel) (Ü29)

Eine konjunktive Regel ist eine Folge von Regeln, $r = (r_1,\ldots,r_n)$, geschrieben (UND $r_1 \ldots r_n$).

Ein Baum v entsteht aus einem Baum u durch Anwendung einer konjunktiven Regel (UND $r_1 \ldots r_n$) unter einer Knotenzuordnung ω, gdw.:

> es gibt Bäume $u_1,\ldots,u_n \in F$, so daß gilt:
> $\forall i,\ i \leq (n-1)$
> (1) $u_1 = u$
> (2) $u_n = v$
> (3) u_{i+1} entsteht aus u_i durch Anwendung der Regel r_i unter der Knotenzuordnung ω_i
> (4) $\omega = \omega_1 \circ \ldots \circ \omega_n$
> (5) ω ist eine einheitliche Knotenzuordnung.

Def.: (disjunktive Regel) (Ü29a)

Eine disjunktive Regel ist eine Folge von Regeln,

$r = (r_1, \ldots, r_n)$, geschrieben (ODER $r_1 \ldots r_n$).

Ein Baum v <u>entsteht</u> <u>aus</u> einem Baum u <u>durch</u> <u>Anwendung</u> einer disjunktiven Regel (ODER $r_1 \ldots r_n$) unter einer Knotenzuordnung ω, gdw.:

> es gibt ein i, $i \in n$, so daß gilt:
> v entsteht aus u durch Anwendung der Regel r_i unter der Knotenzuordnung ω.

<u>Def.:</u> (komplexe Regeln) (Ü30)

Konjunktive Regeln und disjunktive Regeln sind <u>komplexe</u> <u>Regeln</u>.

3.6 Regelordnung

Bisher gingen wir ohne nähere Diskussion davon aus, daß es nicht erforderlich ist, für die Regeln einer Übersetzungsgrammatik eine Ordnungsvorschrift Ω anzugeben, die festlegt, in welcher Reihenfolge die Regeln anzuwenden sind. Wir haben deshalb eine Übersetzungsgrammatik als ein Viertupel ÜG = (A,F,B,R) definiert und eine Ordnungsvorschrift nicht berücksichtigt. Dies würde jedoch eine grundlegende Einschränkung der Generalität unserer Überlegungen bedeuten. Wir führen deshalb die Unterscheidung ein zwischen einer Übersetzungsgrammatik <u>ohne</u> Regelordnung und einer Übersetzungsgrammatik <u>mit</u> Regelordnung, wobei letztere zu definieren ist als Fünftupel $ÜG_\Omega$ = (A,F,B,R,Ω). Eine Übersetzungsgrammatik <u>ohne</u> Regelordnung bezeichnen wir auch weiterhin abgekürzt als Übersetzungsgrammatik, sofern im entsprechenden Kontext kein Mißverständnis möglich oder eine Unterscheidung ohne Belang ist.

Bei den folgenden Definitionen zur Regelordnung gehen wir aus von GINSBURG/PARTEE (1969), pp 315, sowie von KRATZER et al. (1973) und BRESZTOWSZKI et al. (1974). Da wir bei unseren weiteren Überlegungen Regelordnungen nicht näher berücksichtigen werden, beschränken wir uns in diesem Abschnitt auf knappe Erläuterungen und verweisen für ausführlichere Diskussionen und Illustrationen auf die o.a. Autoren.

<u>Def.:</u> (Regelordnung) (Ü31)

Sei R die Regelmenge einer Übersetzungsgrammatik mit Regelordnung; sei STOP $\in$ R ein ausgezeichnetes Symbol.

Eine <u>Regelordnung</u> <u>bezüglich</u> R ist eine Halbordnung

$N = (R', \leq)$ wobei gilt:

(1) $R' = R \cup \{STOP\}$

(2) STOP kommt nur als maximales Element (vgl. unten!) vor.

<u>Erläuterung:</u>

Die Regelordnung N besteht aus der Menge R' und der Relation $\leq$, wobei $\leq \subseteq R \times R'$. D.h. wenn ein Paar (r_i, r_j) in $\leq$ liegt, können wir auch sagen: "r_i ist kleiner als r_j" oder "r_i liegt vor r_j" und es gilt:

(1) wenn $(r_i, r_j) \in \leq$ und $(r_j, r_k) \in \leq$, dann gilt auch $(r_i, r_k) \in \leq$; d.h. $\leq$ ist <u>transitiv</u>;

(2) wenn $r_i \in R'$, dann gilt $(r_i, r_i) \in \leq$; d.h. $\leq$ ist <u>reflexiv</u>;

(3) wenn $(r_i, r_j) \in \leq$ und $(r_j, r_i) \in \leq$, dann gilt $r_i = r_j$; d.h. $\leq$ ist <u>antisymmetrisch</u>;

(4) Es gibt kein r_j, so daß $(STOP, r_j) \in \leq$; d.h. STOP ist <u>maximal</u>.

<u>Def.:</u> (Ordnungsvorschrift) (Ü32)

Sei R die Menge der Regeln einer Übersetzungsgrammatik $ÜG_\Omega$ mit Regelordnung.

<u>Die Ordnungsvorschrift von $ÜG_\Omega$</u> ist ein Viertupel

$\Omega = (K, M, z_o, \delta)$, wobei gilt:

(1) K ist eine Menge von Zuständen z

(2) M ist eine Menge von Regelordnungen bezüglich R mit $M = \{N_z \mid z \in K\}$

(3) $z_o \in K$ ist ein ausgezeichnetes Element von K, der Startzustand

(4) δ ist eine Abbildung $\delta: K \times R \longrightarrow K \setminus \{z_o\}$

Erläuterung:

Mit Hilfe der "Übergangsfunktion" δ werden die Zustände in Zusammenhang gebracht mit der Anwendung von Regeln: d.h. ein Ausdruck $\delta(z_i, r) = z_j$ kann folgendermaßen interpretiert werden: wird in einem Zustand z_i (einer Ableitung) die Regel r angewendet, dann ist nach der Regelanwendung der Zustand z_j erreicht. Der Startzustand z_o ist dann derjenige Zustand, in dem auf Bäume der Basis von $ÜG_\Omega$ eine Regel angewendet werden soll. Ein Element N_z aus M ist die Regelordnung, die für den Zustand z gilt.

Als Konsequenz einer Ordnungsvorschrift ergeben sich für den Ableitungsbegriff einer Übersetzungsgrammatik folgende Einschränkungen:

Def.: (Ü33)

Sei $ÜG_\Omega = (A,F,B,R,\Omega)$ eine Übersetzungsgrammatik mit Regelordnung; sei $\Omega = (K,M,z_o,\delta)$;

ein Baum $u \in F$ ist im Zustand $z \in K$ (vermöge $ÜG_\Omega$) direkt überführbar in v mittels der Regel $r \in R$, gdw.:

(1) v ensteht aus u durch Anwendung der Regel r
(2) r kommt vor in $N_z \in M$
(3) es gibt keine Regel $r' \neq r$, so daß gilt:
  (3a) r' kommt vor in N_z
  (3b) $r' \leq r$
  (3c) es gibt einen Baum $v' \in F$, so daß gilt:
    v' ensteht aus u durch Anwendung der Regel r'.

Eine Folge $(u_1,\dots,u_n)$, $u_i \in F$, ist eine Ableitung (in $ÜG_\Omega$), gdw.:

Es gibt Zustände $z_1,\ldots,z_n$ und Regeln $r_1,\ldots,r_n$, so daß für alle i, $i \leq (n-1)$, gilt:

(1) $u_1 \in B$

(2) $z_1 = z_o$

(3) u_i ist im Zustand z_i direkt überführbar in u_{i+1} mittels r_i

(4) $\delta(z_i,r_i) = z_{i+1}$

Ist darüberhinaus STOP in N_{z_n} und gibt es kein Paar $(r,\text{STOP}) \in N_{z_n}$, so daß u_n im Zustand z_n in einen Baum u_{n+1} direkt überführbar ist mittels der Regel r, dann ist $(u_1,\ldots,u_n)$ eine abbrechende Ableitung.

Die generelle Frage nach der Notwendigkeit von Ordnungsvorschriften für eine Übersetzungsgrammatik müssen wir offenlassen, da nach dem derzeitigen Stand unserer Überlegungen unklar ist, welche Kriterien für die Beantwortung dieser Frage zusammenspielen. Die Fälle, in denen eine Ordnungsvorschrift erforderlich ist, dürften insofern gering sein, als es durch Strukturbedingungen und konjunktive Regeln möglich ist, eine implizite Regelordnung festzulegen im Gegensatz zur expliziten Regelordnung einer Ordnungsvorschrift.

4. ILLUSTRATIVES FRAGMENT EINER ÜBERSETZUNGSGRAMMATIK DEUTSCH/KS (ÜG/KS)

Die im vorangehenden Kapitel beschriebene Übersetzungsgrammatik enthält kaum Hinweise darauf, wie der definierte Formalismus auszufüllen ist, um den Zusammenhang zwischen konkreten Formulierungen einer Ausgangssprache und einer Zielsprache zu beschreiben. Wir geben deshalb in der Folge eine fragmentarische Übersetzungsgrammatik für Ausschnitte des Deutschen und der Konstruktsprache KS an, wobei diesem Fragment ausschließlich exemplarischer Charakter zukommt. Dies bedeutet, daß wir weder für angeschnittene syntaktische Probleme des Deutschen noch für Probleme auf der Ebene der Konstruktsprache irgendwelche Ansprüche auf Vollständigkeit oder allgemeine Gültigkeit stellen. Wir verstehen die nachfolgenden Überlegungen vielmehr als eine erste Überprüfung der Tragfähigkeit und Praktikabilität der vorgeschlagenen Form einer Übersetzungsgrammatik und müssen es zukünftigen Anstrengungen überlassen, die einzelnen angesprochenen Probleme unter Einbeziehung der Ergebnisse syntaktischer und semantischer Forschung gründlich zu untersuchen.

Die Beispiele wählen wir aus dem Anwendungsbereich des Systems PLIDIS, der Industrieabwasserkontrolle. Konstituierend für diesen Bereich sind Firmen, deren Betriebe Abwasser erzeugen und die Erlaubnis haben, unter Berücksichtigung der gesetzlichen Grenzwerte für Schadstoffe, dieses Abwasser in öffentliche Gewässer einzuleiten. Zur Kontrolle ist die Wasserwirtschaftsverwaltung gehalten, bei den Betrieben eine bestimmte Zahl von Abwasserproben zu entnehmen und sie auf ihren Schadstoffgehalt überprüfen zu lassen. Das

Ergebnis der Laboruntersuchung wird in einem Laborbericht festgehalten, der Grundlage für nachfolgende Maßnahmen der Behörden ist. In PLIDIS sind die Daten über die entnommenen Proben sowie die dazugehörigen Laborberichte gespeichert, ebenso die Normen für die erlaubten Grenzwerte der Schadstoffkonzentrationen. Dem Benutzer von PLIDIS soll es möglich sein, mittels Fragen, die in deutscher Sprache formuliert sind, Information über Sachverhalte aus dem umrissenen Anwendungsbereich zu erhalten. Dies mag als Erklärung ausreichen, warum sich die folgenden Beispiele um "Proben", "Chemikalien" und "Laborberichte" drehen (zum Gesamtsystem PLIDIS vgl. BERRY-ROGGHE et al. (1978) bzw. KOLVENBACH et al.(1978)).

Terminologische Festlegung

Die Übersetzungsgrammatik deutsch/KS, die wir in der Folge fragmentarisch erläutern, ist gem. Abschnitt 3.1 definiert als

ÜG/KS = (A,F,B,R).

Da in der Folge ausschließlich von den Komponenten von ÜG/KS die Rede ist, sagen wir nicht jedesmal, wenn wir etwa von B oder R sprechen, "die Basis von ÜG/KS" oder "die Regeln von ÜG/KS", sondern nur kurz "die Basis" oder "die Regeln". Gleiches gilt für das Alphabet A und die Baummenge F sowie die Teilmengen von A, wobei wir zur Bezeichnung der Vokabulare von ÜG/KS die in 3.2 eingeführten Symbole verwenden.

4.1 Die Zielsprache KS

Bei der Beschreibung der Konstruktsprache KS halten wir uns im wesentlichen an ZIFONUN (1978) und DILGER/ZIFONUN (1978), wobei wir in eigener Verantwortung Vereinfachungen vornehmen, die das Bild von KS an manchen Stellen verfälschen mögen, die uns im gegebenen Kontext jedoch ohne Belang erscheinen. Allerdings sind solche Verfälschungen von Bedeutung, wenn man sich allein aufgrund unserer Darstellung ein Urteil über die Eignung von KS als Semantiksprache für Ausschnitte des Deutschen bilden will.

Die S y n t a x von KS wird mit der Regelgrammatik

$$G_{ZS} = (VN_{ZS}, VT_{ZS}, S_{ZS}, R_{ZS})$$

angegeben, wobei S_{ZS} kein ausgezeichnetes Symbol aus VN_{ZS}, sondern die Menge

$$S_{ZS} = \{\text{FORMEL, LAMBDATERM/Sorte=s}\}$$

mit s $\in$ SRT, die Menge der Sorten. Wir verzichten hier einfachheitshalber auf die Aufzählung der Mengen VN_{ZS} und VT_{ZS}, da diese aus den Regeln ersichtlich sind.

Die für den in PLIDIS verwendeten Ausschnitt von KS (vgl. BERRY-ROGGHE et al. (1978)) festgelegten Sorten mit ihrer Hierarchie sind in Tab. 4.1 abgebildet. Die Sortenhierarchie ist beschreibbar durch eine über SRT definierte Relation $\leq$ (zu den Eigenschaften von $\leq$ vgl. Def. Ü32, Ordnungsvorschrift). Sind s_1 und s_2 Sorten, und gilt $(s_1,s_2) \in \leq$, geschrieben $s_1 \leq s_2$, dann sagen wir auch: s_1 ist eine Untersorte von s_2.

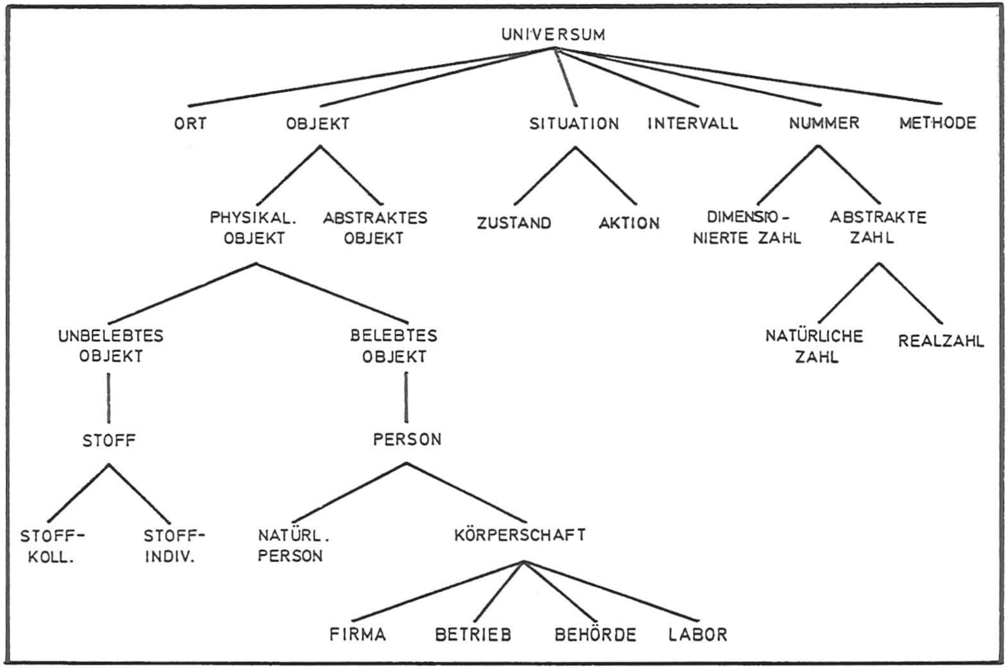


Tabelle 4.1: Sorten und Sortenhierarchie des verwendeten KS-Ausschnitts "Industrieabwasserüberwachung"

Zur Darstellung der Sytax mit komplexer Notation verwenden wir die Attribute 'Sorttup' (für Sortentupel) und 'Sorte', wobei 'Sorttup' als Wert eine Folge von Sorten und 'Sorte' als Wert eine einzige Sorte hat.

Die Regelmenge R_{ZS} der Grammatik von KS ist in Tab. 4.2 aufgelistet, wobei wir aus technischen Gründen für ein Paar $(a,\alpha) \in R_{ZS}$ nicht $a \longrightarrow \alpha$ sondern $a ::= \alpha$ schreiben. Mehrere Regeln der Form

$$a ::= \alpha_1$$
$$a ::= \alpha_2$$
$$a ::= \alpha_3$$
$$a ::= \ldots$$

fassen wir zusammen als

$$a ::= \alpha_1 \mid \alpha_2 \mid \alpha_3 \mid \ldots$$

Tab. 4.2 enthält an verschienen Stellen Regelschemata statt Regeln. Für die Herleitung der Einzelregeln aus den jeweiligen Regelschemata gilt, soweit die einzeln aufgeführten Restriktionen nichts anderes bestimmen, generell:

- die Variablen $s, s_1, \ldots, s_n, s_1', \ldots, s_n'$ sind durch beliebige Sorten zu ersetzen;
- das als Indexvariable als auch als Objektvariable verwendete Symbol n ist zu ersetzen durch natürliche Zahlen.

Genaugenommen müßten die Bestimmungen aufgrund der Regeln R12, R13 und R15 eingeschränkt werden, da sonst Mengen von Regeln entstehen würden, bei deren Verwendung in einer Ableitung Wörter entstehen würden, die nicht zu Wörtern aus VT_{ZS}^* abgeleitet werden können. Da es uns hier jedoch nicht um eine formal stringente Regelgrammatik geht, verfolgen wir die erforderlichen Einschränkungen nicht weiter.

```
(R1)   FORMEL ::= AFOR|
                  QUANTORFOR|
                  (Nicht FORMEL)|
                  (Und FORMEL FORMEL)|
                  (Oder FORMEL FORMEL)|
                  (Implik FORMEL FORMEL)|
                  (Aequiv FORMEL FORMEL)|
                  (? FORMEL)|
                  (! FORMEL)

(R2)   QUANTORFOR ::= (Exist KSVAR/Sorte=s FORMEL)|
                      (Fürall KSVAR/Sorte=s FORMEL)

(R3)   AFOR ::=   RELFOR|
                  OPFOR

(R4)   RELFOR ::= (KSREL/Sorttup=(s_1,...,s_n)
                  TERM_1/Sorte=s_1' ... TERM_n/Sorte=s_n')

                  wobei n ∈ ℕ und s_i' = s_i. Dies gilt
                  auch für alle nachfolgenden Regeln!

(R5)   OPFOR ::=  (KSOP/Sorttup=(s_1,...,s_n) TERM_1/
                  Sorte=s_1' ... TERM_n/Sorte=s_n')

(R6)   TERM/Sorte=s ::= KSVAR/Sorte=s|
                        INDIVKONST/Sorte=s|
                        RELTERM/Sorte=s|
                        OPTERM/Sorte=s|
                        QUANTTERM/Sorte=s|
                        LAMBDATERM/Sorte=s|
                        (Liste TERMLISTE/Sorte=s)|
                        (Jota TERM/Sorte=s)
```

Tabelle 4.2: Regeln der KS-Grammatik

(R7)	RELTERM/Sorte=s ::=	(KSREL/Sorttup=($s_1, \ldots, s_n$, s) TERM$_1$/Sorte=s_1' ... TERM$_n$/Sorte=s_n')
(R8)	OPTERM/Sorte=s ::=	(KSOP/Sorttup=($s_1, \ldots, s_n$, s) TERM$_1$/Sorte=s_1' ... TERM$_n$/Sorte=s_n')
(R9)	QUANTTERM/Sorte=s ::=	(KSQUANT TERM/Sorte=s)
(R10)	LAMBDATERM/Sorte=s ::=	(Lambda KSVAR/sorte=s FORMEL)
(R11)	TERMLISTE/Sorte=s ::=	TERM/Sorte=s' | TERM/Sorte=s' | (mit s' $\leq$ s) TERMLISTE/Sorte=s
(R12)	KSREL/Sorttup=(ort, ort) ::= IN-LOK | ... (weitere Regeln für KSREL sind aus Tab. 4.3 herleitbar)	
(R13)	KSOP/Sorttup=(betrieb, int, stoffkoll) ::= PROBE | ... (weitere Regeln für KSOP sind aus Tab. 4.3 herleitbar)	
(R14)	KSVAR/Sorte=s ::=	x.s | x_n.s | ... wobei n $\in \mathbb{N}$
(R15)	INDIVKONST/Sorte=firma ::= Lauxmann|Joos | ... (weitere Regeln für INDIVKONST sind aus Tab. 4.3 herleitbar)	
(R16)	KSQUANT ::=	irgendein | all | 1 | 2 | ... | n | QUANTMOD n wobei n $\in \mathbb{N}$
(R17)	QUANTMOD ::=	genau | mindestens | höchstens | wenigerals | mehrals

Tabelle 4.2: (Fortsetzung)

Individuenkonstanten	Sorte
ARSEN	stoff
BRECHT	natper
G-LAUXMANN	firma
PLOCHINGEN	ort
75/10/13/12/00	int
Ø.35-MG/L	dimzahl

Relationszeichen	Sortentupel
ADRESSE	(pers, ort)
IN-LOK	(ort, ort)
IN-TEMP	(int, int)
INTER	(int, int, int)
LETZTNVOR	(natzahl s_1, s_2, s_1)
NACHGEWIESEN	(stoffkoll, methode, abstrobj, stoff)
VOR	(int, int)

Operationszeichen	Sortentupel
ANTEIL	(stoff, stoffk., abstrobj, dimzahl)
ANZAHL	(uni, natzahl)
BETRIEB	(firma, ort, betrieb)
GRENZWERTVERLETZUNG	(stoff, sit)
GROESSER	(num, num, sit)
LABORBERICHT	(labor, stoffk., int, abstrobj)
MAHNUNG	(stoffk, sit)
O-GRENZE	(num, num)
PH-WERT	(stoffk., methode, abstrobj, abstrzahl)
PROBE	(betrieb, int, stoffk.)
PROBE-ENTNAHMEBERICHT	(stoffk., abstrobj)
PROBENEHMER	(stoffk., per)
TEMPERATUR	(stoffk., int, abstrobj, dimzahl)
U-GRENZE	(num, num)
VERWARNUNG	(stoffk, sit)

Tabelle 4.3: Ausschnitt aus dem terminalen Vokabular von KS für den Bereich "Industrieabwasserkontrolle"

Erläuterung zu Tab. 4.2

Zentrale syntaktische Einheiten von KS sind Operationszeichen (KSOP) und Relationszeichen (KSREL), mit denen atomare Formeln (AFOR) und Terme gebildet werden. Sofern es auf die noch zu erläuternde Unterscheidung zwischen Operations- und Relationszeichen nicht ankommt, verwenden wir die Bezeichnung 'Prädikatszeichen' oder kurz 'Prädikat'.

Eine Besonderheit von KS gegenüber klassischen prädikatenlogischen Sprachen liegt in der Verwendung von Prädikaten zur Bildung von Termen, die an folgendem Beispiel erklärt sei:

Beispiel:

Es seien folgende Regeln gegeben:

```
KSOP/Sorttup=(pers pers)::=  VATER
KSREL/Sorttup=(pers pers)::= FREUND
INDIVKONST/Sorte=pers    ::= FRITZ|HANS|KLAUS|PETER
```

Aufgrund dieser Regeln sowie der Regeln von Tab. 4.2 können die atomaren Formeln (vgl. R4,R5,R6)

(VATER FRITZ HANS)   "Der Vater von Fritz ist Hans"

(FREUND FRITZ KLAUS) "Fritz und Klaus sind Freunde"

(FREUND FRITZ PETER)

(FREUND FRITZ (LISTE KLAUS PETER))

"Klaus und Peter sind Freunde von Fritz" sowie die Terme der Sorte 'pers' (vgl. R6 - R9)

(VATER FRITZ)	"der Vater von Fritz"
(FREUND FRITZ)	"der/die Freund/Freunde von Fritz"
(IRGENDEIN (FREUND FRITZ))	"Freunde von Fritz" "irgendein (mindestens ein) Freund von Fritz"
(ALL (FREUND FRITZ))	"alle Freunde von Fritz"

gebildet werden, deren Bedeutung wir in Anführungszeichen paraphrasiert haben. D.h. also, n-stellige Prädikate können (n-1)-stellig gebraucht zur Bezeichnung von Individuen (z.B. VATER) oder von Elementen von Mengen (z.B. BRUDER) verwendet werden (vgl. R7, R8).

Verwendet man x.pers als Variable der Sorte 'pers' (vgl. R14), dann kann man mit Hilfe des λ-Operators LAMBDA die skizzierte Termbildung als Kurzschreibweise für ein Lambdaabstrakt (LAMBDATERM, vgl. R10) erklären, so daß gilt:

(VATER FRITZ) ≡ (LAMBDA x.pers (VATER FRITZ x.pers))

Lambdaterme werden auch verwendet, wenn eine Termbildung durch (n-1)-stellige Verwendung eines Prädikats nicht möglich ist wie etwa bei

(LAMBDA x.pers (VATER x.pers HANS))

"diejenigen, deren Vater Hans ist" i.e.
"die Söhne von Hans".

Die Unterscheidung von Relationstermen und Operationstermen wurde von ZIFONUN eingeführt zur Differenzierung zwischen Individuentermen, die ein Individuum bezeichnen (OPTERM), und Listentermen die Elemente einer Menge aufzählen (RELTERM). Da diese Unterscheidung eine nicht unwesentliche Erweiterung der KS-Regelmenge bedingen würde und auch für die nachfolgenden Abschnitte von untergeordneter Bedeutung ist, haben wir diese Differenzierung in Tab. 4.2 nicht weiter berücksichtigt.

Für die Darstellung von Befehlen und Entscheidungsfragen verfügt KS über die pragmatischen Operatoren '?' und '!' (vgl. R1), während Ergänzungsfragen dargestellt werden durch einen selbständigen LAMBDATERM (vgl. R10), d.h. also durch einen TERM, der nicht Argument eines Prädikats ist. Von daher ist auch die Festlegung von zwei Startsymbolen, FORMEL und LAMBDATERM, in der Regelgrammatik von KS begründet.

4.2 Die Basis von ÜG/KS

Die Möglichkeit, die Basis einer Übersetzungsgrammatik mit Hilfe einer Regelgrammatik festzulegen haben wir bereits erläutert. In Abhängigkeit von der Verwendung der Regeln von ÜG/KS zur Steuerung derjenigen Komponente im System PLIDIS, die aus deutschen Sätzen konstruktsprachliche Formeln erzeugen soll, schlagen wir hier einen anderen Weg ein. Dazu gehen wir aus von einem Korpus K, $K \subset VT^{+}_{AS}$, und einer Funktion π, die wir Parsingfunktion, d.h. eine syntaktische Analysefunktion nennen.

Def.: (Parsingfunktion)

Sei $V_{AS} = VN_{AS} \cup VT_{AS}$ das ausgangssprachliche Vokabular von ÜG/KS, F die Baummenge von ÜG/KS; sei $n \in \mathbb{N}$.

Eine Parsingfunktion ist eine Abbildung

$$\pi : VT_{AS}^{+} \longrightarrow \mathbb{P}(F)$$

$$\pi(\alpha) = \{ t_1, \ldots, t_n \mid \text{(1) } t_i, i \leq n, \text{ ist ein Ableitungsbaum der Regelgrammatik von } G_{AS} \quad \text{(2) } \alpha \text{ ist das Wort von } t_i \}$$

Die Basis von ÜG/KS ist dann angebbar als die Menge

$$B = \{ t \mid \text{es gibt ein } \alpha \in VT_{AS}^{+}, \text{ so daß } t \in \pi(\alpha) \}$$

D.h. also, daß die verwendete Regelgrammatik des Deutschen nicht explizit in der Definition von B vorkommt, sondern in die Definition der Parsingfunktion eingegangen ist.

Der Algorithmus, der die unseren Beispielen zugrundegelegte Parsingfunktion π berechnet, sowie die entsprechende Grammatik ist beschrieben in LÖTSCHER (1978) und LÖTSCHER/KOLVENBACH (1978), eine in alle Einzelheiten gehende Darstellung findet sich in LÖTSCHER (1977). Die dort für die syntaktische Analyse verwendete Grammatik ist in "Augmented Transition Networks (ATN)" (vgl. WOODS (1970)) formuliert. Da die Erläuterung des ATN-Formalismus über den Rahmen dieser Arbeit hinausgehen würde, geben wir in Tab. 4.6 hilfsweise den für unsere Beispiele relevanten Teil der Phrasenstrukturgrammatik an, mit der LÖTSCHER (1977) die Ergebnisstrukturen des Parsingalgorithmus skizziert. D.h. die von dem Algorithmus "erzeugten" Baumstrukturen sind eine Teilmenge der Ableitungsbäume der in Tab. 4.6 angegebenen Grammatik. Zur Exaktheit dieser Grammatik bemerkt LÖTSCHER:

> "Die hier angegebene Grammatik kann nicht Anspruch auf völlige Exaktheit machen, da ja nicht nur die Operationen zur Ergebniserzeugung, sondern auch die Netze (d.h. die ATN, d.Verf.) selbst berücksichtigt werden müssen. Wollte man alle Bedingungen berücksichtigen, würde die Grammatik zu kompliziert." (LÖTSCHER (1977), p. 84)

Aus darstellungstechnischen Gründen werden die Symbole in Tab. 4.6 nicht komplex notiert. Die verwendeten Symbole sind jedoch in Tab. 4.4 und 4.5 mit ihren möglichen Merkmalen und Werten aufgelistet.

Zur Charakterisierung der vom Analysealgorithmus erzeugten Strukturbeschreibungen schreibt LÖTSCHER (1977), pp. 79 f.:

- Die einzelnen Wörter werden nicht in ihrer voll flektierten Form, sondern in ihrer Normalform angegeben. Die morphologischen Spezifikationen werden notwendigenfalls als Merkmalsspezifikationen angegeben.

- Morphologische und morphosyntaktische Spezifikationen werden dort angegeben, wo sie aus logischen Gründen (strukturellen oder semantischen) Gründen zur Weiterverarbeitung benötigt werden, nicht unbedingt dort, wo sie gefunden worden sind. Besonders wenn sie an mehreren zusammenwirkenden Stellen zugleich erschlossen oder angetroffen worden sind, sollen sie so "hoch oben" in der Ergebnisstruktur wie möglich vermerkt werden, da sie so am schnellsten aufgefunden werden können.

Auf konkrete Fälle bezogen heißt das, daß z.B. die Passivierung nicht als Merkmal des Verbs, sondern des ganzen Satzes angegeben wird; denn sie betrifft ja auch die Kasusbedingungen für die Ergänzungen...

- Wenn analytische Konstruktionen (aus mehreren Wörtern) funktional einzelnen flektierten Wörtern bezüglich einer morphologischen Kategorie äquivalent sind, so sind solche Konstruktionen wenn möglich durch entsprechende einfache Angaben (z.B. einfache Normalform) anzugeben. Die zusammengesetzten Zeiten eines Verbs können somit durch die Normalform des Verbs mit der entsprechenden Tempusangabe dargestellt werden.

- Umgekehrt sind Konstruktionen, bei denen ein Wort funktional eine mehrwortige Konstruktion vertritt, auch durch mehrwortige Ausdrücke wiederzugeben. <u>am</u>, <u>beim</u> usw. in ihrer Funktion als Zusammenziehung von Präposition und Artikel (nicht aber bei Superlativen wie am schönsten) sind also explizit als Präposition + Artikel darzustellen.

- Diskontinuierliche Konstituenten (Konstituenten, deren Teile durch eine womöglich variable Menge gleich- oder übergeordneter Konstituenten voneinander getrennt sind) sind im Ergebnis möglichst als eine Konstituente darzustellen. Dabei sollte die Position der Teile im Gesamtsatz aber erkennbar bleiben. Dieses Problem stellt sich im Deutschen für mehrgliedrige Verbalkomplexe im Hauptsatz. Es wird hier so gelöst, daß die gesamte Darstellung des Verbalkomplexes am Satzanfang erscheint, und daß das Vorkommen der natürlich-sprachlichen Formen unter dem sonst nirgend verwendeten Kategoriensymbol VERB angegeben wird.

Kategorie	Kurzcharakterisierung	Merkmal	mögliche Werte des Merkmals
S	Hauptsatz	TYPE DIATHESE	AUSSAGE, FRAGE, BEFEHL AKTIV, PASSIV
NG	Nominalgruppe	K (Kasus) PN (Person-Numerus) G (Genus)	NOM, GEN, DAT, AKK 1, 2, 3, 4, 5, 6 M, F, N
PNG	Präpositional-Nominalgruppe	- wie NG -	
ADJG	Adjektivgruppe (Adjektiv mit vorangehenden Adverbien in NG und PNG)	- wie NG -	
ZAHL	Zahl mit vorangehenden Adverbien (z.B. <u>fast</u> 8, <u>etwa</u> 23)	-	-
ADJUG	Adjektivgruppe mit unflektiertem Adjektiv	-	-
VK	Hauptsatzverbkomplex	PN (Person-Numerus) MODUS	1, 2, 3, 4, 5, 6 IND, KONJ, BEF
FG	freie Gruppe (Partikeln wie <u>nicht</u>, <u>nur</u>, <u>auch</u>, <u>als</u>, <u>wie</u>)	-	-

<u>Tabelle 4.4:</u> Liste der nichtterminalen Symbole ohne Wortklassenkategorien (nach LÖTSCHER (1977) pp 82)

Kategorie	Kurzcharakterisierung	Notierte Normalform	Merkmale und mögliche Werte
N	Nomen	Nominativ Singular oder Plural (plurale tantum)	K = NOM, GEN, DAT, AKK PN = 3, 6 G = M, F, N
NPR	Eigenname	- wie bei N -	- wie bei N -
MASSN	Maßnomen	normierte Abkürzung	- wie bei N -
VERB	-	Infinitiv	-
V	Verb	Infinitiv	TEMP=IN (Infinitiv) TEMP=IZ (Infinitiv mit eingeschlossenem "-zu-") TEMP=GE (Gegenwart) TEMP=VE (Vergangenheit) TEMP=P1 (Partizip Präs.) TEMP=P2 (Partizip Perf.) PN = 1, 2, 3, 4, 5, 6 MODUS=IND, KONJ, BEF DIATHESE=AKTIV
ADJ	Adjektiv	unflektierte Form	K, PN, G wie bei N S = POS, KOMP, SUP (Steigerungsstufe)
ADJU	unflektiertes Adjektiv	unflektierte Form	S = POS
ADV	Adverb	Wortform	--
WADV	Frage-Adverb	Wortform	--

Tabelle 4.5: Liste der Wortklassenkategorien (nach (LÖTSCHER/KOLVENBACH (1978), Kap. 2.2.5)

Kategorie	Kurzcharakterisierung	Notierte Normalform	Merkmale mögliche Werte
PRAEP	Praeposition	Wortform	R = GEN, DAT, AKK (Rektion)
DET	Determinans	Nominativ Singular	- wie bei N -
QDET	quantifizierendes Determinans	Nominativ Singular	- wie bei N -
WDET	Fragedeterminans	Lexem ohne Endung	- wie bei N -
NEGDET	negierendes Determinans	Lexem ohne Endung	- wie bei N -
PRON	Pronomen	Nominativ	K = NOM, GEN, DAT, AKK PN = 1, 2, 3, 4, 5, 6 G = M, F, N
REFLPRON	Reflexiopronomen		- wie bei N -
WPRON	Fragepronomen		- wie bei N -
INTEGER ZAHL	ganze Zahl	Ziffer	
REALZAHL	rationale Zahl	Zifferndarstellung	
ORDZAHL	Ordnungszahl	Ziffer	- wie bei N -
DATUMS ZAHL	z.B. 1976 15.11.76	z.B. 1976/ 76/11/15	

Tabelle 4.5: (Fortsetzung)

```
             ┌┌NG ┐*  ┐                      ┌NG ┐*    ┌ADJUG┐┌NG ┐
S ::= VK (FG) (│└PNG┘   │) VERB (REFLPRON)  │PNG│    (│ FG  ││PNG│) (VERB)
             └ ADJUG   ┘                    └FG ┘     └     ┘└   ┘

        ┌  ┌       ┐                  ┌        ┐*  ┌                          ┐ ┐
        │  │DET    │                  │ADJG    │   │ZAHL   MASSN  ADJG*   N  │ │
NG ::=  │ (│QDET   │ (QDET))         (│ORDZAHL │)  │(ZAHL) N                  │ │
        │  │NEGDET │                  └        ┘   │NPR                       │ │
        │  │WDET   │                               └                          ┘ │
        │  └       ┘                                                            │
        │                        DATUMSZAHL                                     │
        │                        PRON                                           │
        │                        NEGPRON                                        │
        └                        WPRON                                          ┘

PNG ::= ┌ PRAEP  ... - wie bei NG - ... ┐
        │ ADV                           │
        └ WADV                          ┘
```

Tabelle 4.6: Angenäherte Phrasenstrukturgrammatik für die vom Parsingalgorithmus erzeugten Strukturen (nach LÖTSCHER (1977), pp 86)

$$\text{ADJG} ::= \left(\begin{bmatrix} \text{PNG} \\ \text{NG} \end{bmatrix}^{*} (\text{ADJU}) \right) \text{ADJ}$$

$$\text{ZAHL} ::= (\text{ADV}) \begin{bmatrix} \text{INTEGERZAHL} \\ \text{REALZAHL} \end{bmatrix}$$

$$\text{ADJUG} ::= \text{ADV}^{*} \ \text{ADJU}$$

$$\text{VK} ::= \text{V}$$

<u>Für die Darstellung der Regeln sind die folgenden Konventionen vereinbart:</u>

(X)	:	fakultative Konstituente
$\begin{bmatrix} \text{X} \\ \text{Y} \end{bmatrix}$	:	alternative Konstituenten
X^{*}	:	Kleene'scher Stern: n-fache Wiederholung von X, $n \geq \emptyset$
$\begin{bmatrix} \text{X} \\ \text{Y} \end{bmatrix}^{*}$	:	beliebige Folge von beliebiger Länge aus den Elementen X und Y

<u>Tabelle 4.6</u>: (Fortsetzung)

Neben der Voranstellung des Verbkomplexes ist auf eine weitere Besonderheit der erzeugten Analysestrukturen hinzuweisen: die schwache Strukturierungskapazität bezüglich der Struktur von Nominalgruppen bzw. praepositionalen Nominalgruppen. Da in den Strukturbäumen NG bzw. PNG immer direkt von S dominiert wird - mit Ausnahme des Nebensatzbereichs und der Koordination, deren Regeln wir hier weggelassen haben -, ist nicht unmittelbar aus den Strukturen entscheidbar, ob eine NG oder PNG in der Beziehung "Attribut" zu einer anderen NG bzw. PNG steht oder in der Beziehung "Ergänzung" zum Verbkomplex des Satzes.

Ein Beispiel für die vom Parsingalgorithmus erzeugte Struktur ist als Baum in Abb. 4.1 dargestellt.

<u>Zur Darstellungstechnik in den Abbildungen</u>

1. Aus zeichentechnischen Gründen verzichten wir in dieser sowie in allen folgenden Abbildungen auf die komplexe Notation der Symbole und lassen die Merkmale und ihre Werte wegfallen.

2. Häufig notieren wir die terminalen AS-Symbole nicht mit der in Tab. 4.5 angegebenen Normalform, sondern zur besseren Verständlichkeit in der flektierten Form.

3. In Strukturen von Beispielsätzen, in denen das Verb an erster Stelle steht, verzichten wir auf die Darstellung der auf VK direkt folgenden Konstituente VERB.

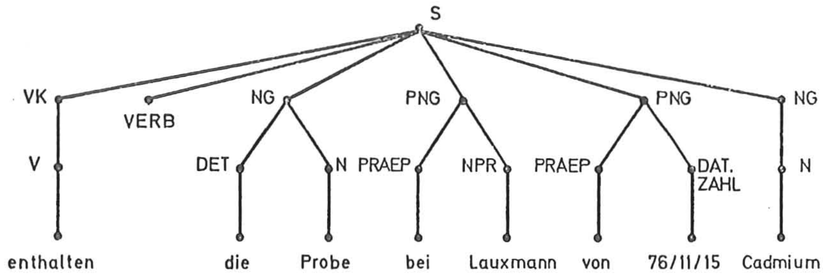


Abbildung 4.1: Baumdarstellung einer vom Parsingalgorithmus erzeugten Struktur (ohne Berücksichtigung der komplexen Notation) zum Satz "Enthielten die Proben bei Lauxmann vom 15.11.76 Cadmium?"

4.3 Konventionen für Knotenzuordnungen und besondere Strukturbedingungen

Bei den Regeln von ÜG/KS spielen die syntaktischen Relationen "Attribut" und "Ergänzung" eine besondere Rolle. Dabei charakterisiert die Relation "Attribut" einen zu präzisierenden strukturellen Zusammenhang zwischen einem Nomen (N) und einer (Präpositionalen) Nominalgruppe (NG bzw. PNG), die Relation "Ergänzung" den Zusammenhang zwischen einem Verb (V) und einer NG bzw. PNG. Wir betrachten die Termini "Attribut" und "Ergänzung" rein mnemotechnisch als Namen für Strukturbedingungen, die wir in der Folge definieren, und gehen deshalb nicht auf terminologische und theoretische Variationen und Kontroversen zu diesen Begriffen ein, die sich in der linguistischen Literatur finden. Da es uns in diesem Abschnitt zudem nur darum geht, den Umgang mit Strukturbedingungen zu illustrieren, sind insbesondere die Festlegungen für die Relation "Attribut" unvollständig und bedürfen verschiedener Erweiterungen unter Berücksichtigung des neueren Forschungsstandes (vgl. z.B. DROOP (1977) und die dort angegebene Literatur).

Bevor wir die entsprechenden Strukturbedingungen festlegen, definieren wir zuerst eine Erweiterung der Möglichkeiten der Knotenzuordnung, die den Umgang mit komplex notierten Symbolen erleichtern soll.

4.3.1 Komplex notierte Symbole und Knotenzuordnung

Der in Abb. 4.1 dargestellte Baum sollte nicht darüber hinwegtäuschen, daß die Knoten der Bäume von ÜG/KS mit komplex notierten Symbolen markiert sind. Bei der Festlegung der Regeln von ÜG/KS kommt es jedoch häufig nicht auf einzelne Merkmale und ihre Werte an. So können beispielsweise für eine Regel über eine Nominalgruppe die Werte der Merkmale 'Genus' und 'Numerus' irrelevant sein, weshalb es wünschenswert ist, lediglich das Symbol NG zu verwenden und für die Knotenzuordnung festzulegen, daß dem Symbol NG ein Knoten zuordenbar ist, der mit einem komplex notierten Symbol markiert ist, dessen Hauptkategorie NG ist.

Für Symbole der Konstruktsprache mit dem Merkmal 'Sorte' entstehen aufgrund der hierarchischen Sortenstruktur zusätzliche Probleme. Wird etwa in einer Regel festgelegt, daß als Argument eines Prädikats ein Term der Sorte x zu verwenden ist, muß aufgrund der Syntax von KS auch zulässig sein, einen Term der Sorte y zu verwenden, wenn y eine Untersorte von x ist. Anstatt für die Terme der entsprechenden Untersorten eigene Regeln festzulegen, wird eine zweite Erweiterung der Knotenzuordnung eingeführt. Diese Erweiterung soll beispielsweise erlauben einem Symbol TERM/Sorte=x einen Knoten zuzuordnen, der mit TERM/Sorte=y markiert ist, wenn y eine Untersorte von x ist.

Zur Definition der Erweiterung nehmen wir Bezug auf Def. Ü8.

Def.:

Einem komplex notierten Symbol $a = A/m_1=w_1,\ldots,m_i=w_i$ mit $i \geqq \emptyset$, ist ein Knoten k in einem Baum zuordenbar, gdw.:

> Es gibt ein komplex notiertes Symbol
> $a' = A/m_1'=w_1',\ldots,m_j'=w_j'$, so daß gilt:
> $(\forall n \leqq i)\ (\exists 1 \leqq j)$
> (1) $m_n = m_1$
> (2) $(m_n = \text{Sorte} \longrightarrow w_n \leqq w_1)$
> (3) k ist a' zuordenbar

4.3.2 Strukturbedingung "Attribut"

Die Strukturbedingung "Attribut" legen wir fest, mit Hilfe der Dominanzbedingung und der mittelbaren Linksbedingung, wobei wir zwei Fälle unterscheiden: den Fall des Praepositionalattributs und den Fall des Genitivattributs.

Strukturbedingungen für ein Praepositionalattribut werden nach dem Schema $sb_{p.attr}$ gebildet, wobei i, j zwei beliebige Indizierungen, $i \neq j$, und N sowie PNG_j für beliebige komplexe Symbole mit der Hauptkategorie N bzw. PNG stehen:

```
sb_p.attr = (UND (EQ X1 (ODER NG PNG_i))          (1)
                 (DOM X1 N )                      (2)
                 (LFT* X1 PNG_j)                  (3)
                 (NICHT                           (4)
                  (EXIST X2                       (5)
                   (UND (EQ X2 (ODER VK VERB))    (6)
                        (LFT* X2 PNG_j)           (7)
                        (LFT* X1 X2))))))         (8)
```

<u>Festlegung</u>

Erfüllt ein Baum eine nach $sb_{p.attr}$ gebildete Strukturbedingung, dann sagen wir auch: PNG_j ist ein (<u>mögliches</u>) (<u>Praepositional</u>-) <u>Attribut</u> <u>von</u> N. Zur Vereinfachung vereinbaren wir:

<u>Schreibkonvention für eine Strukturbedingung "mögliches Praepositonalattribut"</u>

Für eine nach dem Schema $sb_{p.attr}$ gebildete Strukturbedingung sb schreiben wir auch

(M.ATTR N PNG_j)

mit der Festlegung (M.ATTR N PNG_j) = sb.

<u>Erläuterung</u>

Wird ein Nomen N von NG bzw. PNG dominiert (vgl. (1) -(2) oben), dann ist ein mögliches Praepositionalattribut eine PNG_j, die rechts von NG bzw. PNG_i liegt (3) und zwischen NG bzw. PNG_i und PNG_j darf kein Knoten liegen mit der Markierung VK bzw. VERB (7), (8).

Strukturbedingungen für ein <u>Genitivattribut</u> sind nach dem Schema $sb_{g.attr}$ zu definieren, wobei i, j zwei beliebige Indizierungen, $i \neq j$, und N und NG_j für beliebige komplex notierte Symbole mit der Hauptkategorie N bzw. NG stehen:

$$sb_{g.attr} = (\text{UND } (\text{EQ X1 } (\text{ODER } NG_i \text{ PNG})) \quad (1)$$
$$(\text{DOM X1 N}) \quad (2)$$
$$(\text{LFT X1 } NG_j/\text{K=GEN}) \quad (3)$$

<u>Festlegung</u>

Erfüllt ein Baum eine nach $sb_{g.attr}$ gebildete Strukturbedingung, dann sagen wir auch: NG_j ist ein (<u>mögliches</u>) (<u>Genitiv-</u>) <u>Attribut</u> <u>von</u> N. Zur Vereinfachung vereinbaren wir:

<u>Schreibkonvention für eine Strukturbedingung "mögliches Genitivattribut"</u>

Für eine nach dem Schema $sb_{g.attr}$ gebildete Strukturbedingung sb schreiben wir auch

$$(\text{M.ATTR N } NG_j)$$

mit der Festlegung $(\text{M.ATTR N } NG_j) \equiv sb$.

<u>Erläuterung</u>

Wird ein Nomen N direkt dominiert von NG_i oder PNG (vgl. (1) - (2) oben), dann ist sein mögliches Genitivattribut eine NG_j, die unmittelbar rechts von NG_i bzw. PNG liegt (3) und für die Kasus=Genitiv gilt.

4.3.3 Strukturbedingung "Ergänzung"

Die Strukturbedingung "Ergänzung" definieren wir im

Hinblick auf Nominalgruppen (NG) oder präpositionale Nominalgruppen (PNG), die strukturell auf gleicher Ebene liegen wie der Verbkomplex (VK), d.h. die mit NG bzw. PNG markierten Knoten sind Geschwister von dem mit VK markierten Knoten. Wir legen zuerst die allgemeinere Strukturbedingung "Ergänzung" fest, um danach die Strukturbedingungen "Ergänzung im Nominativ" und "Ergänzung im Akkusativ" zu definieren.

Strukturbedingungen für eine <u>Ergänzung</u> sind nach dem Schema sb_{erg} zu bilden, wobei i eine beliebige Indizierung und y ein komplex notiertes Symbol mit der Hauptkategorie NG oder PNG:

$$
\begin{array}{lll}
sb_{erg} = (\text{UND} & (\text{DOM VK } V_i) & \\
 & (\text{ODER } (\text{LFT}^* \text{ VK y}) & \\
 & \qquad\quad (\text{LFT}^* \text{ y VK}))) &
\end{array}
$$

<u>Festlegung</u>

Erfüllt ein Baum eine nach dem Schema sb_{erg} gebildete Strukturbedingung, dann sagen wir auch y ist eine (<u>mögliche</u>) <u>Ergänzung</u> <u>von</u> V_i. Zur Vereinfachung vereinbaren wir:

<u>Schreibkonvention für eine Strukturbedingung "(mögliche) Ergänzung"</u>

Für eine nach dem Schema sb_{erg} gebildete Struturbedingung sb schreiben wir auch

$$(\text{M.ERG } V_i \text{ y}), \text{ wobei y} = NG_i \text{ oder } PNG_i$$

mit der Festlegung $(\text{M.ERG } V_i \text{ y}) \equiv \text{sb}$.

Strukturbedingungen für eine <u>Ergänzung im Nominativ</u> sind nach dem Schema sb_{erg1} zu bilden, wobei i eine beliebige Indizierung, $0 < n \leq 6$, und y eine Symbolvariable oder ein komplexes Symbol mit der Hauptkategorie NG bzw. PNG.

```
sb_erg1 = (UND (M.ERG V_i y)                        (1)
          (ODER
          (UND (EQ y NG/K=NOM,PN=n)                 (2)
               (DOM S/DIATHESE=AKTIV NG)            (3)
               (DOM VK/PN=n V_i))                   (4)
          (UND (EQ y PNG)                           (5)
               (DOM S/DIATHESE=PASSIV PNG)          (6)
               (ODER
               (UND (DOM PNG PRAEP/R=DAT)           (7)
                    (DOM PRAEP von))                (8)
               (UND (DOM PNG PRAEP/R=AKK)           (9)
                    (DOM PRAEP durch))))))          (10)
```

Erfüllt ein Baum eine nach sb_{erg1} gebildete Strukturbedingung, dann sagen wir auch: y ist eine (<u>mögliche</u>) <u>Ergänzung</u> <u>im</u> <u>Nominativ</u> <u>von</u> V_i. Zur Vereinfachung vereinbaren wir:

<u>Schreibkonvention für eine Strukturbedingung "(mögliche) Ergänzung im Nominativ"</u>

Für eine nach dem Schema sb_{erg1} gebildete Strukturbedingung sb schreiben wir auch

(M.ERG1 V_i y)

mit der Festlegung (M.ERG1 V_i y) $\equiv$ sb.

<u>Erläuterung</u>

Die Strukturbedingung "Ergänzung im Nominativ" ist eine Einschränkung der allgemeineren Strukturbedingung "Ergänzung", da für eine "Ergänzung im Nominativ" der Kasus vorgeschrieben wird (2) und die Diathese des

Satzes Aktiv sein muß (3). Da wir bei den Regeln von ÜG/KS nicht ständig alternative Regeln für Aktiv- und Passiv-Sätze angeben wollen, sehen wir den Fall vor, daß eine NG, die im Aktivsatz Ergänzung im Nominativ ist, in einem entsprechenden Passivsatz als PNG mit der Praeposition "durch" oder "von" formuliert ist ((5)-(10)). Dadurch erfüllt sowohl ein Aktiv- als auch ein Passiv-Satz eine Sturkturbedingung (M.ERG V X1) unter einer Knotenzuordnung, die im Aktivsatz dem Symbol X1 einen mit NG,K=NOM markierten Knoten zuordnet, im Passivsatz jedoch einen mit PNG markierten Knoten, der eine Präposition "durch" oder "von" dominiert. Im Passivsatz ist dann allerdings die Terminologie "Ergänzung im Nominativ" mißverständlich, weshalb wir zur Unterstreichung des mnemotechnischen Charakters der Terminologie lieber von ERG1 bzw. im nachstehenden Fall von ERG4 reden.

Strukturbedingungen für eine <u>Ergänzung im Akkusativ</u> sind nach dem Schema sb_{erg4} zu bilden, wobei i eine beliebige Indizierung, $0 < n \leq 6$, und NG für ein beliebiges komplexes Symbol mit der Hauptkategorie NG steht.

```
sb_erg4 = (UND (M.ERG V_i NG)
               (ODER
                (UND (EQ NG NG/K=AKK)
                     (DOM S/DIATHESE=AKTIV NG))
                (UND (EQ NG NG/K=NOM, PN=n)
                     (DOM S/DIATHESE=PASSIV VK/PN=n)
                     (DOM VK V_i))))
```

<u>Festlegung</u>

Erfüllt ein Baum eine nach sb_{erg4} gebildete Strukturbedingung sb, dann sagen wir auch: NG ist eine (mögliche) Ergänzung im Akkusativ von V_i. Zur Vereinfachung vereinbaren wir:

Schreibkonvention für eine Strukturbedingung "(mögliche) Ergänzung im Akkusativ"

Für die Strukturbedingung sb schreiben wir auch

$$(\mathrm{M.ERG4}\ V_i\ \mathrm{NG})$$

mit der Festlegung $(\mathrm{M.ERG4}\ V_i\ \mathrm{NG}) \equiv \mathrm{sb}$.

Weitere Konventionen für Strukturbedingungen führen wir in den folgenden Abschnitten ein, insbesondere die Bedingung TRANSLAT (vgl. 4.4.3) sowie die Verallgemeinerungen für Attribute und Ergänzungen: ATTR, ERG, ERG1, ERG4 (vgl. 4.4.4).

4.4 Grundlegende Überführungsstrategien

Die Regeln einer Übersetzungsgrammatik, wie sie in Kapitel 3 definiert wurden, verwenden wir im Rahmen einer Strategie, die im wesentlichen auf drei Schritten aufbaut:

1. Die Grundbausteine zur Konstruktion einer KS-Formulierung sind Kontextpattern, die hauptsächlich zu deutschen Lexemen formuliert werden und die angeben, welches KS-Symbol einem deutschen Lexem entspricht.

2. Kontextpattern beschreiben auch den KS-syntaktischen Kontext, der für ein KS-Symbol obligatorisch ist. Für die Konstituenten des Kontexts sind Leerstellen ausgewiesen, in die bereits

ausgefüllte Kontextpattern eingesetzt werden können. D.h. die Leerstellenregeln legen fest, wie aus mehreren einzelnen Teilstrukturen eine größere Struktur konstituiert wird.

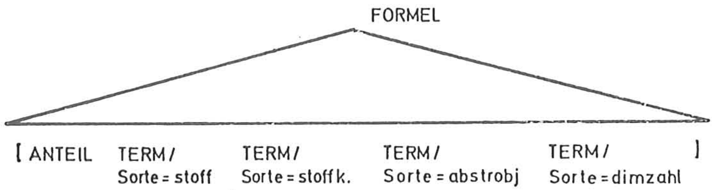
3. Dominiert ein Knoten mehrere Translate, i.e. ausgefüllte Kontextpattern, wird von T r a n s l a t h e b u n g s regeln beschrieben, welches dieser Translate zum Translat dieses Knotens angehoben wird.

Bei der Erläuterung dieser Schritte werden wir aufzeigen, wie mit Hilfe von Notationskonventionen die erforderlichen komplexen Regeln im Interesse einer praktischen Handhabbarkeit vereinfacht werden können, ohne den Umfang an Regeltypen einer Übersetzungsgrammatik aufzublähen.

4.4.1 Terminale Ersetzung

Die greifbarste Regularität des Zusammenhangs zwischen einer Ausgangs- (AS) und einer Zielsprache (ZS) ist die Entsprechung ihrer terminalen Symbole. Diese Regularität bringen wir jedoch nicht dadurch zum Ausdruck, daß bei einer Überführung ein terminales AS-Symbol durch ein entsprechendes ZS-Symbol ersetzt wird. Vielmehr nutzen wir für ÜG/KS die Möglichkeit aus, daß aufgrund der Grammatik von KS der engere syntaktische Kontext eines KS-Symbols in vielen Fällen wenige oder gar keine Varianten aufweist. Ist z.B. festgelegt, daß dem Verb "enthalten" das KS-Operationszeichen ANTEIL mit dem Sortentupel (stoff, stoffkoll, abstrobj, dimzahl) entspricht, dann folgt aus der Syntax von KS, daß ANTEIL nur zur Bildung eines dreistelligen Terms oder einer vierstelligen Formel verwendet werden kann. Geht man weiter davon aus, daß "enthalten" als Verb in einem Satz den Sachverhalt des "Enthaltens" und nicht das "Enthaltene" bezeichnet, kann aufgrund der Semantik von KS ANTEIL als Entsprechung zu "enthalten" nur im Rahmen einer Formel vorkommen. Daraus kann man schließen, daß in der KS-Struktur der Übersetzung eines deutschen Satzes mit "enthalten" auf jeden Fall die Struktur

(1)



enthalten sein wird. Wir verzichten hier auf die Angabe der Binnenstruktur des Baumes, da sie für die Folge nicht von Bedeutung ist und werden sie auch in den Regeln der Einfachheit halber nicht berücksichtigen. Dies wäre allerdings erforderlich, wenn mit Hilfe von ÜG/KS vollständige KS-Syntaxbäume aufgebaut werden sollten.

Betrachtet man terminal abgeleitete Formeln, die mit ANTEIL gebildet sind, dann sieht man, daß die Argumente von ANTEIL folgende Bezeichnungsfunktion haben: das 1. Argument, ein TERM der Sorte 'stoff', bezeichnet den Stoff, der in der vom 2. TERM bezeichneten Entität enthalten ist. Das 4. Argument bezeichnet den mengenmäßigen Anteil dieses Stoffes und das 3. Argument bezeichnet die Entscheidungsgrundlage, etwa einen

Laborbericht, aufgrund deren der Sachverhalt des ANTEIL-habens behauptet wird.

Legt man ANTEIL als Entsprechung des Verbs "enthalten" fest, kann man durch Untersuchung der natürlichsprachlichen Kontexte, in denen "enthalten" auftritt, feststellen, welche natürlichsprachlichen Konstituenten im Kontext von "enthalten" die gleichen Bezeichnungsfunktionen haben wie die Argumente von ANTEIL (vgl. hierzu das Beispielmaterial in 2.2). Ersetzt man in (1) die Terme durch die Überführungen der entsprechenden Konstituenten aus dem Kontext von "enthalten", entsteht aus (1) eine größere Struktur, die entweder nun ihrerseits in eine andere Struktur eingesetzt werden kann oder entsprechend dem Kontext weiterverändert werden muß.

Die Informationen über den syntaktischen Kontext eines KS-Symbols notieren wir mit einem "Kontextpattern", in dem die Stellen, in die Überführungen anderer AS-Konstituenten einzusetzen sind, durch Hilfssymbole markiert sind, so daß wir in Anlehnung an (1) den Klammerausdruck

(2) (FORMEL [ANTEIL A1 A2 A3 A4])

schreiben.

Um etwa bei "enthalten" differenzieren zu können zwischen dem Gebrauch als Verb und als Adjektiv (z.B. "die in den Proben <u>enthaltenen</u> Giftstoffe") ordnen wir ein Kontextpattern nicht einem Lexem, sondern einer Struktur zu, die aus dem Lexem und seiner lexikalischen Kategorie gebildet wird. D.h. (2) wird der als Liste dargestellten Struktur (V enthalten) zugeordnet. Diese Zuordnung erfolgt unter Verwendung des komplex notierten Hilfssymbols TR/ind=Ø mittels der Regel "Erweiterung um rechten Sohn":

(3) (EW.RSO (V enthalten)
(TR/ind=Ø (FORMEL [ANTEIL A1 A2 A3 A4])))

deren Anwendung in Abb. 4.2 dargestellt ist.

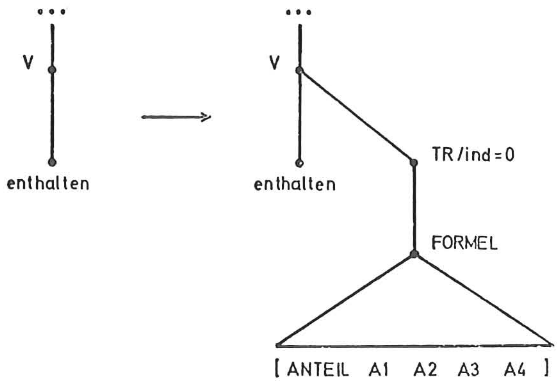


Abbildung 4.2: Kontextpatternregel

Ausgehend von (3) treffen wir folgende Festlegung:

Festlegung für Kontextpattern:

1. Hat eine Regel EW.RSO die Form

(EW.RSO l_1 (TR/ind=Ø l_2)) oder
(EW.RSO l_1 TR/ind=1)

wobei l_1 die Liste eines nichtverzweigenden Baumes, dessen terminaler Knoten mit einem Symbol aus VT_{AS}

markiert ist, TR/ind=$\emptyset$ ein Hilfssymbol, und l_2 eine Liste ungleich der leeren Liste, dann nennen wir l_1 eine AS-Basisstruktur, l_2 Kontextpattern von l_1 und die im Baum von l_2 mit Hilfssymbolen A1,...,An, n $\in \mathbb{N}$, markierten terminalen Knoten Leerstellen.

Bei einer Regel (EW.RSO l_1 TR/ind=1) sagen wir: l_1 wird das leere Kontextpattern zugeordnet.

2. Denjenigen Baum, der aus dem Baum von l_2 durch Ausfüllen aller Leerstellen entsteht, nennen wir Translat von l_1. Ist a ein Symbol aus V_{AS} und ist der Wurzelknoten oder der terminale Knoten des Baums von l_1 mit a markiert, sagen wir statt 'Translat von l_1' auch 'Translat von a'.

Erläuterungen

Dieser Festlegung liegt folgende Überlegung zugrunde:

(i) Die Teilbäume, die an als Leerstellen bezeichneten Knoten einzusetzen sind, werden nicht explizit angegeben wie z.B. bei "enthalten" die Angabe "Translat von Cadmium" oder "Translat von Abwasserprobe", da sonst für alle potentiellen Kontextrealisierungen Regeln angegeben werden müssen. Vielmehr werden die einzusetzenden Teilbäume durch syntaktische Relationen bezüglich der AS-Basis-Struktur des Kontextpatterns wie Attribut oder Ergänzung spezifiziert.

Wird nun der regelhafte Zusammenhang zwischen einer AS-Basis-Struktur und ihrem Kontextpattern durch eine Regel ER.LIT ("Ersetzte Struktur durch Literal") zum Ausdruck gebracht, d.h. wird die AS-Basisstruktur in einem Baum durch ihr Kontextpattern ersetzt, dann sind danach Einsetzungsregeln, deren Strukturbedingungen sich mit ERG oder ATTR auf die AS-Basisstruktur beziehen, nicht mehr anwendbar, da die AS-Basisstruktur gar nicht mehr vorhanden ist. Prozedural betrachtet wurde sie vom Kontextpattern "überschrieben". Andrerseits ist auch nicht ohne umfangreiche Änderung der in 3.5 vorgeschlagenen Regelformalismen möglich, ein Kontextpattern "separat" auszufüllen und dann erst die Ersetzung der AS-Teilstruktur vorzunehmen. Ein Kontextpattern muß deshalb durch eine Erweiterungsregel in einen Baum eingebracht werden, die die AS-Basisstruktur des Kontextpatterns unverändert läßt.

(ii) Auch zur Identifizierung eines einzusetzenden Translats ist neben der KS-syntaktischen Struktur des Translats sein gesamter AS-syntaktischer Kontext erforderlich, denn ein einzusetzendes Translat ist spezifiziert durch

- Strukturbeziehungen zwischen der AS-Entsprechung des Translats und der AS-Basisstruktur zum Kontextpattern (z.B. ATTR, ERG)

- Strukturbedingungen über die AS-Entsprechung des Translats (z.B. Spezifizierung möglicher Praepositionen)

- Bedingungen über das Translat selbst (z.B. Sortenrestriktionen)

(iii) Für die weiteren Schritte der Überführung ist es erforderlich, Translate eindeutig identifizieren zu können. Um dies in einfacher Weise zu ermöglichen, wird eine AS-Basisstruktur nicht einfach um das Kontextpattern erweitert, sondern um einen Baum, dessen Wurzelknoten mit dem Hilfssymbol TR/ind=Ø markiert ist und der den Wurzelknoten des Kontextpatterns direkt dominiert.

Schreibkonvention für Kontextpatternregeln

Da in allen Regeln für nicht-leere Kontextpattern das Symbol TR/ind=Ø mitnotiert werden muß, führen wir zur Verminderung von Redundanz und zur besseren Überschaubarkeit der Funktion einzelner Regeln folgende Schreibkonvention ein:

Ist l_1 eine Basisstruktur und l_2 ein Kontextpattern, dann schreiben wir

(CP l_1 l_2) für (EW.RSO l_1 (TR/ind=Ø l_2))
und
(CP l_1 ()) für (EW.RSO l_1 TR/ind=1)

und nennen die festgelegte Form CP-Regel.

4.4.2 Translathebung

Das Prinzip der Translathebung wurde bereits hinlänglich in Abschnitt 2.3 erläutert, so daß wir uns hier mit der formalen Definition der T r a n s l a t - h e b u n g s r e g e l von ÜG/KS begnügen können. Bei der Definition gehen wir von folgenden Punkten aus:

1. Die Einsetzung von Translaten in Leerstellen wird

durch die Regelfolge einer komplexen Regel formuliert, in der der Regel für die Einsetzung im engeren Sinn eine Regel folgt, die die Markierung des Knotens, der den Wurzelknoten des eingesetzten Translats dominiert, ersetzt durch das komplex notierte Symbol TR/ind=1.

2. Das Translat eines Knotens k_1 kann angehoben werden zum Translat eines Knoten k_2, wenn k_2 den Knoten k_1 direkt dominiert und wenn alle Geschwisterknoten von k_1 Knoten mit der Markierung TR/ind=1 direkt dominieren.

3. Dominiert ein Knoten k_1 nur solche Knoten, die ihrerseits jeweils mit TR/ind=1 markierte Knoten dominieren, ist das anzuhebende Translat nicht bestimmbar und die Struktur mit dem Wurzelknoten k_1 erhält als Translat lediglich einen Baum mit einem einzigen Knoten, der mit TR/ind=1 markiert ist.

<u>Festlegung für die Translathebung</u>

Sei R die Menge der Überführungsregeln von ÜG/KS; sei H das Hilfsvokabular von ÜG/KS; sei $TR^{(i)}$/ind=n $\in H^{(IN)}$; seien X1, X2, X3, X4 Symbolvariablen von ÜG/KS; seien $sb_1, \ldots, sb_4$ Strukturbedingungen.

Eine komplexe Regel $r \in R$ nennen wir <u>Translathebungsregel der Form 1</u>, gdw.:

(1) r = (UND r_1 r_2)

(2) r_1 = ((EW.RSO X1 A1) (UND sb_1 sb_2 sb_3))

(3) sb_1 = (DOM X1 $\neq TR_1$)

(4) sb_2 = (UND (DOM X1 X2)
(DOM X2 TR_2/ind=Ø))

(5) sb_3 = (FÜRALL X3 (EXIST TR_3/ind=1
(IMPLIK (DOM X1 X3)
(DOM X3 TR_3/ind=1))))

(6) r_2 = (ER.ST A1 TR_2/ind=Ø)

Erläuterung

Wir veranschaulichen die Translathebung durch das Baumschema in Abb. 4.3, in dem wir zur Knotenmarkierung die Symbolvariablen verwendet haben.

Die Translathebung erfolgt in zwei Schritten: mit der Regel r_1 wird zu einem beliebigen Knoten k_1, der der Symbolvariablen X1 zugeordnet ist, ein neuer Knoten k_2 mit der Markierung A1 als rechter Sohn hinzugefügt. D.h. es wird eine Leerstelle geschaffen, in die das anzuhebende Translat eingesetzt wird.

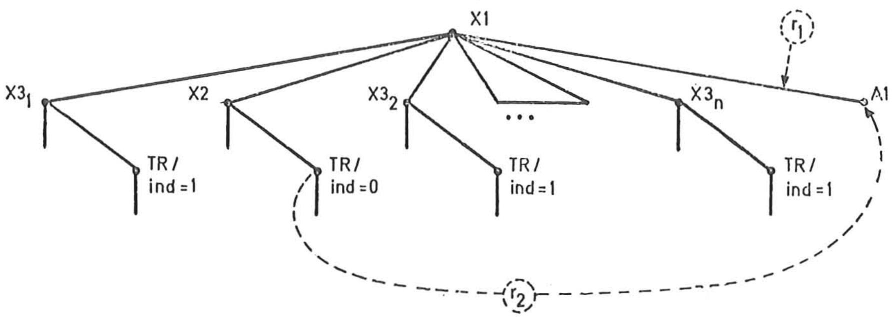


Abbildung 4.3: Translathebungsregel der Form 1

Mit der Regel r_2 wird in diese Leerstelle das Translat von X2 zuzüglich des mit TR/ind=Ø markierten Knotens eingesetzt. Die Strukturbedingung sb_1 ist erforderlich, um eine mehrfache Anwendung der Regel zu verhindern. In (5) schließlich wird bestimmt, daß die Regel r_1 nur anwendbar ist, wenn alle anderen Knoten, die k_1 dominiert, ihrerseits ein bereits eingesetztes Translat dominieren, d.h. also Knoten, die mit dem Symbol TR/ind=1 markiert sind.

Eine Regel r ∈ R nennen wir <u>Translathebungsregel der Form 2</u>, gdw.:

(1) r = ((EW.RSO X1 TR;ind=1) (UND sb_1 sb_2))

(2) sb_1 = (DOM X1 ≠TR_1)

(3) sb_2 = (FÜRALL X2 (EXIST TR_2/ind=1
(IMPLIK (DOM X1 X2)
(DOM X2 TR_2/ind=1))))

<u>Erläuterung</u>

Die Translathebungsregel der Form 2 berücksichtigt den Fall, daß alle Translate von Geschwisterknoten in Leerstellen eingesetzt wurden und deshalb von einem Knoten mit der Markierung TR/ind=1 dominiert werden. Das Schema in Abb. 4.4 illustriert einen solchen Fall, wobei wir wiederum die Symbolvariablen der Regel verwendet haben.

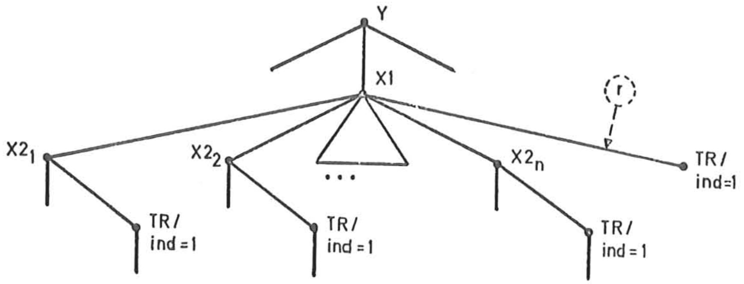


Abbildung 4.4: Translathebungsregel der Form 2

In diesem Fall ist nicht entscheidbar, welches der Translate zum Translat des mit X1 markierten Knotens in Abb. 4.4 angehoben werden soll. Da aufgrund des Merkmals "ind=1" ersichtlich ist, daß die Translate im größeren Kontext bereits berücksichtigt sind, muß sichergestellt werden, daß die Translathebung für y nicht dadurch verhindert wird, daß für X1 kein Translat ermittelt werden kann.

4.4.3 Ausfüllung von Leerstellen

In 4.4.1 haben wir bereits eine Möglichkeit der Ausfüllung von Leerstellen eines Kontextpatterns erläutert: die Einsetzung von Translaten anderer AS-Strukturen mit Hilfe einer Regel ER.ST ("Ersetze Struktur durch Struktur"), die dann die Form hat

((ER.ST a c) (UND (DOM TR_i c) sb))

wobei a die Leerstelle bezeichnet,

c im Normalfall den Wurzelknoten des Translats,

TR_i ein beliebiges indiziertes Symbol mit der Hauptkategorie TR,

(DOM TR_i c) verwendet das Symbol TR, durch das Translate erkennbar sind; d.h. der Wurzelknoten eines Translats wird dominiert von einem Knoten mit der Markierung TR,

sb ist eine beliebige Strukturbedingung, die u.a. den strukturellen Zusammenhang zwischen a und c näher bestimmt.

Im Zusammenhang mit der Translathebung haben wir in 4.4.2 festgelegt, daß ein Symbol TR/ind=∅ ersetzt werden muß durch TR/ind=1, wenn das von TR/ind=∅ dominierte Translat in eine Leerstelle eingesetzt wird. Die Ausfüllung einer Leerstelle durch ein Translat muß deshalb mit einer komplexen Regel formuliert werden, die die Form hat:

(UND ((ER.ST a c) (UND (DOM TR_i c) sb))
(ER.S TR_i TR/ind=1))

Zur Vereinfachung vereinbaren wir folgende

Schreibkonvention für Leerstellenregeln (SLOT-Regeln)

Eine komplexe Regel für die Ausfüllung von Leerstellen (SLOT-Regel) schreiben wir

((SLOT a c) sb')

mit der Festlegung

((SLOT a c) sb') ≡
(UND ((ER.ST a c) sb')
(ER.S TR_i TR;ind=1))

wobei sb' = (UND (DOM TR_i c) sb) und sb eine beliebige Folge von Strukturbedingungen.

D.h. wir sparen durch die Schreibkonvention das Schreiben der Regel ER.S ("Ersetze Symbol durch Symbol").

Bei näherer Betrachtung der Strukturbedingung sb' einer Leerstellenregel ((SLOT a c) sb') stellt sich heraus, daß sich für einen Teil dieser Strukturbedingung eine weitere Vereinfachung festlegen läßt. sb' wird in den meisten Fällen die Form (UND sb_1 sb_2 sb_3) haben mit sb_1 = (DOM TR_i c) und sb_2 = (DOM b TR_i). D.h. sb_1 legt fest, daß c ein Translat ist, während sb_2 den Wurzelknoten der AS-Struktur präzisiert, zu der die von c dominierte Struktur ein Translat ist. Mit Hilfe von sb_3 kann dann der strukturelle Zusammenhang zwischen der AS-Basisstruktur des Kontextpatterns mit der Leerstelle a einerseits und der von b dominierten Struktur andrerseits näher bestimmt werden.

<u>Beispiel</u>:

Die Leerstelle A1 des CP von "enthalten" ist durch das Translat einer NG auszufüllen, die in der syntaktischen Relation ERG4 zu "enthalten" steht. Formulieren wir die CP-Regel für "enthalten" mit der Regel

(CP (V enthalten) (FORMEL [ANTEIL A1 A2 A3 A4]))

dann können wir die Leerstellenregel für A1 angeben mit

```
((SLOT A1 TERM/Sorte=stoff)
  (UND (DOM TR TERM/Sorte=stoff)          (1)
       (DOM NG TR)                        (2)
       (ERG4 V NG)                        (3)
       (DOM V enthalten)))                (4)
```

(Die Strukturbedingung (4) ist überflüssig, wenn die beiden Regeln in einer konjunktiven Regel zusammengefaßt werden.)
Die ersten beiden Strukturbedingungen können wir auch in einer PATTERNbedingung zusammenfassen zu

```
(PATTERN.BA (NG(TR TERM/Sorte=stoff)))
```

Zur Verbesserung der Lesbarkeit der Regeln schreiben wir für diese Bedingung auch (TRANSLAT NG TERM/Sorte =stoff) und legen diese Abkürzung fest in einer

<u>Schreibkonvention für eine Strukturbedingung TRANSLAT</u>:

Zur Spezifizierung eines Translats mit der Wurzelknotenmarkierung c, dessen AS-Struktur von einem Wurzelknoten mit der Markierung b dominiert wird, schreiben wir auch

```
(TRANSLAT b c)
```

mit der Festlegung

```
(TRANSLAT b c) ≡
(UND (DOM b TR) (DOM TR c))≡ (PATTERN.BA (b(TR c)))
```

Eine andere Möglichkeit der Ausfüllung von Leerstellen ist vorzusehen für den Fall, daß eine natürlichsprachliche Konstituente, deren Translat für die Ausfüllung vorgesehen ist, in einem gegebenen Kontext gar nicht formuliert ist.

<u>Beispiel</u>:

Für das deutsche Wort "Probe" legen wir das Kontext-pattern (TERM/Sorte=stoffkoll [PROBE A1 A2]) fest mit folgenden Bestimmungen für die Leerstellen:

A1 ist auszufüllen mit einem TERM/Sorte=betrieb, der Translat eines Praepositionalattributs zu "Probe" mit einer der Praepositionen "von", "in", "bei" ist;

A2 ist auszufüllen durch einen TERM/Sorte=int, der ebenfalls Translat eines Praepositional-attributs von "Probe" ist, mit einer der Praepositionen "von", "in",

so daß Formulierungen berücksichtigt sind wie

die Probe	beim im vom	Betrieb x in Stuttgart	vom 25.11.76 im Jahr 1976

Für die Zuordnung des Kontextpatterns und die Ausfüllung der Leerstellen kann dann folgende vorläufige konjunktive Regel formuliert werden:

(UND
(CP (N_1 PROBE) (TERM/Sorte=stoffkoll [PROBE A1 A2]))
((SLOT A1 $TERM_1$/Sorte=betrieb)
(UND (TRANSLAT PNG_1/K=DAT $TERM_1$)
(ATTR N_1 PNG_1)
(DOM PNG_1 $PRAEP_1$)
(DOM $PRAEP_1$ (ODER bei in von)))))

```
((SLOT A2 TERM2/Sorte=int)
  (UND (TRANSLAT PNG2/K=DAT TERM2)
       (ATTR N1 PNG2)
       (DOM PNG2 PRAEP2)
       (DOM PRAEP2 (ODER von in )))))
```

Versucht man nun diese Regel auf den Beispielsatz

"Enthielten die Proben bei Lauxmann in Stuttgart Cadmium?"

anzuwenden und legt man als Regel für Lauxmann

(CP (NPR Lauxmann) (TERM/Sorte=firma G-L))

fest, dann ist die Regel für Leerstelle A1 nicht anwendbar, da "Lauxmann" als TERM der Sorte "firma" übersetzt wird, für die Leerstelle A1 des PROBE-Patterns jedoch ein TERM der Sorte "betrieb" einzusetzen ist. Hält man an dem Kontextpattern für "Lauxmann" fest, ohne gleichzeitig eine alternative Regel der Form

(CP (NPR Lauxmann) (TERM/Sorte=betrieb [BETRIEB G-L A1]))

zu formulieren+), kann man für die Ausfüllung von A1 im PROBE-Pattern im Rahmen einer disjunktiven Regel die Alternative formulieren:

```
(UND (ER.LIT A1 (TERM4/Sorte=betrieb [BETRIEB A3 A4]))
     ((SLOT A3 TERM5/Sorte=firma)
       (UND (TRANSLAT PNG1/K=DAT TERM5)
```

+) durch eine solche zweite Regel würde die Mehrdeutigkeit des Eigennamens "Lauxmann", der einmal eine Firma gleichen Namens und einmal verschiedene Betriebe der Firma bezeichnen kann, besser wiedergegeben.

```
        (ATTR N1 PNG1)
        (DOM PNG1 PRAEP1)
        (DOM PRAEP1 (ODER bei in von))))
   ((SLOT A4 TERM6/Sorte=ort)
     (UND (TRANSLAT PNG3/K=DAT TERM6)
        (DOM PNG1 N2)
        (ATTR N2 PNG3)
        (DOM PNG3 PRAEP3)
        (DOM PRAEP3 in))))
```

Durch die Regel ER.LIT ("Ersetze Struktur durch Literal") wird in die Leerstelle A1 ein neues Pattern eingesetzt, das seinerseits wieder zwei Leerstellen, A3 und A4, hat. A3 wird durch das Translat eines Attributs von "Probe" ausgefüllt, während in A4 ein Term der Sorte 'ort' eingesetzt wird, der Attribut dieses Attributs von "Probe" ist (vgl. Abb. 4.5).

Ein anderes Beispiel für die Einsetzung von Literalen in Leerstellen eines Kontextpatterns ist die Einsetzung von KS-Variablen, die zu unterscheiden sind von den Symbolvariablen der Übersetzungsgrammatik ÜG/KS. Die Leerstellen eines (Kontext-)Patterns sind mit Hilfssymbolen wie A1, A2 etc. gekennzeichnet, die keine Symbole von KS sind.
Kann nun eine Leerstelle nicht ausgefüllt werden, da kein Translat im Kontext der Leerstelle die notwendigen Bedingungen erfüllt, muß das Hilfssymbol der Leerstelle durch ein "neutrales" Symbol oder eine "neutrale" Struktur von KS ausgefüllt werden. Als solche Symbole betrachten wir die Variablen von KS.

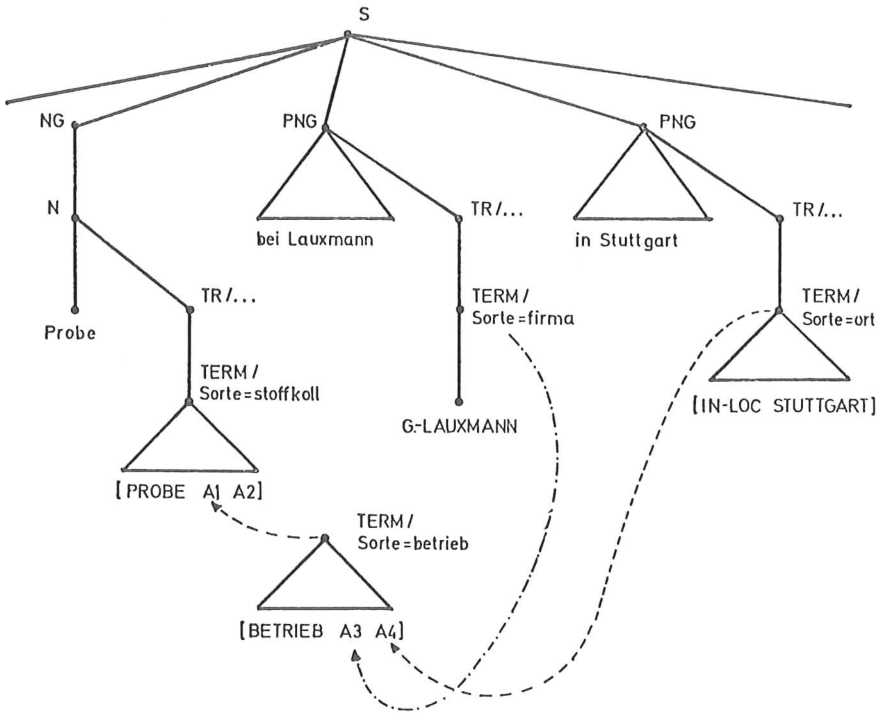


Abbildung 4.5

Zur Erläuterung des Vorgehens greifen wir auf die voranstehende Regel und auf Abb. 4.5 zurück.
Für die Leerstelle A4 ist die Einsetzung eines Terms der Sorte "ort" vorgeschrieben, der Translat einer PNG sein soll. Läßt man im betrachteten Beispielsatz die Ortsangabe weg, so daß der Satz lautet, "Ent-

hielten Proben bei Lauxmann Cadmium?" dann kann die Leerstelle A4 aufgrund der bisherigen Regeln nicht ausgefüllt werden. Um dies zu vermeiden, formulieren wir die Regel

(ER.LIT A4 (TERM/Sorte=ort (EXISTVAR X1.ORT)))

durch die in die Leerstelle A4 die in Abb. 4.6 dargestellte Struktur eingesetzt wird.

Abbildung 4.6

Dabei ist allerdings das Symbol EXISTVAR ein Hilfssymbol, für dessen Beseitigung entsprechende Regeln anzugeben sind, die formulieren, wie Variablen durch Quantoren zu binden sind. Das Symbol EXISTVAR ist dann in diesem Zusammenhang eine Markierung dafür, daß die Variable X1.ORT durch einen Existenzquantor zu binden ist.

Die Leerstellenregel für die Leerstelle A4 ist dann zu reformulieren als disjunktive Regel.

(ODER ((SLOT A4 $TERM_6$/Sorte=ort) ....)
(ER.LIT A4 (TERM/Sorte=ort (EXISTVAR X1.ORT))))

4.4.4 Transformationen der AS-Struktur

Wir haben bereits in 4.2 darauf hingewiesen, daß in den ausgangssprachlichen Strukturen von ÜG/KS für eine Nominalgruppe (NG) oder eine Nominalgruppe mit Praeposition (PNG) oft nicht entscheidbar ist, ob sie Ergänzung zum Verb oder Attribut zu einem Nomen ist.
In der Praxis zeigt sich zwar, daß die Ambiguität, ob eine Konstituente in der Relation "Attribut" bzw. "Ergänzung" steht, in zahlreichen Fällen aufgelöst werden kann mit Hilfe der Kontextpattern und ihren Restriktionen für die Ausfüllung von Leerstellen. Trotzdem bleibt ein Rest von nichtauflösbaren Ambiguitäten, der zur Konsequenz hat, daß mehrere Ableitungen möglich sind. In beiden Fällen ist jedoch festzuhalten, ob und welche Disambiguierung möglich war bzw. welche Annahme gemacht wurde. M.a.W.: wurde eine Leerstellenregel angewendet, weil die Strukturbedingung "Attribute" erfüllt war für eine Konstituente PNG, so daß ihr Translat in eine Leerstelle eingesetzt wurde, dann darf dieses Translat nicht gleichzeitig an einer anderen Stelle eingesetzt werden, für die eine Strukturbedingung "Ergänzung" erfüllt sein muß und sofern diese von PNG ebenfalls erfüllt ist.

Beispiel:

In dem Satz "Wurde in Proben bei Lauxmann in Stuttgart Cadmium festgestellt?" kann die PNG_2 "in Stuttgart" sowohl Attribut zur PNG_1 "bei Lauxmann" als auch Praepositionalergänzung zu "feststellen" sein. Wir gehen davon aus, daß im Kontextpattern von "feststellen" eine Leerstelle für einen $TERM_1$ der Sorte "ort" vorgesehen ist. Daneben ist auch entsprechend dem Beispiel von 4.4.3 eine Leerstelle für einen $TERM_2$ der Sorte "ort" in dem BETRIEB-Pattern vorgesehen, das in das Kontextpattern von Probe eingesetzt werden kann. Wir nehmen weiter an, daß $TERM_1$ im CP von "feststellen" den Ort bezeichnet, an dem das behauptete "feststellen" erfolgte und daß dieser Ort in der Regel Ort eines Untersuchungslabors ist und nicht der Ort, an dem die Probe gezogen wurde.

Will man unter Bezug auf einen hier nicht näher erläuterten fachsprachlichen Kontext die PNG_2 des o.a. Fragesatzes als Attribut zu "Lauxmann" bzw. zu "Proben" interpretieren, muß das Translat der PNG_2 als $TERM_2$ in das BETRIEB-Pattern eingesetzt werden. Es darf dann jedoch nicht gleichzeitig als $TERM_1$ in das CP von "feststellen" eingesetzt werden, da aus der Herkunft der Proben nicht auf den Ort der Schadstoff-Feststellung geschlossen werden kann.

Dies bedeutet für die Regeln von ÜG/KS, daß im Beispielfall den Regeln für "entnehmen" irgendwie "mitgeteilt" werden müßte, daß die PNG_2 als Attribut behandelt wurde und deshalb nicht mehr als Ergänzung verwendet werden darf. Die einfachste Lösung wäre die Festlegung, daß ein Translat nur einmal in eine Leerstelle eingesetzt werden darf. Da nach der Einsetzung die Markierung des Knotens, der das Translat dominiert, ersetzt wird durch TR/ind=1, dürften dann nur noch Translate mit der Markierung TR/ind=Ø zur Einsetzung verwendet werden. Dies ist allerdings eine sehr weitreichende Festlegung, deren Konsequenzen schwer zu überblicken sind. Da unsere Untersuchungen dazu auch noch nicht abgeschlossen sind, schlagen wir einen anderen Weg ein, der uns linguistisch plausibler erschein: die Veränderung der ausgangssprachlichen Teile der Baumstruktur bei Einsetzung des Translats. Während durch die bisher besprochenen Regeln weder die Markierungsrelation noch die Dominanzrelation und Linksrelation von Knoten, die mit Symbolen des ausgangssprachlichen Vokabulars von ÜG/KS markiert sind, verändert wurden, führen wir nun Regeln ein, die solche Veränderungen bewirken.

Festlegung für Transformationen von AS-Teilstrukturen

1. Transformation bei Attributen

Soll in eine Leerstelle A des Kontextpatterns einer AS-Basisstruktur u das Translat einer AS-struktur v eingesetzt werden und ist die Einsetzung an die Bedingung gebunden, daß v Attribut zu u ist, dann wird derjenige Baum, in dem u Teilbaum ist, rechts vom Wurzelknoten von u um v erweitert. v wird an seiner ursprünglichen Stelle gelöscht. Eine solche Erweiterung nennen wir dann auch AS-Strukturtransformation.

Beispiel:

Der obere Baum von Abb. 4.7 zeigt einen Ausschnitt der Struktur des Satzes "Enthielten die Proben vom 25.11.77 Cadmium?", wobei wir die Translate, die uns hier nicht näher interessieren, in der Abbildung weggelassen haben. Wir gehen davon aus, daß die Leerstelle A2 des CP von "Probe" auszufüllen ist durch einen Term der Sorte 'int', der Translat einer PNG ist, die Praepositionalattribut von "Probe" ist. Das Ergebnis der dann auszuführenden AS-Strukturtransformation ist im unteren Baum von Abb. 4.7 dargestellt.

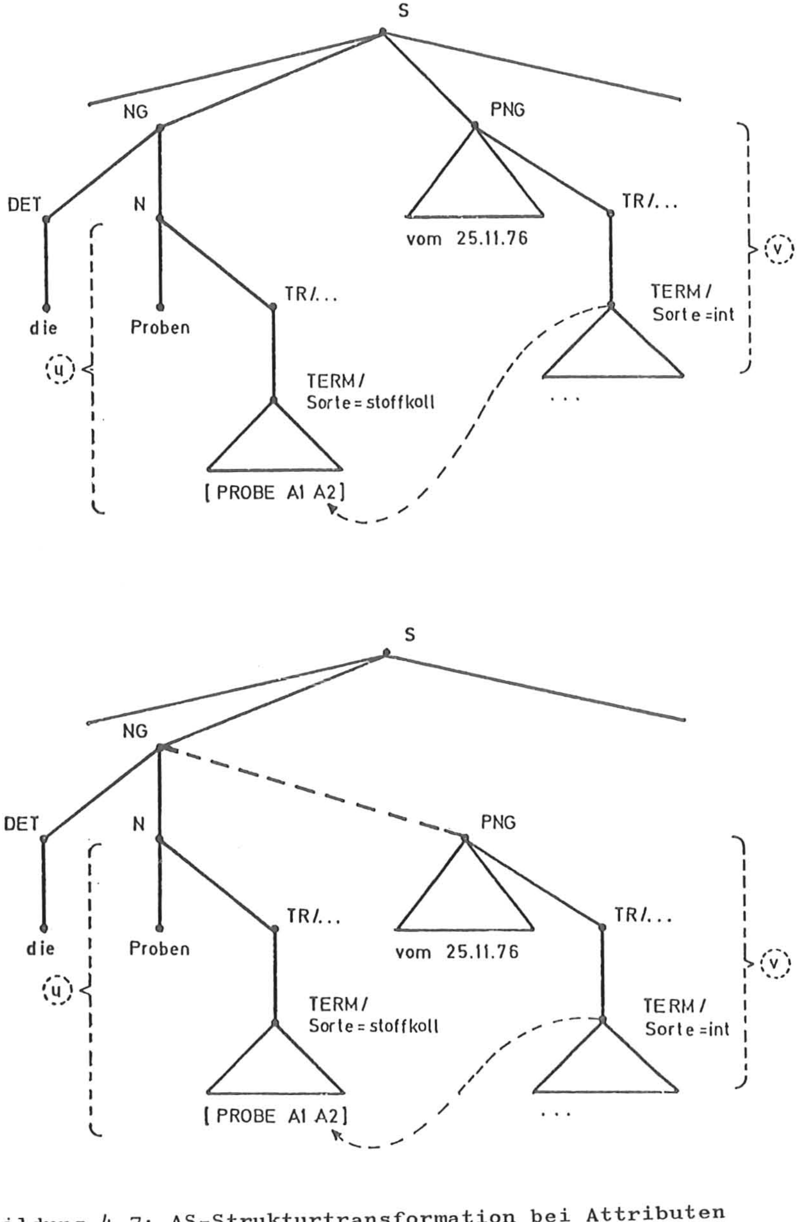


Abbildung 4.7: AS-Strukturtransformation bei Attributen

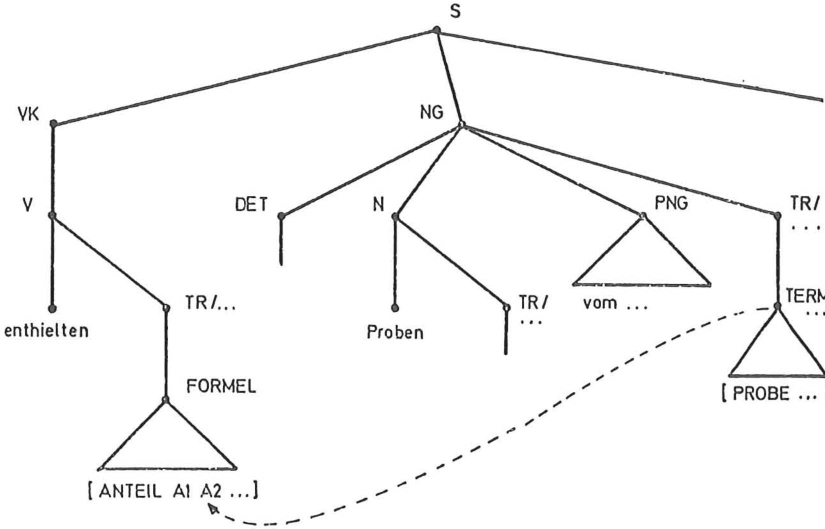
2. Transformationen bei Ergänzungen

Soll in eine Leerstelle A des Kontextpatterns einer AS-Basisstruktur, deren Wurzelknoten mit dem Symbol V markiert ist, das Translat einer AS-Struktur u eingesetzt werden und ist die Einsetzung an die Bedingung gebunden, daß u Ergänzung zu V ist, dann wird u ersetzt durch einen Baum v, dessen Wurzelknoten nichtverzweigend ist und mit dem Hilfssymbol NK (für Nominalkomplex) markiert ist, so daß gilt: u ist terminaler Teilbaum von v und der Wurzelknoten von v dominiert direkt den Wurzelknoten von u. Eine solche Veränderung nennen wir ebenfalls AS-Strukturtransformation.

Beispiel:

Der obere Baum von Abb. 4.8 zeigt einen Ausschnitt des o.a. Beispielsatzes "Enthielten die Proben vom 25.11.77 Cadmium?", nachdem die AS-Strukturtransformation für Attribute durchgeführt wurde.
Wir gehen davon aus, daß die Leerstelle A2 des CP von "enthalten" auszufüllen ist durch das Translat einer Ergänzung und daß NG sowie sein Translat die entsprechenden Strukturbedingungen für die Einsetzung in A2 erfüllen. Das Ergebnis der dann auszuführenden AS-Strukturtransformation ist im unteren Baum von Abb. 4.8 dargestellt.

Bevor wir die Regeln definieren, die solche AS-Strukturtransformationen beschreiben, betrachten wir zuerst die Konsequenzen für die Strukturbedingungen für Attribute und Ergänzungen.



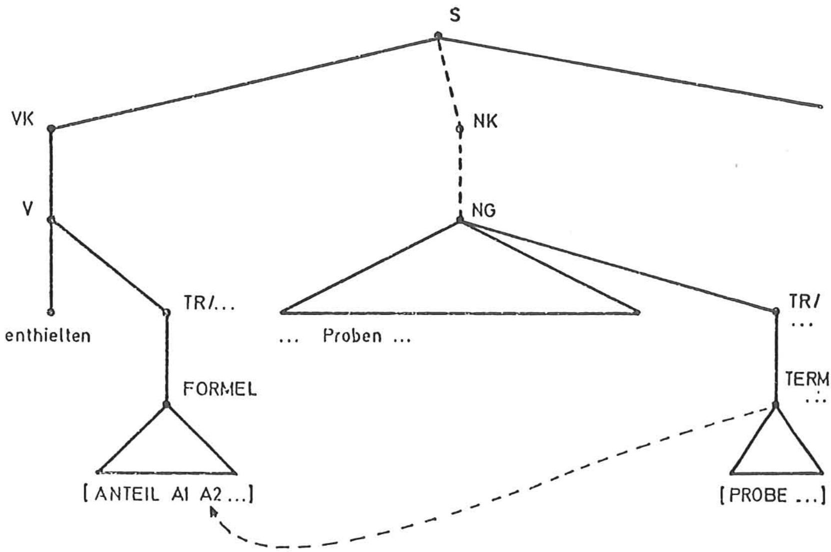


Abbildung 4.8: AS-Strukturtransformation bei Ergänzungen

Da in der Festlegung der Strukturbedingungen für mögliche Attribute AS-Strukturtransformationen nicht berücksichtigt sind, definieren wir eine Strukturbedingung "transformiertes Attribut" (T.ATTR) und fassen die Strukturbedingungen M.ATTR und T.ATTR zu einer Strukturbedingung "Attribut" (ATTR) zusammen. In ähnlicher Weise verfahren wir dann auch mit den Strukturbedingungen für Ergänzungen.

<u>Festlegung</u>

Ist y ein komplexes Symbol mit der Hauptkategorie NG oder PNG und erfüllt ein Baum eine nach dem Schema

$$
\begin{aligned}
sb_{t.attr} = (\mathrm{UND}\ & \\
& (\mathrm{ODER}\ (\mathrm{EQ}\ y\ \mathrm{PNG}) \\
& \qquad\quad (\mathrm{EQ}\ y\ \mathrm{NG/K{=}GEN})) \\
& (\mathrm{LFT}^{*}\ N\ y))
\end{aligned}
$$

gebildete Strukturbedingung, dann sagen wir, y ist ein (<u>transformiertes</u>) <u>Attribut</u> <u>von</u> N. Zur Vereinfachung vereinbaren wir:

<u>Schreibkonvention für Strukturbedingungen "transformiertes Attribut" (T.ATTR) und "Attribut" (ATTR):</u>

Für eine nach dem Schema $sb_{t.attr}$ gebildete Strukturbedingung sb schreiben wir auch

$$(\mathrm{T.ATTR}\ N\ y)$$

mit der Festlegung (T.ATTR N y) ≡ sb.

Für eine nach dem Schema

$$
\begin{aligned}
sb_{attr} = (\mathrm{ODER}\ & (\mathrm{M.ATTR}\ N\ y) \\
& (\mathrm{T.ATTR}\ N\ y)
\end{aligned}
$$

gebildeten Strukturbedingung sb schreiben wir auch

(ATTR N y)

mit der Festlegung (ATTR N y) ≡ sb und nennen y ein <u>Attribut von</u> N.

<u>Festlegung</u>

Ist y ein komplexes Symbol mit der Hauptkategorie NG oder PNG und erfüllt ein Baum eine nach dem Schema

```
sb_t.erg = (UND (EQ   y   (ODER NG PNG))
                (DOM NK y)
                (DOM VK V)
                (ODER (LFT* VK y)
                      (LFT* y VK)))
```

gebildete Strukturbedingung, dann sagen wir auch, y ist eine <u>transformierte Ergänzung</u> von V. Zur Vereinfachung vereinbaren wir:

<u>Schreibkonvention für Strukturbedingungen</u>

"<u>(transformierte) Ergänzung</u>"
(T.ERG bzw. ERG)

"<u>(transformierte) Ergänzung im Nominativ</u>"
(T.ERG1 bzw. ERG1)

"<u>(transformierte) Ergänzung im Akkusativ</u>"
(T.ERG4 bzw. ERG4):

Für eine nach dem Schema $sb_{t.erg}$ gebildete Strukturbedingung sb schreiben wir auch

(T.ERG V y)

mit der Festlegung (T.ERG V y) ≡ sb.

Für eine nach dem Schema

$$sb_{erg} = (\text{ODER} \quad (\text{M.ERG V y}) \quad (\text{T.ERG V y}))$$

gebildete Strukturbedingung sb schreiben wir auch

(ERG V y)

mit der Festlegung (ERG V y) ≡ sb.

Die Herleitung von Strukturbedingungen

(T.ERG1 V NG)
(T.ERG4 V NG)
(ERG1 V NG)
(ERG4 V NG)

erscheint uns auf der Hand zu liegen, so daß wir auf eine detaillierte Explikation verzichten.

Regel zur AS-Strukturtransformation bei Attributen

Die Ausfüllung einer Leerstelle a im Kontextpattern eines Nomens N durch das Translat t eines Attributs y von N formulieren wir als disjunktive Regel r = (ODER r_1 r_2). Die Regel r_1 beschreibt den Fall, daß y ein mögliches Attribut (M.ATTR) von N ist, wobei dann die erforderliche AS-Strukturtransformation zu beschreiben ist. Die Regel r_2 wird für den Fall definiert, daß y bereits ein transformiertes Attribut (T.ATTR) von N ist. Die Regel r ist dann nach folgendem Schema zu bilden, wobei a und B1 Hilfssymbole, N ein komplexes Symbol mit der Hauptkategorie N, y ein komplexes Symbol mit der Hauptkategorie NG oder PNG, sb eine beliebige Strukturbedingung:

```
r_attr = (ODER
           (UND ((SLOT a t)
                  (UND (TRANSLAT y t)
                       (M.ATTR N y)
                        sb))

                 (EW.RBR N B1)
                 (ER.ST B1  y)
                 (ER.LIT y ()))

           ((SLOT a t) (UND (TRANSLAT y t)
                            (T.ATTR N y)
                             sb)))
```

Zur Vereinfachung vereinbaren wir:

<u>Schreibkonvention für Leerstellenregeln mit AS-Strukturtransformationen für Attribute</u>:

Für eine Regel r, die nach dem Schema r_{attr} gebildet ist, schreiben wir auch

```
r' = ((SLOT.ATTR a t) (UND (TRANSLAT y t)
                           (ATTR N y)
                            sb))
```

mit der Festlegung r' ≡ r.

<u>Erläuterung</u>

Zur Erläuterung greifen wir zurück auf die Regeln für "Probe" (vgl. 4.4.3), die reformuliert lauten:

```
(UND
 (CP(N_1 Probe)(TERM/Sorte=stoffkoll [ PROBE A1 A2])) (i)
 ((SLOT.ATTR A1 TERM_1 Sorte=betrieb)                  (ii)
  (UND (TRANSLAT PNG_1/K=DAT TERM_1)
       (ATTR N_1 PNG_1)
```

(DOM PNG$_1$ PRAEP$_1$)
(DOM PRAEP$_1$ (ODER bei in von))))
((SLOT.ATTR A2 TERM$_2$/Sorte=int) ...)) (iii)

Die Regel (i) ist die CP-Regel für Probe, während die Regeln (ii) bzw. (iii) die Regeln sind zur Ausfüllung der Leerstellen A1 und A2 durch die Translate von Attributen. Entsprechend der o.a. Schreibkonvention kann die Regel (ii) für die Leerstelle A1 aufgelöst werden in die Formulierung:

(ODER (1)
(UND ((SLOT A1 TERM$_1$/Sorte=betrieb) (2)
(UND (TRANSLAT PNG$_1$/K=DAT TERM$_1$) (3)
(M.ATTR N$_1$ PNG$_1$) (4)
(DOM PNG$_1$ PRAEP$_1$) (5)
(DOM PRAEP$_1$ (ODER bei in von)))) (6)
(EW.RBR N$_1$ B1) (7)
(ER.ST B1 PNG$_1$) (8)
(ER.LIT PNG$_1$ ())) (9)
((SLOT A1 TERM$_1$/Sorte=betrieb) (10)
(UND (TRANSLAT PNG$_1$/K=DAT TERM$_1$) (11)
(T.ATTR N$_1$ PNG$_1$) (12)
(DOM PNG$_1$ PRAEP$_1$) (13)
(DOM PRAEP$_1$ (ODER bei in von))))) (14)

Abb. 4.9 zeigt einen Ausschnitt aus der Struktur des Satzes "Enthielten Proben in einem Betrieb von Lauxmann Cadmium?" nach Anwendung der CP-Regel für "Probe". Zur besseren Verständlichkeit wurden die Indizierungen, die in den o.a. Regeln verwendet wurden, auf die Symbole im Baum übertragen.

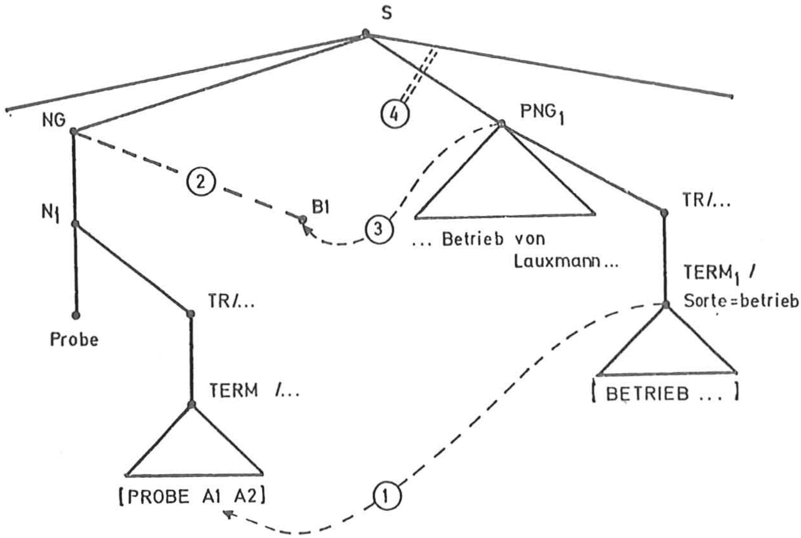


Abbildung 4.9: AS-Strukturtransformation bei Attributen

Die SLOT-Regel in Zeile (2) formuliert die Einsetzung des Translats eines Attributs von N_1 in die Leerstelle A1 (vgl. Pfeil Nr. 1 in Abb. 4.9). Die Regel in Zeile (7) beschreibt die Erweiterung von N_1 um einen rechten Bruderknoten, der mit dem Hilfssymbol B1 markiert ist (vgl. Pfeil 2). Aufgrund der Regel in Zeile (8) wird der mit B1 markierte Knoten durch das Attribut von N_1 ersetzt (vgl. Pfeil 3). Da durch Erweiterungsregeln ein Baum nur um Literale erweitert werden kann und nicht um bereits im Baum vorhandene Teilstrukturen, kann das Attribut nicht direkt als rechter Sohn von NG eingefügt werden.

Die Regel in Zeile (9) schließlich formuliert die Ersetzung der PNG_1 durch den leeren Baum. D.h. der Teilbaum mit der Wurzelknotenmarkierung PNG_1 wird gelöscht.

Die konjunktive Regel der Zeilen (3) - (9) ist jedoch, abgesehen von den anderen Strukturbedingungen, nur anwendbar, wenn es ein mögliches Attribut gibt bzw. wenn nicht bereits im Laufe einer Ableitung eine AS-Strukturtransformation für Attribute erfolgt ist. Für diesen Fall ist die Regel in Zeile (10) vorgesehen, die aufgrund der Strukturbedingung in Zeile (12) ermöglicht, das Translat eines transformierten Attributs noch einmal einzusetzen.

Regel zu AS-Strukturtransformation bei Ergänzungen

Analog zu den Regeln für Attribute wird eine Regel zur Einsetzung eines Translats t einer Ergänzung y in die Leerstelle a des Patterns eines Verbs V formuliert als disjunktive Regel r = (ODER r_1 r_2). Die Regel r_1 steht für den Fall, daß y eine mögliche Ergänzung (M.ERG) von V ist, wobei dann eine AS-Strukturtransformation durchzuführen ist. Die Regel r_2 ist erforerlich für den Fall, daß y bereits eine transformierte Ergänzung (T.ERG) von V ist. Die Regel r ist dann nach folgendem Schema zu bilden, wobei wir die gleichen Symbole verwenden wie beim Schema r_{attr} sowie das Hilfssymbol NK, die Strukturvariable & und V für ein komplexes Symbol mit der Hauptkategorie V:

```
r_erg = (ODER
          (UND ((SLOT a t)
                 (UND (TRANSLAT y t)
                      (M.ERG V y)
                      sb))
               (ER.LIT y (NK &)))
          ((SLOT a t) (UND (TRANSLAT y t)
                           (T.ERG V y)
                           sb)))
```

Zur Vereinfachung vereinbaren wir:

<u>Schreibkonvention für Leerstellenregeln mit AS-Strukturtransformationen für Ergänzungen</u>:

Für eine Regel r, die nach dem Schema r_{erg} gebildet ist, schreiben wir auch

```
r' = ((SLOT.ERG a t) (UND (TRANSLAT y t)
                          (ERG V y)
                          sb))
```

mit der Festlegung r' ≡ r.

<u>Erläuterung</u>

Die Regeln r und r' erläutern wir am Beispiel der Regeln für "enthalten", die ausschnittsweise entsprechend der eingeführten Schreibkonventionen formuliert werden können als

```
(UND (CP (V enthalten) (FORMEL [ ANTEIL A1 A2 A3 A4])) (i)
     ...                                               (ii)
     ((SLOT.ERG A2 TERM_1/Sorte=stoffkoll)             (iii)
               (UND (TRANSLAT X1 TERM_1)
                    (ERG1 V X1)))
     ...)
```

Die Regel in Zeile (i) ordnet dem Verb "enthalten" ein Kontextpattern mit vier Leerstellen zu. Wir betrachten hier lediglich die Regel für die Leerstelle A2 (iii). Diese Leerstelle ist auszufüllen durch ein Translat einer ERG1 von enthalten wie etwa "Proben" im Satz "Enthielten Proben in einem Betrieb von Lauxmann Cadmium?".

Entsprechend der Schreibkonvention kann die Leerstellenregel für A2 umformuliert werden zu der Regel

```
    (ODER                                            (1)
     (UND ((SLOT A2 TERM/Sorte=stoffkoll)            (2)
            (UND (TRANSLAT X1 TERM)                  (3)
                 (M.ERG1 V X1)))                     (4)
          (ER.LIT X1  (NK &)))                       (5)
     ((SLOT A2 TERM/Sorte=stoffkoll)                 (6)
       (UND (TRANSLAT X1 TERM)                       (7)
            (T.ERG V X1))))                          (8)
```

In Abb. 4.10 ist ein Ausschnitt des o.a. Beispielsatzes dargestellt, auf den die CP-Regel für "enthalten" angewendet wurde.

Die Regel in Zeile (2) formuliert die Ausfüllung der Leerstelle A2 durch einen Term der Sorte 'stoffkoll', der Translat einer Struktur ist, die mögliche Ergänzung zu "enthalten" sein soll. In Abb. 4.10 erfüllt die NG "Probe ...." diese Bedingung, so daß die Einsetzung vorgenommen werden kann (vgl. Pfeil 1 in Abb. 4.10). Aufgrund der Regel in Zeile (5) wird die Teilstruktur in Abb. 4.10, die von NG dominiert wird, durch eine Struktur mit der Wurzelknotenmarkierung NK (Nominalkomplex) ersetzt, wobei NK die von NG dominierte Struktur dominiert (vgl. 2 in Abbildung 4.10; zur Verwendung der Strukturvariablen "&" vgl. die Definition zur Regel ER.LIT in 3.5.2, Def. Ü20). Wurde diese AS-Strukturtransformation bereits bei einer vorangehenden Regelanwendung durchgeführt, trifft die Strukturbedingung von Zeile (4) nicht zu, so daß alternativ die SLOT-Regel in Zeile (6) anzuwenden ist.

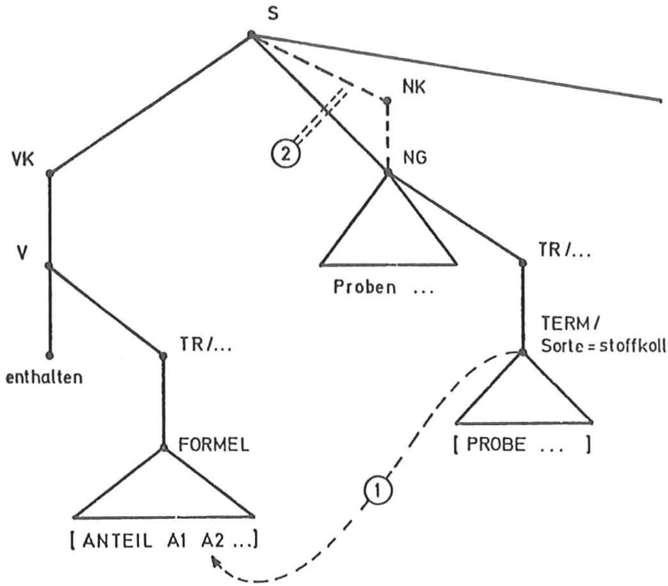


Abbildung 4.10: AS-Strukturtransformation bei Ergänzungen

4.4.5 Aufpfropfung von Strukturen

Die Kernstücke der bisher erläuterten Überführungsstrategie sind die Zuordnung von Kontextpattern, die Ausfüllung von Leerstellen und die Translathebung. Dabei wird die Zusammensetzung von Teiltranslaten einer Struktur zu größeren Translaten durch Leerstellenregeln für Kontextpattern beschrieben. Würde man es dabei belassen, dann würde vorausgesetzt, daß es zu jeder vom AS-Startsymbol S dominierten Struktur ein Kontextpattern gibt, das so viele Leerstellen hat, daß alle Teiltranslate aus dem Kontext in diese Leerstellen einsetzbar sind. Ein solches Verfahren ist vorstellbar, wenn die KS-Prädikate in Anlehnung an eine Dependenzgrammatik des Deutschen derart definiert werden, daß die Stelligkeit eines KS-Prädikats dem Valenzrahmen des korrespondierenden deutschen Wortes entspricht. Dies würde jedoch die Allgemeinheit von ÜG/KS insofern einschränken, als sie nur dann anwendbar wäre, wenn KS tatsächlich nach dem angedeuteten Valenzprinzip definiert wäre. Wir erweitern deshalb unsere Strategie auch für solche Fälle, für die vorhersagbar ist, daß entstehende Translate in keine Leerstellen von anderen Kontextpattern einsetzbar sind, so daß das Prinzip der Translathebung versagen würde, da mit Sicherheit mehrere nichteingesetzte Translate vorliegen. Zur Lösung dieses Problems verwenden wir Strukturen mit Leerstellen, die jedoch nicht wie Kontextpattern verwendet werden, sondern auf Bäume aufgepfropft werden. Einen einfacheren Fall einer Aufpfropfung haben wir am Beispiel von "kein" in 2.5 illustriert, während wir hier einen etwas komplexeren Fall am Beispiel von Adjektivgruppen (ADJG) untersuchen wollen. Das in der Folge skizzierte Vorgehen ist lediglich eine Möglichkeit, Adjektivgruppen zu behandeln, und es ist uns noch nicht endgültig klar, ob es nicht bessere Alternativen gibt, die hier jedoch nicht diskutiert werden sollen.

Für unsere Überlegungen betrachten wir die Nominalgruppe "die von der CLUA untersuchten Proben" aus dem Beispielsatz

> "Enthalten die von der CLUA untersuchten Proben Cadmium?"

in dem CLUA als Abkürzung für "Chem. Landesuntersuchungsanstalt" steht. Die Struktur des Satzes ist in Abb. 4.11 dargestellt.

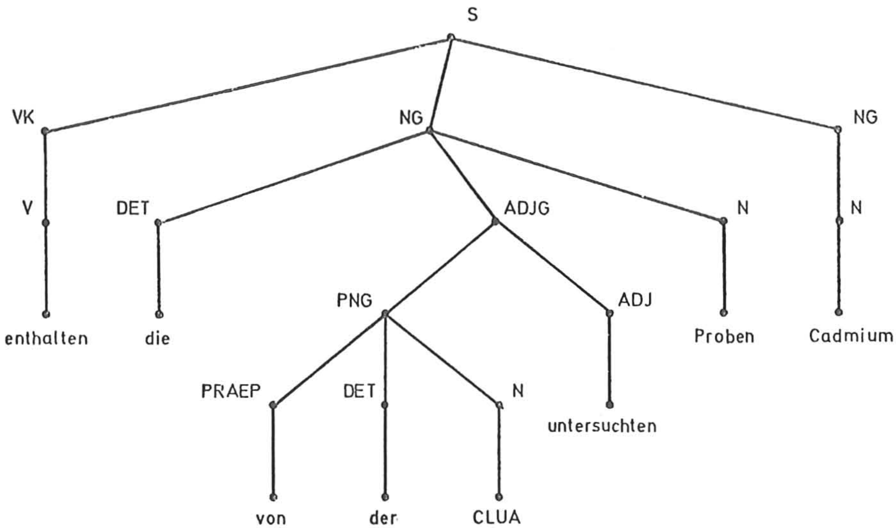


Abbildung 4.11

In dem hier zugrundegelegten KS-Vokabular gibt es keine Formulierung, die den Vorgang des "Untersuchens" beschreibt, sondern lediglich die Möglichkeit, das Ergebnis eines solchen Vorgangs zu beschreiben mit dem KS-Symbol LABORBERICHT. D.h. es wird davon ausgegangen, daß wenn etwas untersucht wurde (von einem Labor), dann manifestiert sich dies im Vorhandensein eines Laborberichts. Dem Verb "untersuchen" wird man deshalb das Kontextpattern (FORMEL [LABORBERICHT A1 A2 A3 A4]) zuordnen, wobei die Bedeutung des Patterns umschrieben werden kann mit: "Das Labor A1 erstellte zum Zeitpunkt A3 den Laborbericht A4 über (die Proben) A2". Das heißt also, daß die Leerstelle

A1 mit einem TERM/Sorte=labor

A2 mit einem TERM/Sorte=stoffkoll

A3 mit einem TERM/Sorte=int

A4 mit einem TERM/Sorte=abstrobj

auszufüllen ist. In einem Satz mit "untersuchen" wie etwa "Die CLUA untersuchte am 25.3.77 die Proben der Firma Lauxmann." findet man A1 als Translat der ERG1 von "untersuchen", A2 als Translat der ERG4, A3 als Translat einer Praepositionalergänzung und für A4 gibt es in der Regel keine natürlichsprachliche Entsprechung, so daß für A4 eine durch einen Existenzquantor zu bindende Variable einzusetzen ist.

Ordnet man nun dem Partizip "untersucht" das gleiche Kontextpattern wie dem Verb "untersuchen" zu (vgl. 1 in Abb. 4.12), wobei die Leerstellenregeln sehr ähnlich wie beim Verb "untersuchen" formuliert werden können, dann entsteht als Translat eine Formel.

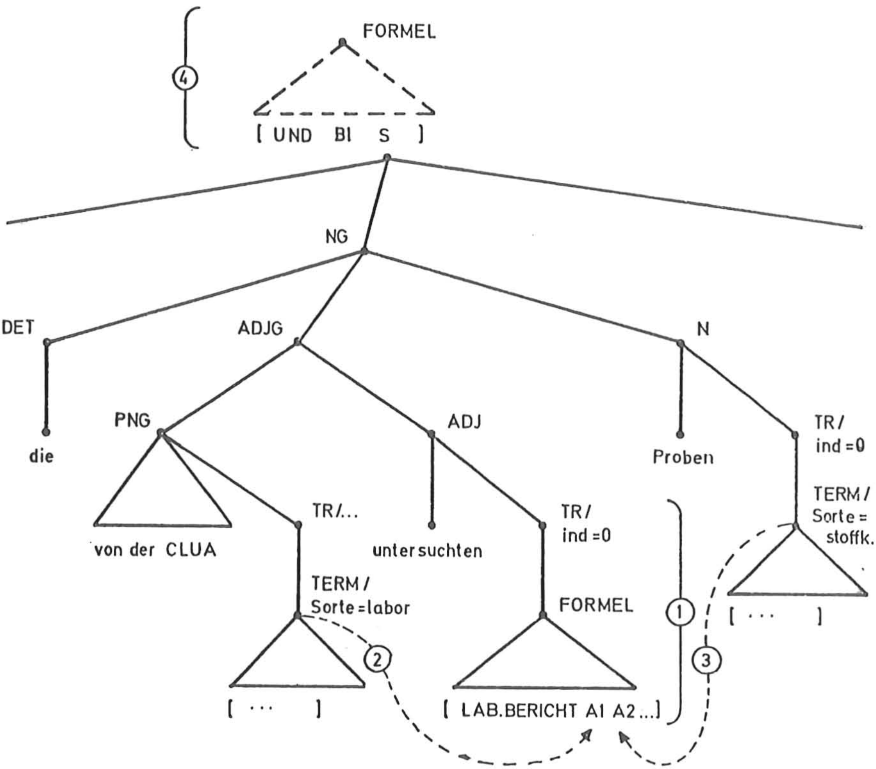


Abbildung 4.12

Daraus ergeben sich zwei Probleme:

1. Wird das Translat des Nomens "Probe" aufgrund einer SLOT-Regel in die Leerstelle A2 eingesetzt (vgl. Pfeil 3 in Abb. 4.12), dann erhält das von N dominierte Translat die Markierung TR/ind=1 für die erfolgte Einsetzung. Die als Translat von "untersucht" entstandene Formel wird dann nicht nur zum Translat der ADJG angehoben, son-

dern auch zum Translat der NG. Diese NG ist jedoch ERG1 zu "entnehmen" und in das Kontextpattern von "entnehmen" soll - wie wir bei früheren Beispielen gezeigt haben - ein TERM/Sorte=stoffkoll eingesetzt werden, der Translat der ERG1 sein soll und wie er auch normalerweise bei Nominalgruppen mit "Probe" entsteht. Dieses Problem kann dadurch gelöst werden, daß die Ausfüllung der Leerstelle A2 der FORMEL nicht mit einer SLOT-Regel festgelegt wird, sondern mit einer einfachen Regel "Ersetze Struktur durch Struktur" (ER.ST), die lediglich A2 durch das Translat von "Probe" ersetzt, das von N dominierte TR/ind=0 jedoch nicht verändert.

2. Da weder im Kontextpattern von "Probe" noch in dem von "enthalten" eine Leerstelle für eine Formel vorgesehen ist, gibt es im Beispielsatz keine Möglichkeit, das Translat von "untersucht" bzw. nach der Translathebung das Translat von ADJG in eine Leerstelle einzusetzen. Bei der Lösung dieses Problems gehen wir davon aus, daß die den Verben zugeordneten Kontextpattern Formeln beschreiben und daß die aus der Überführung des Verbkomplexes (VK) eines Satzes entstehende Formel das "Kerntranslat" des Satzes ist. Entstehen nun bei einer Überführung neben dem Kerntranslat weitere Formeln, werden diese Formeln sowie das Kerntranslat mit Hilfe von UND zu einer Konjunktion von Formeln verbunden.

Zur Konstruktion einer solchen Formelkonjunktion verwenden wir den Baum der Liste (FORMEL [UND B1 B1]), der in Abb. 4.13 dargestellt ist.

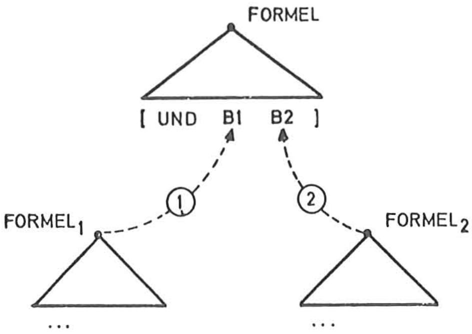


Abbildung 4.13: Konstruktion einer Formel-Konjunktion

Die mit B1 und B2 markierten Knoten können als Leerstellen aufgefaßt werden, in die Formeln einzusetzen sind. Entsteht nun bei einer Überführung neben dem Kerntranslat eine weitere Formel, muß die in Abb. 4.13 dargestellte Struktur erzeugt werden, wobei in die Leerstelle B1 die entstandene Formel (vgl. Pfeil 1 in Abb. 4.13), in die Leerstelle B2 das Kerntranslat, i.e. das Translat von S, einzusetzen ist. Dies wird dadurch erreicht, daß auf die Gesamtstruktur eines Satzes die o.a. Struktur "aufgepfropft" wird (vgl. 4 in Abb. 4.12) und anschließend in die Leerstelle B1 die entstandene Formel - in Abb. 4.12 also das Translat des Adjektivs "untersucht" - eingesetzt wird. Nach Überführung der verbleibenden Teilstrukturen und nach der Translathebung kann eine "Terminierungsregel" (vgl. 4.4.6) angewendet werden, die die von S dominierte Struktur durch das Translat von S ersetzt, so daß ein Baum entsteht, der dem in Abb. 4.13 dargestellten entspricht.

Die Aufpfropfung wird mittels der Regel "Ersetze Struktur durch Literal" (ER.LIT) formuliert unter Verwendung der Strukturvariablen "&", so daß die gesamte Regel für das Adjektiv "untersucht" lautet:

(UND
(CP (ADJ untersucht)
      (FORMEL$_1$ [LABORBERICHT B1 B2 B3 B4]))
(... Leerstellenregel für A1 ...)
(... Leerstellenregel für A2 ...)
(... Leerstellenregel für A3 ...)
(... Leerstellenregel für A4 ...)
(ER.LIT S (FORMEL$_2$ [UND B1 &]))
(SLOT B1 FORMEL$_1$))

4.4.6 Terminierungsregeln

In 3.4 haben wir die von einer Übersetzungsgrammatik erzeugte Sprache L_T als eine Menge von Wörtern definiert, wobei es für jedes Wort α aus L_T einen in der Übersetzungsgrammatik terminal abgeleiteten Baum gibt, dessen Wort gleich α ist (vgl. Def. Ü7). Aufgrund der bisher erläuterten Strategie von ÜG/KS sind die Teile der natürlichsprachlichen Struktur jedoch auch nach Anwendung der Translathebungsregeln noch in einem Baum vorhanden, so daß eine spezielle Regel zu formulieren ist, die die Entfernung dieser natürlichsprachlichen Teile beschreibt. Diese Tilgung kann ganz einfach dadurch erfolgen, daß die von S dominierte Struktur durch das Translat von S ersetzt wird (vgl. die schematische Darstellung der Anwendung einer solchen Regel r in Abb. 4.14).

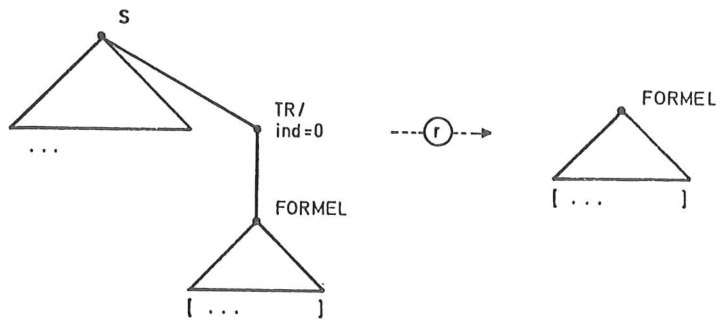


Abbildung 4.14: Anwendung der Terminierungsregel

Da solche Tilgungsregeln die Ermittlung von Translaten zu Teilen von ausgangssprachlichen Strukturen beenden, nennen wir diese Regeln auch Terminierungsregeln.

Bei der Definition der Terminierungsregeln unterscheiden wir drei Fälle: die Terminierung bei Aussagesätzen, bei Fragesätzen und bei Befehlssätzen.

Terminierungsregel bei Aussagesätzen:

r_1 = ((ER.ST S/TYPE=AUSSAGE FORMEL)
(TRANSLAT S FORMEL))

Erläuterung

Durch die Terminierungsregel r_1 wird die gesamte Struktur, deren Wurzelknoten mit einem komplexen Symbol mit der Hauptkategorie S und dem Merkmal TYPE = AUSSAGE ersetzt durch das Translat dieser Struktur (vgl. Abb. 4.14).

Terminierungsregeln bei Fragesätzen

```
r2 = (UND
       ((ER.LIT X1 (FORMEL1 [ ? & ]))
         (UND (ODER (EQ X1 S/TYPE=FRAGE)
                    (DOM* X1 S/TYPE=FRAGE))
              (NICHT (EXIST X2 (DOM X2 X1)))
              (NICHT (DOM* S (ODER WADV WDET WPRON)))))
       ((ER.ST S FORMEL2) (TRANSLAT S FORMEL2)))

r3 = ((ER.ST S/TYPE=FRAGE FORMEL1)
        (UND (TRANSLAT S FORMEL1)
             (DOM* S (ODER WADV WDET WPRON))))
```

Erläuterung

Bei den Terminierungsregeln für Fragesätze gehen wir davon aus, daß Entscheidungsfragen durch eine Formel übersetzt werden, die mit dem pragmatischen Operator "?" präfigiert ist, während Ergänzungsfragen durch LAMBDA-Terme wiederzugeben sind.
Die Regel r_2 ist für Entscheidungsfragen vorgesehen, deren Strukturen daran zu erkennen sind, daß das Merkmal TYPE des komplexen Symbols S mit dem Wert FRAGE belegt ist, in der Struktur jedoch kein WADV, WDET oder WPRON vorkommt. Die Präfigierung der aus der Überführung resultierenden Formel erfolgt durch

Aufpfropfung des Baums der Liste (FORMEL [? &]), wobei der Strukturvariablen "&" aufgrund der Strukturbedingung für X1 die gesamte Struktur zugeordnet wird, auf die die Regel r_2 angewendet wird. D.h. aufgrund der Strukturbedingung (NICHT (EXIST X2 (DOM X2 X1))) wird der Symbolvariablen X1 der Wurzelknoten der Gesamtstruktur zugeordnet, der nicht notwendigerweise mit S markiert sein muß. Dies ist dann nicht der Fall, wenn in vorangehenden Überführungsschritten eine Aufpfropfung auf den mit S markierten Knoten vorgenommen wurde. Nach der Aufpfropfung wird durch die Regel ER.ST die von S dominierte Struktur durch das Translat von S ersetzt.

Die Regel r_3 ist für Ergänzungsfragen vorgesehen, wobei wir davon ausgehen, daß die Konstruktion eines entsprechenden LAMBDA-Terms direkt durch die Überführungsregeln für WADV, WDET und WPRON gesteuert wird. Deshalb kann ohne vorangehende Aufpfropfung die von S dominierte Struktur durch das Translat von S ersetzt werden.

<u>Terminierungsregel bei Befehlssätzen</u>

```
r4 = (UND
       ((ER.LIT X1 (FORMEL1 [! & ]))
         (UND (ODER (EQ X1 S/TYPE=BEFEHL)
                    (DOM* X1 S/TYPE=BEFEHL))
              (NICHT (EXIST X2 (DOM X2 X1)))))
       ((ER.ST S FORMEL2) (TRANSLAT S FORMEL2)))
```

<u>Erläuterung</u>

Die Regel r_4 entspricht im Prinzip der Regel r_2, so daß sich eine weitere Erläuterung erübrigt.

Die Bezeichnung "Terminierungsregel" sollte allerdings nicht zu dem Schluß verleiten, daß die Terminierungsregel die letzte, auf einen Baum anwendbare Regel sein soll. Es ist vielmehr davon auszugehen, daß es noch weitere anwendbare Regeln gibt, mit denen die von einer Terminierungsregel erzeugte Struktur noch weiter verändert werden kann.

5. KOMMENTIERTES BEISPIEL EINER ABLEITUNG IN ÜG/KS

Zur Gesamtillustration der Anwendung von ÜG/KS geben wir in der Folge ein kommentiertes Beispiel einer vollständigen Ableitung in ÜG/KS. Wir gehen dabei aus von der Basisstruktur des Satzes

"Enthielten die Proben bei Lauxmann Cadmium?"

die in Abb. 5.1 dargestellt ist.

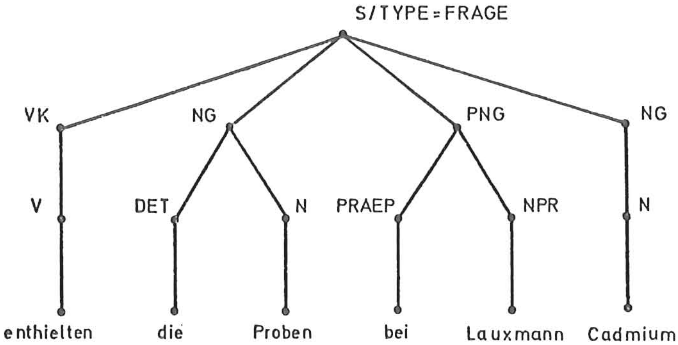


Abbildung 5.1: Basisstruktur für das Ableitungsbeispiel

Die Anwendung der Regeln erfolgt im Beispiel von rechts nach links und "bottom-up". D.h. wir beginnen mit der Anwendung derjenigen Kontextpatternregel, die den am weitesten rechts im Gesamtbaum liegenden terminalen Teilbaum u_1 verändert (in Abb. 5.1 der aus den mit N und Cadmium markierten Knoten bestehende Teilbaum). Ist das Translat von

u_1 ermittelt, wird das Translat desjenigen Teilbaums u_2 ermittelt, dessen Wurzelknoten linker Bruder des Wurzelknotens von u_1 ist (in Abb. 5.1 nicht vorhanden) u.s.f. bis es keinen terminalen Teilbaum mehr gibt, dessen Wurzelknoten mittelbar links vom Wurzelknoten von u_1 liegt. Danach wird auf den nächstgrößeren Teilbaum, d.h. einen terminalen Teilbaum t_1, dessen Wurzelknoten den Wurzelknoten von u_1 dominiert (in Abb. 5.1 mit NG markiert), die Translathebungsregel angewendet. Im nächsten Schritt wird das Translat des Teilbaums t_2 ermittelt, dessen Wurzelknoten (in Abb. 5.1 mit PNG markiert) linker Bruder des Wurzelknotens von t_1 ist, und zwar wiederum beginnend mit dem rechtesten terminalen Teilbaum von t_2, der durch eine Kontextpatternregel veränderbar ist. Auf die Generalität oder den heuristischen Charakter dieses Vorgehens bei der Regelanwendung gehen wir hier nicht weiter ein, da hierzu eine genaue Untersuchung des Zusammenhangs zwischen syntaktischen Strukturen des Deutschen und der Konstruktsprache erforderlich wäre.

Regel für "Cadmium"

(R1) (CP (N Cadmium) (TERM/Sorte=stoff CD))

Anwendung der Regel R1

Entsprechend der Schreibkonvention für die Zuordnung von Kontextpatternregeln (vgl. 4.4.1) wird der terminale Teilbaum von Abb. 5.1, dessen terminaler Knoten mit "Cadmium" markiert ist, zu dem in Abb. 5.2 dargestellten Teilbaum verändert.

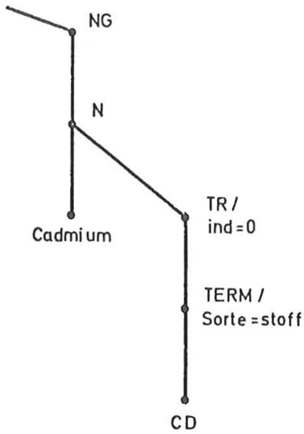


Abbildung 5.2: Anwendung der Regel R1

Regel für "Lauxmann"

(R2) (ODER

(R2.1) (CP (NPR Lauxmann)
(TERM/Sorte=firma G.-Lauxmann))

(R2.2) (UND

(R2.2.1) (CP (NPR Lauxmann)
(TERM/Sorte=betrieb [BETRIEB G.-LAUXMANN A1]))

(R2.2.2) (ODER)
(($SLOT.ATTR$ A1 $TERM_1$/Sorte=ort)
(UND (TRANSLAT PNG_1/K=DAT $TERM_1$)

```
                    (ATTR NPR PNG1)
                    (DOM PNG1 PRAEP)
                    (DOM PRAEP in)))
          (ER.LIT A1 (TERM1/Sorte=ort
                         (EXISTVAR X.ORT))))))
```

Im Sprachausschnitt, den wir hier zugrundelegen, bezeichnet der Eigenname "Lauxmann" nicht nur eine Firma, sondern möglicherweise auch die verschiedenen Betriebe dieser Firma. Diese Mehrdeutigkeit wird durch die alternativen Regeln R2.1 und R2.2 zum Ausdruck gebracht, wobei im zweiten Fall "Lauxmann" übersetzt wird zu einem Term der Sorte "betrieb". Die Leerstelle A1 (vgl. R2.2.2) wird ausgefüllt durch das Translat eines Praepositionalattributs, das den Ort der Firma Lauxmann näher bestimmt wie etwa in der Formulierung "bei Lauxmann in Stuttgart". In der Strukturbedingung der Leerstellenregel SLOT.ATTR für A1 haben wir die Praepositionalattribute eingeschränkt auf PNGs mit der Praeposition "in". Möglicherweise ist diese Einschränkung zu stark und bedarf einer Erweiterung, wenn man Formulierungen zulassen will wie "<u>bei</u> Stuttgart" oder "<u>am</u> Meßpunkt 523". Eine genauere Untersuchung des deutschen Sprachmaterials könnte auch erweisen, daß eine Präzisierung der zulässigen Praepositionen gar nicht erforderlich ist, sondern die Sortenfestlegung "ort" und die syntaktische Bestimmung "Praepositionalattribut" ausreicht.

Die zweite Regel der disjunktiven Regel R2.2.2 ist für den Fall vorgesehen, daß in einem Satz kein Praepositionalattribut zu "Lauxmann" formuliert ist. Die Leerstelle A1 wird dann durch eine KS-Variable ausgefüllt.

Am Beispiel der Regel R2 läßt sich die disambiguierende Funktion der Sorten von KS im ausgangssprachlichen Bereich aufzeigen. Obwohl in einer Formulierung wie "die Proben bei Lauxmann im Jahr 1975" die PNG "im Jahr 1975" die Strukturbedingung der Regel SLOT.ATTR in R2.2.2 erfüllt, kann das Translat dieser PNG, ein Term der Sorte "int", aufgrund der Sortenrestriktion "ort" nicht in die Leerstelle A1 eingesetzt werden.

Regel für die Praeposition "bei"

(R3) (CP (PRAEP bei) ())

Die Praeposition "bei" erhält ein leeres Kontextpattern, da es gem. Tab. 4.3 in dem verwendeten KS-Vokabular kein Prädikat gibt, das dieser Praeposition entspricht. In unserem Kontext wird "bei" nur im Rahmen von Strukturbedingungen zur Präzisierung von Praepositionalattributen und -ergänzungen berücksichtigt. Läßt man allerdings Formulierungen wie "bei Stuttgart" zu, müßte ein KS-Prädikat definiert werden, mit dem sich der Sachverhalt "der Ort X1 liegt in der Nähe von Ort X2" ausdrücken läßt, was dann eine entsprechende alternative Regel zu R3 bedingen würde.

Ein Beispiel für eine differenziertere Behandlung von Praepositionen ist die Praeposition "in", bei der temporaler Gebrauch (vgl. R4.1), lokaler Gebrauch (vgl. R4.2) und sonstiger Gebrauch (vgl. R4.3) unterschieden werden:

```
(R4)    (ODER
(R4.1)   (UND
          (CP (PRAEP in) (TERM1/Sorte=int [IN-TEMP A1])
           (UND (LFT* PRAEP N)
                (TRANSLAT N TERM2/Sorte=int)))
          (SLOT A1 TERM2/Sorte=int))
(R4.2)   (UND
          (CP (PRAEP in) (TERM1/Sorte=ort [IN-LOK A1])
           (UND (LFT* PRAEP N)
                (TRANSLAT N TERM2/Sorte=ort)))
          (SLOT A1 TERM2/Sorte=ort))
(R4.3)    (CP (PRAEP in) ()))
```

An diesem Beispiel läßt sich auch eine andere Disambiguierungsleistung der Sorten von KS aufweisen. In Abhängigkeit der Sorte des Translats des nominalen Nukleus einer PNG mit der Praeposition "in" kann entschieden werden, ob "in" als temporale oder lokale Praeposition verwendet wird. Daß diese Differenzierung rudimentär und noch viel zu unscharf ist, liegt auf der Hand, soll aber hier nicht weiter verfolgt werden.

Anwendung der Regeln R2 und R3

Die Anwendung der Regeln R2 und R3 ist in Abb. 5.3 illustriert. Im Hinblick auf die Ausfüllung der Leerstelle A1 des CP von "Probe" haben wir statt R2.1 die Regel R2.2 angewendet, da im Beispielsatz die Anwendung von R2.1 nicht zu einem terminal abgeleiteten Baum führen würde.

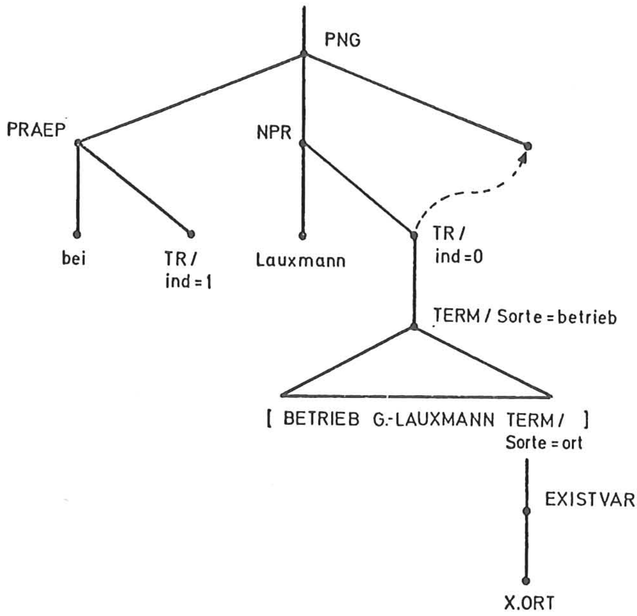


Abbildung 5.3: Anwendung der Regeln R2 und R3; Translathebung

Entsprechend der Schreibkonvention für Kontextpatternregeln mit einem leeren Kontextpattern (vgl. 4.4.1) wird die von PRAEP dominierte Struktur nur um einen mit TR/ind=1 markierten Knoten erweitert. Da der mit PNG markierte Knoten genau einen mit TR/ind=Ø markierten Knoten dominiert, kann die "Translathebungsregel der Form 1" (vgl. 4.4.2) angewendet werden, deren Resultat in Abb. 5.3 angedeutet ist.

<u>Regel für "Probe"</u>

```
(R5)    (UND
(R5.1)  (CP (N₁ Probe) (TERM/Sorte=stoffkoll [PROBE A1 A2]))
(R5.2)  (ODER
          ((SLOT.ATTR A1 TERM₁/Sorte=betrieb)
            (UND (TRANSLAT PNG₁/K=DAT TERM₁)
                 (ATTR N₁ PNG₁)
                 (DOM PNG₁ PRAEP₁)
                 (DOM PRAEP₁ (ODER bei in von))))
          (ER.LIT A1 (TERM₁/Sorte=betrieb
                        (EXISTVAR X.BETRIEB)))

(R5.3)  (ODER
          ((SLOT.ATTR A2 TERM₂/Sorte=int)
            (UND (TRANSLAT PNG₂/K=DAT TERM₂)
                 (ATTR N₁ PNG₂)
                 (DOM PNG₂ Praep₂)
                 (DOM PRAEP₂ (ODER von in))))
          (ER.LIT A2 (TERM₂/Sorte=int (EXISTVAR X.INT)))))
```

<u>Anwendung der Regel R5</u>

Nach Anwendung der CP-Regel R5.1 kann die Regel SLOT.ATTR von R5.2 angewendet werden, die drei Veränderungen bewirkt:

1. Das Translat der PNG "bei Lauxmann" wird in die Leerstelle A1 eingesetzt (vgl. Pfeil 1 in Abb. 5.4).

2. Die Markierung des Knotens, der das eingesetzte Translat dominiert, wird ersetzt durch die Mar-

kierung TR/ind=1 zur Kennzeichnung, daß das Translat eingesetzt wurde.

3. Es wird eine Transformation für Attribute durchgeführt (vgl. 4.4.4), die die NG-Struktur so verändert, daß der mit PNG markierte Knoten rechter Bruder des mit N markierten Knotens wird (vgl. Pfeil 3 in Abb. 5.4). Die PNG wird an ihrem ursprünglichen Platz gelöscht (nicht eingezeichnet in Abb. 5.4).

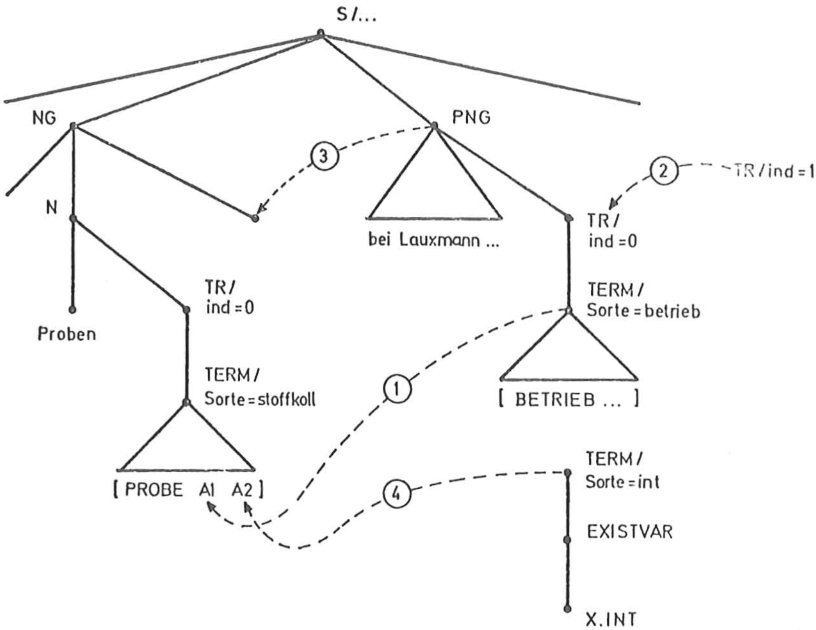


Abbildung 5.4: Anwendung der Regel R5 mit AS-Transformation für Attribute

Die Regel SLOT.ATTR von R5.3 ist nicht anwendbar, da es kein Attribut zu "Probe" gibt, dessen Translat ein Term der Sorte "int" ist. Es wird deshalb mittels der Regel ER.LIT in die Leerstelle A2 die Variable X.INT eingesetzt (vgl. Pfeil 4 in Abb. 5.4).

<u>Regel für "die"</u>

(R6)   (CP (DET/PN=6 die) ())

Da determinierte pluralische Nominalgruppen durch nichtquantifizierte Terme in KS wiedergegeben werden können, erhält der bestimmte Artikel in pluralischer Verwendung keine Entsprechung in KS. Deshalb wird in (R6) ein leeres Kontextpattern zugeordnet.

Die Einfachheit der Regel R6 soll nicht darüber hinwegtäuschen, daß trotz des umfangreichen KS-syntaktischen Inventars wie Jota-Operator, KS-Quantoren (KSQUANT) und Modifikatoren von KS-Quantoren (QUANTMOD) die Überführung von Determinantien bzw. der gesamte Bereich der natürlichsprachlichen Quantifizierung zu den schwierigsten Problemen der Überführung gehört. Eine gründliche Untersuchung dieses Bereichs wird erweisen müssen, ob die gefundenen Regularitäten mit dem bisher vorgeschlagenen Regelinventar darstellbar sind oder ob Erweiterungen erforderlich sind.

<u>Anwendung der Regel R6 und Translathebung</u>

Das Resultat der Anwendung von R6 ist in Abb. 5.5 dargestellt. Nach dieser Regelanwendung kann die "Translathebungsregel der Form 1" angewendet werden (vgl. gestrichelten Pfeil in Abb. 5.5).

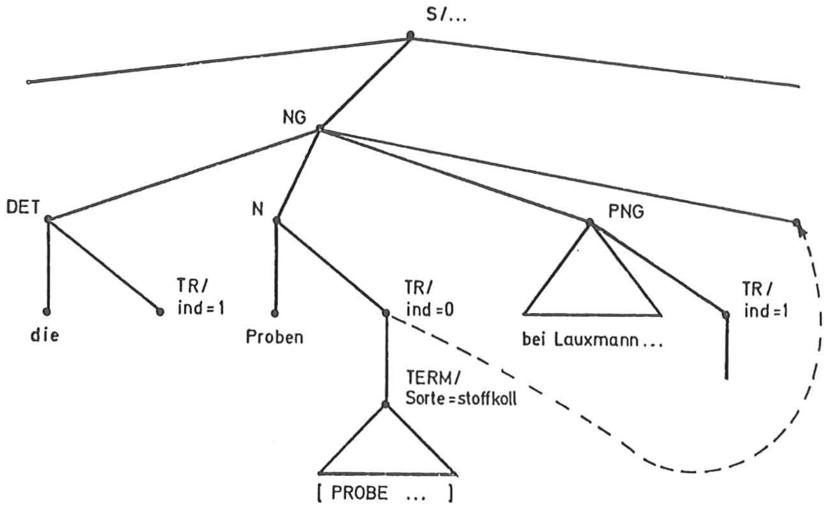


Abbildung 5.5: Anwendung der Regel R6 und Translathebung

Regel für "enthalten"

(R7) (UND

(R7.1) (CP (V enthalten) (FORMEL [ANTEIL A1 A2 A3 A4])

(R7.2) (UND

(SLOT A1 $TERM_1$/Sorte=stoff)

(UND (TRANSLAT N $TERM_1$)

```
                (DOM NG_1 N)
                (ERG4 V NG_1)))
          (ER.LIT NG (NK &)))

(R7.3     ((SLOT.ERG A2 TERM_2/Sorte=stoffkoll)
            (UND (TRANSLAT NG_2 TERM_2)
                 (ERG1 V NG_2)))

(R7.4)    (ODER
(R7.4.1)  ((SLOT.ERG A3 TERM_3/Sorte=abstrobj)
            (UND (TRANSLAT PNG TERM_3)
                 (ERG V PNG)
                 (DOM PNG PRAEP)
                 (DOM PRAEP (ODER gemäß laut entspre-
                                  chend aufgrund))))

(R7.4.2)  (ER.LIT A3 (TERM_3/Sorte=abstrobj
                      (EXISTVAR X.ABSTROBJ))))
(R7.5)    (ODER
(R7.5.1)  ((SLOT A4 TERM_4/Sorte=dimz)
            (UND (TRANSLAT ZAHL TERM_4)
                 (LFT* ZAHL N)

(R7.5.2)  (ER.LIT A4 (TERM_4/Sorte=dimzahl
                      (EXISTVAR X.DIMZAHL)))))
```

Anwendung der Regel R7

Durch die Regel R7.1 wird dem Verb "enthalten" ein Kontextpattern mit 4 Leerstellen zugeordnet (vgl. Abb. 5.6), wobei die erste Leerstelle durch das Translat des Nomens der Akkusativergänzung (R7.2) ausgefüllt wird. D.h. bei einer NG wie "3,5 mg/l Cadmium" wird lediglich das Translat des Nomens "Cadmium" in die Leerstelle A1 eingesetzt, während das Translat von "3,5 mg/l" gem. R7.5 in die Leerstelle A4 einzusetzen ist.

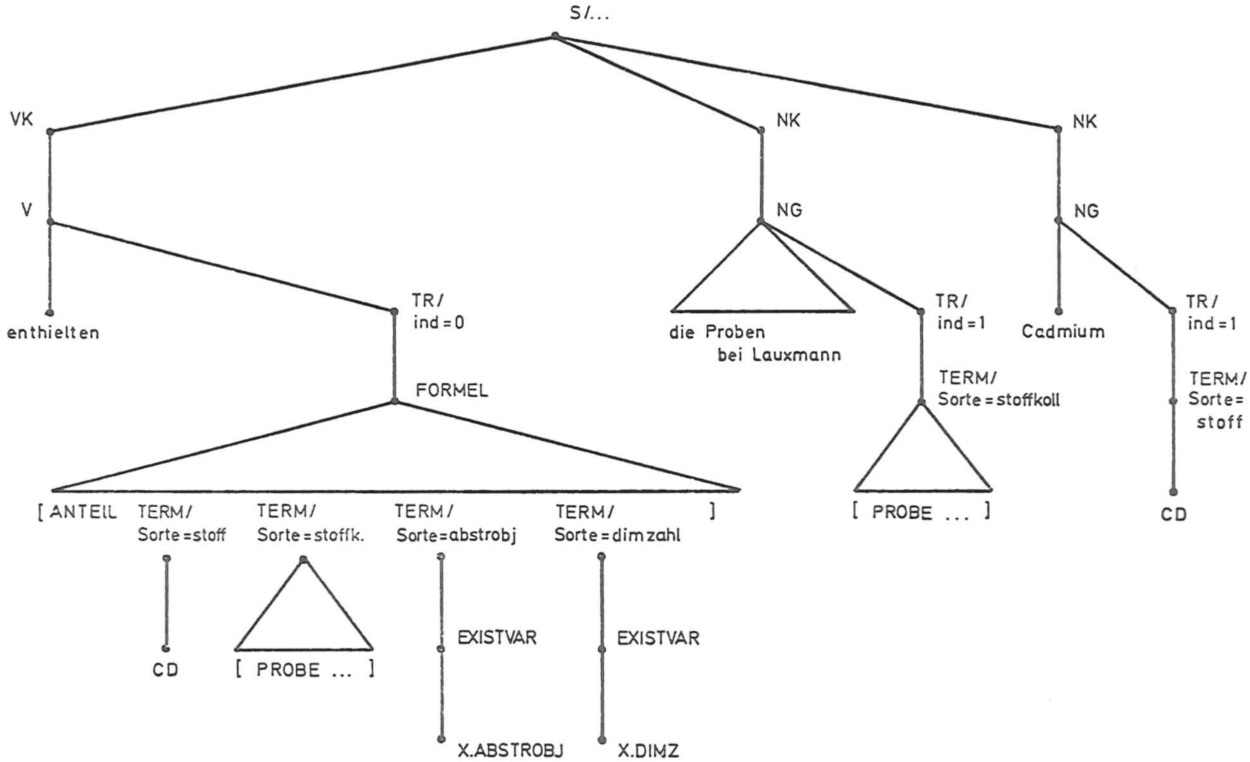


Abbildung 5.6: Anwendung der Regel R6

Da die Schreibkonvention für Leerstellenregeln mit AS-Strukturtransformation für Ergänzungen (vgl. 4.4.4) nur für Translate von NG bzw. PNG definiert ist und nicht für Translate von Nomina, können wir uns einer dieser Schreibkonventionen nicht bedienen. Die Strukturtransformation wird deshalb in einer gesonderten Regel ER.LIT in R7.2 festgelegt.

In der Regel R7.3 wird die Einsetzung des Translats der Nominativergänzung von "enthalten" in die Leerstelle A2 beschrieben.

In den Regeln R7.4 und R7.5 für die Leerstellen A3 und A4 ist vorgesehen, daß die entsprechenden natürlichsprachlichen Formulierungen der gesuchten Translate im deutschen Satz nicht ausformuliert sind. So fehlt etwa im zugrunde gelegten Beispielsatz sowohl eine PNG mit einer der Praepositionen "gemäß", "laut", "entsprechend", "aufgrund" sowie die Konstituente ZAHL in der NG_1, die von den Strukturbedingungen der Regel R7.2 näher spezifiziert wurde. In diesem Fall werden die Leerstellen gem. der Regel R7.4.2 und R7.5.2 durch Variable ausgefüllt.

Das Ergebnis der Anwendung von R7 ist in Abb. 5.6 illustriert.

<u>Anwendung der Terminierungsregel</u>

Wendet man auf die in Abb. 5.6 dargestellte Struktur mehrfach die "Translathebungsregel vom Typ1" an, entsteht die Struktur von Abb. 5.7.

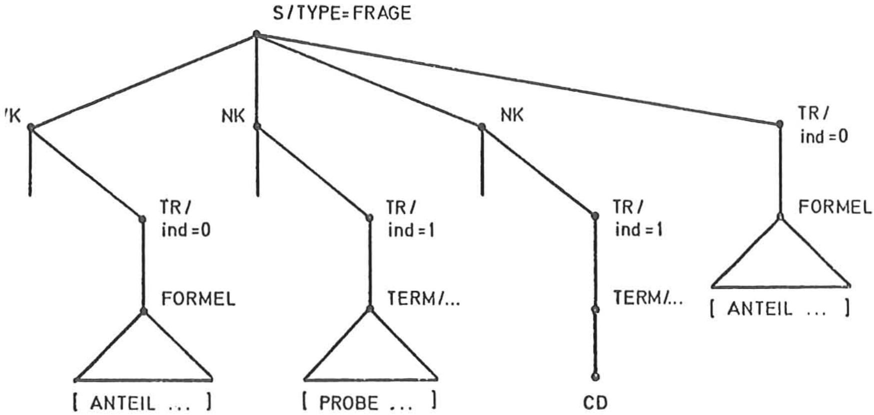


Abbildung 5.7: Beispielstruktur nach Anwendung der Translathebungsregeln.

Da es sich beim Beispielsatz um eine Entscheidungsfrage handelt, wird durch die Anwendung der entsprechenden Terminierungsregel (vgl. 4.4.6) zuerst eine Struktur mit dem pragmatischen Operator "?" aufgepfropft (vgl. Abb. 5.8).

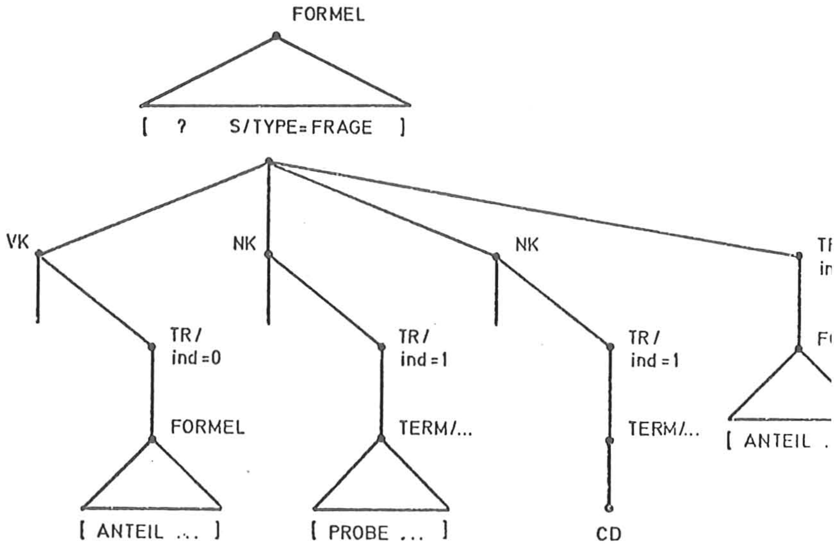


Abbildung 5.8: Anwendung der ersten Teilregel der Terminierungsregel für Entscheidungsfragen.

Danach erst wird die von S dominierte Struktur durch das Translat von S ersetzt, woraus die in Abb. 5.9 dargestellte Struktur entsteht.

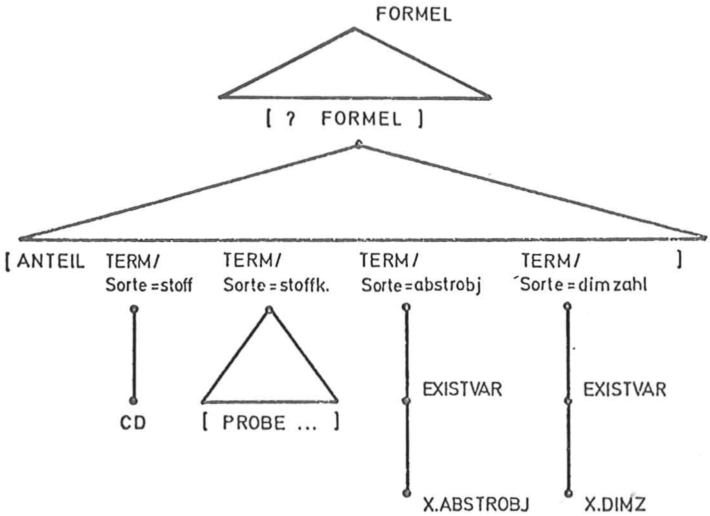


Abbildung 5.9: Resultierende Struktur nach Anwendung der Terminierungsregel

Regel für "EXISTVAR"

```
(R8)    (UND
(R8.1)   ((ER.LIT FORMEL1 (QUANTORFOR  EXIST A1 & ))
            (UND (DOM* FORMEL1 EXISTVAR)
                 (ODER (UND (DOM FORMEL2 FORMEL1)
                            (DOM FORMEL2 (ODER ? !)))
                       (DOM* FORMEL2 FORMEL1))))
(R8.2)   ((ER.ST A1 X1)  (DOM EXISTVAR X1))
(R8.3)   (ER.S EXISTVAR KSVAR))
```

Erläuterung

Zur Bindung von Variablen durch den Quantor EXIST wird auf die Formel, in der eine zu bindende Variable vorkommt, eine Struktur aufgepfropft (vgl. R8.1 bzw. Pfeil 1 in Abb. 5.10), die den Quantor enthält sowie eine Leerstelle, in die die zu bindende Variable einzusetzen ist (vgl. Pfeil 2 in Abb. 5.10). In den Strukturbedingungen von R8.1 schlägt sich folgendes Vorgehen nieder:

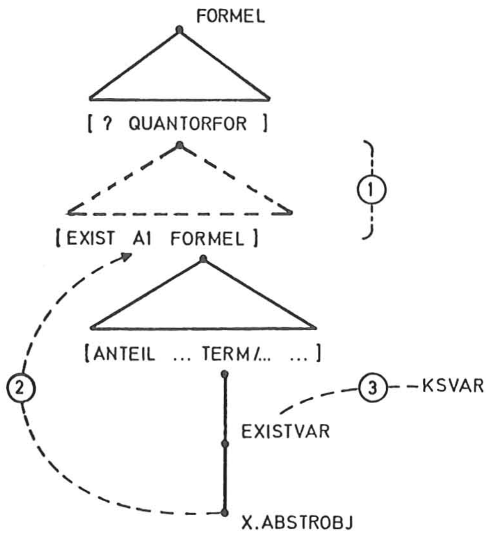


Abbildung 5.10: Beispiel einer Variablenbindung durch Anwendung der Regel R8.

1. Die Quantoren sollen nach dem pragmatischen Operator "?" oder "!" stehen.

2. Bei nichtatomaren Formeln, i.e. bei Formeln, die mehrere atomare Formeln enthalten, sollen die Quantoren am Anfang der gesamten Formel stehen.

Dies wird dadurch erreicht, daß die durch R8.1 aufzupfropfende Struktur auf die "höchste" Formel im Baum aufgepfropft wird, die keinen pragmatischen Operator enthält.
Um eine mehrfache Quantifizierung einer Variablen durch mehrfache Anwendung von R8 zu verhindern, wird durch R8.3 das Symbol EXISTVAR ersetzt durch das Symbol KSVAR (vgl. Pfeil 3 in Abb. 5.10).

Anwendung der Regel R8

Die Regel R8 kann auf die in Abb. 5.6 dargestellte Struktur viermal angewendet werden, da neben den Variablen X.ABSTROBJ und X.DIMZ der PROBE-Term noch die Variablen X.ORT und X.INT enthält (vgl. Abb. 5.3 und 5.4), so daß die Struktur von Abb. 5.11 entsteht.

Auf die Struktur von Abb. 5.11 sind keine weiteren Regeln mehr anwendbar, so daß das Wort dieses Baumes ermittelt werden kann,

```
[? [EXIST X.ORT [EXIST X.INT [EXIST X.ABSTROBJ
                              [EXIST X.DIMZ
   [ANTEIL CD
           [PROBE [BETRIEB G.-LAUXMANN X.ORT] X.INT]
            X.ABSTROBJ
            X.DIMZ]]]]]]
```

das in Übersetzungsrelation steht zu dem Satz: "Enthielten die Proben bei Lauxmann Cadmium?"

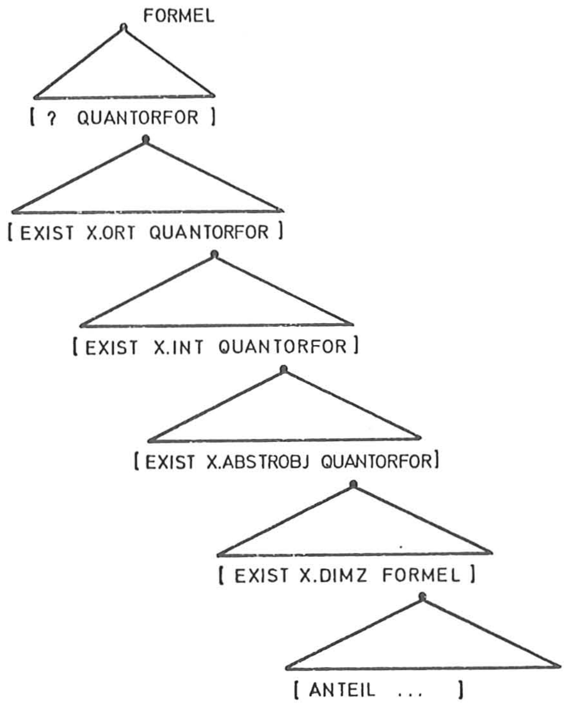


Abbildung 5.11: Terminal abgeleitete Struktur

6. ANMERKUNGEN ZU EINER THEORETISCHEN FUNDIERUNG

Im Rahmen des Systems PLIDIS wurde ein Transduktor programmiert, der von den Regeln einer Übersetzungsgrammatik gesteuert wird und dessen Übersetzungsstrategie der Ableitung in einer Übersetzungsgrammatik folgt, wie sie hier skizziert wurde. Damit mag zwar ein erster Nachweis der Praktikabilität unserer Vorschläge erbracht sein, eine theoretische Fundierung wird dadurch jedoch nicht ersetzt, die uns insbesondere zur Klärung der in Kapitel 1 aufgeworfenen Fragen erforderlich erscheint. Überlegungen zu einer solchen Fundierung sind in zwei Richtungen voranzutreiben: zum einen im Hinblick auf eine Theorie des Formalismus einer Übersetzungsgrammatik, zum andern unter dem Gesichtspunkt einer Theorie der Anwendung eines solchen Formalismus zur Beschreibung von Übersetzungszusammenhängen. Unter letzterem verstehen wir die Ausarbeitung der Bedingungen, unter denen ein Übersetzungszusammenhang mit dem vorgeschlagenen Formalismus beschrieben werden kann, bzw. die Explikation der Hypothesen, die mit der Anwendung einer Übersetzungsgrammatik bezüglich des Übersetzungsvorgangs gemacht werden.

Für die Anwendung einer Übersetzungsgrammatik sind jedoch auch die formalen Eigenschaften des Formalismus insofern von Bedeutung, als sie Rückschlüsse zulassen über die Art der von einer Übersetzungsgrammatik erzeugten Sprache und über Verfahren zur Algorithmisierung der Ableitung in einer Übersetzungsgrammatik. Da uns die Untersuchung der formalen Eigenschaften einer Übersetzungsgrammatik im

Rahmen der vorliegenden Arbeit nicht leistbar erscheint, wollen wir uns in der Folge auf einige Hinweise beschränken, wie die formalen Eigenschaften von ÜG bewiesen werden könnten.

6.1 Transformationsgrammatiken

Beim Definitionsgang unserer Übersetzungsgrammatik haben wir uns über weite Strecken an den Definitionen des Formalismus einer Transformationsgrammatik von GINSBURG/PARTEE (1969) sowie BRESZTOWSZKI et al. (1974) orientiert. Allerdings weichen die Definitionen unserer Regeln und ihrer Anwendung von denen einer Transformationsgrammatik ab. Es erscheint uns jedoch möglich, nachzuweisen, daß unsere Regeln äquivalent sind zu denen einer Transformationsgrammatik, so daß auf diesem Weg bewiesen werden könnte, daß unsere Übersetzungsgrammatik ein Spezialfall einer Transformationsgrammatik ist. Dies hätte den Vorteil, daß die formalen Eigenschaften einer Übersetzungsgrammatik nicht eigens zu beweisen wären, sondern daß für eine Übersetzungsgrammatik diejenigen Sätze gelten, die für Transformationsgrammatiken bewiesen wurden. Hierzu liegt eine umfangreiche Untersuchung von PAUSE (1976) vor, in der auch die Bedingungen für die Umkehrbarkeit von Transformationsgrammatiken untersucht werden, was für die Übersetzungsgrammatik von besonderem Interesse ist.

Für den Fall, daß die Rückführbarkeit unserer Übersetzungsgrammatik auf eine Transformationsgrammatik bzw. die Äquivalenz der beiden Formalismen nachweisbar ist, bedeutet das nicht, daß unsere Übersetzungsgrammatik überflüssig ist. Wir nehmen vielmehr an, daß die Beschreibung von Regularitäten wie etwa bei der Übersetzung deutsch/KS im Formalismus einer Transformationsgrammatik zu derart umfangreichen und komplizierten Regeln führen würde, daß eine Anwendung in der Praxis kaum oder nur unter Schwierigkeiten in Frage käme. Dies müßte allerdings im Detail nachgewiesen werden.

6.2 Schemata für Programmiersprachenübersetzung

Betrachtet man Transformationsgrammatiken - losgelöst von ihrem linguistischen Anwendungszweck - als formale Systeme zur Beschreibung von Baummanipulationen, ergibt sich eine Verbindung zu einem Forschungsbereich, dessen Ergebnisse nach unserem Überblick von einer mehr linguistisch orientierten Forschung bisher kaum beachtet und rezipiert wurden: die Beschreibung von Programmiersprachenübersetzern (Compiler). Vereinfacht vollzieht sich der Übersetzungsvorgang in gängigen Compilern (vgl. etwa GRIES (1971)) in drei mehr oder weniger scharf getrennten Schritten: eine zu übersetzende Symbolkette wird zuerst segmentiert (Scanning) in Einzelsymbole und dann syntaktisch analysiert (Parsing). Aufgrund der im Parsing gewonnenen Informationen wird dann der zielsprachliche Code erzeugt (code generation), der von der Rechenanlage ohne zusätzliche größere Umformungsschritte berechnet werden kann.

Während lange Zeit Compiler nur durch Darstellung ihres Aufbaus und Erläuterung ihrer spezifischen Programmierung beschrieben wurden, wurde in der zweiten Hälfte der sechziger Jahre verstärkt damit begonnen, Compiler durch theoretische Modelle zu erklären und Compilertheorien zu entwickeln (vgl. etwa AHO/ULLMAN (1973)). Die Rolle der Baummanipulation im Zusammenhang mit der Beschreibung von Compilern sieht SCHREIBER (1976) so:

> "Theoretische Hilfsmittel zur Konstruktion und Untersuchung des Analyseteils eines Compilers sind Grammatiken und zu ihnen äquivalente Automaten. Mit diesen elementaren automatentheoretischen Hilfsmitteln lassen sich jedoch nur die lexikalische und die syntaktische Analyse geeignet untersuchen. Für die schwierigeren Teile eines Compilers, beginnend bei der semantischen Analyse, gibt es keine geeigneten Hilfsmittel aus der klassischen Automatentheorie. Da die klassische Automatentheorie auf der Verarbeitung von Zeichenketten, also linearen -eindimensionalen- Strukturen basiert und in den der syntaktischen Analyse folgenden Phasen Bäume verarbeitet werden, kann die klassische Automatentheorie keine geeigneten Modelle zur Untersuchung jener Phasen bereitstellen. Erst seit einigen Jahren beginnt sich die Auffassung durchzusetzen, daß man die Übersetzung von Programmiersprachen nicht als einen zeichenkettenmanipulierenden, sondern als einen baumverarbeitenden Prozeß zu sehen hat - also als einen Prozeß, bei dem Bäume in Bäume übersetzt werden." (p. 118)

Von besonderem Interesse für die Klärung der formalen Eigenschaften unserer Übersetzungsgrammatik als Beschreibung von baumverarbeitenden Prozessen erscheint uns die Untersuchung des Zusammenhangs zwischen unserer Übersetzungsgrammatik und den von ROSEN (1973) beschriebenen Teilbaumersetzungssystemen sowie zu den, teilweise auf der Basis von ROSEN, in CHAUCHÉ (1974) und SCHREIBER (1976) entwickelten Formalismen zur Baumverarbeitung. (Eine Fülle weiterer Literatur zur Baumverarbeitung findet sich in CHAUCHÉ und SCHREIBER.)

6.3 Konkurrierende Ansätze

Zur Bewertung des von uns vorgeschlagenen Formalismus müßte nach einer theoretischen Fundierung eine Abgrenzung und ein Vergleich zu konkurrierenden Ansätzen erfolgen.

In diesem Zusammenhang ist auf einen bisher noch nicht erwähnten Ansatz aus dem Bereich der Programmiersprachenübersetzung einzugehen: den von KNUTH (1968) vorgeschlagenen Formalismus einer Attributsgrammatik, der von SCHWIND (1977) verwendet wurde zur Beschreibung des Übersetzungszusammenhangs zwischen deutschen Sätzen und "zustandslogischen Formeln". In einer Attributsgrammatik werden zu einer kontextfreien Grammatik G = (VN, VT, S, P) zu allen Symbolen des Alphabets Attribute und mögliche Werte dieser Attribute definiert, die nicht zu verwechseln sind mit den Attributen von komplexen Symbolen. Zu jeder Regel $p \in P$ der Form $X_o \rightarrow X_1 \ldots X_n$ wird nun eine Menge F_p von "semantischen Regeln" definiert (vgl. KNUTH (1968), p. 132), wobei

$$F_p = \left\{ f_{i\alpha} \,\middle|\, \begin{array}{l} (1)\ 0 \leq i \leq n \\ (2)\ \alpha \text{ ist ein Attribut von } X_i \end{array} \right\} .$$

Eine semantische Regel $f_{i\alpha}$ ist eine Funktion zur Berechnung des Wertes des Attributs α, wobei die Berechnung abhängig gemacht werden kann von Werten anderer Attribute von $X_o, \ldots, X_n$, d.h. $f_{i\alpha}$ kann eine mehrstellige Funktion sein. Dies bedeutet, daß Attribute von X_i, $i > \emptyset$, in Abhängigkeit von X_o berechnet werden, d.h. von der linken zur rechten Regelseite "weitergegeben" werden können bzw. umgekehrt. Um nun mit Hilfe der semantischen Regeln die "Bedeutung" bzw. in unserem Kontext die Übersetzung eines in G terminal abgeleiteten Wortes zu bestimmen, werden die Attributsfunktionen von KNUTH in Verbindung mit Ableitungsbäumen gebracht, was SCHWIND (1977) wie folgt beschreibt:

> "Durch Attributfunktionen werden ... Werte von Attributen in einem Ableitungsbaum von Knoten zu Knoten weitergereicht, und zwar in zwei Richtungen: Von der Wurzel zu den Blättern (i.e. die terminalen Knoten; d.Verf.) und von den Blättern zu der Wurzel. Daher gibt es zwei Arten von Attributen: abgeleitete (derived), die Werte von den Blättern zur Wurzel transportieren und ableitende (inherited), die Werte von der Wurzel zu den Blättern befördern. Zu jeder Produktion und zu jedem abgeleiteten (derived) Attribut, das zu einem Alphabetelement auf der linken Seite der Produktion gehört, gibt es dann eine Attributfunktion, die Werte von Attributen, die zu Alphabetelementen auf der rechten Seite der Produktion gehören, abbildet auf einen Wert, der zur Wertemenge jenes abgeleiteten Attributs gehört, und Wert dieses Attributs für denjenigen Knoten ist, der mit dem entsprechenden Alphabetelement der linken Seite markiert ist. Ebenso gibt es zu jedem ableitenden (inherited) Attribut, das zu einem Alphabetelement auf der rechten Seite der Produktion gehört, eine Attributfunktion, die Werte von Attributen anderer in derselben Produktion vorkommender Alphabetelemente abbildet auf einen Wert aus der Wertemenge jenes ableitenden Attributs; und dieser Wert ist dann wieder der Wert des ableitenden Attributs für jenen Knoten."
> (p. 51 f.)

Es bleibt bei KNUTH ebenso wie bei SCHWIND offen, ob es allgemeine Typen von Attributen und Funktionen gibt, die über eine einzelne von einer Attributsgrammatik beschriebene Sprache hinaus Gültigkeit haben.

Ein Zusammenhang zwischen einer Attributsgrammatik und unserer Übersetzungsgrammatik besteht nun insofern, als in einer Attributsgrammatik eine Alternative aufgezeigt werden könnte für die Behandlung der von uns als Translate bezeichneten Strukturen, wenn diese nicht wie bei uns in eine Basisstruktur integriert, sondern als Werte eines Attributs TRANSLAT berechnet werden. Ebenso könnten Teile der Strukturbedingungen als Attributsberechnungen umformuliert werden. Soweit wir jedoch die von SCHWIND gemachten Vorschläge übersehen können, scheint dieser Formalismus vor allem dann zu sehr komplexen und umfangreichen Regelmengen zu führen, wenn die Konstruktion von zielsprachlichen Strukturen starke strukturbildende Operationen erfordert - wie bei ÜG/KS die Ausfüllung von Leerstellen und die Aufpropfung - und nicht wie bei SCHWIND hauptsächlich durch Verknüpfung einfacher Strukturen oder gar Aneinanderreihung von Symbolketten bewältigt werden kann. Diese möglicherweise überspitzte Formulierung bedürfte jedoch noch gründlicherer Untersuchung.

Eine andere Alternative zu unseren Vorschlägen haben wir bereits in Kapitel 1 erwähnt, die von PRATT (1971) dargestellte "Pair-Grammar", die wir jedoch in Zusammenhang mit den "Parallel Graph Grammars" von EHRIG/KREOWSKI (1976) sehen und zur Klasse der Parallelgrammatiken rechnen (vgl. hierzu auch Kap. IV, "Parallel Graph Generating and Related Systems" in LINDENMAYER/ROZENBERG (1976)). Durch die Verbindung von Parallelgrammatiken mit Graphensprachen entsteht ein Formalismus, der auf den ersten Blick ähnliche Anwendungen zuläßt wie unsere Übersetzungsgrammatik:

> "A pair grammar is composed of a pair of grammars whose rules and nonterminals are paired. The pair grammar defines a correspondence between elements of the languages defined by the two grammars. This correspondence may be viewed as a definition of the translation of the elements of one language into the other...
>
> A graph grammar defines a language composed of a set of directed graphs. Pair grammars are constructed from pairs of graph grammars. Each unambiguous pair grammar defines a reversible function mapping one graph language onto another. Special cases of interest include string-to-graph, graph-to-string, and string-to-string mappings. In the general case a pair grammar defines a transformation on a set of graphs".
>
> (PRATT (1971), p. 560).

Vereinfacht dargestellt entsteht bei einer Pair-Grammar für ein Sprachenpaar L_1, L_2 und den entsprechenden Alphabeten V_1, V_2 ein Wortpaar (w_1', w_2') aus einem Wortpaar (w_1, w_2) durch Anwendung einer Paarregel (r_1, r_2), wobei w_1' aus w_1 durch Anwendung von r_1 und w_2' durch Anwendung von r_2 aus w_2 entsteht.

Aufgrund der Anwendungsbeispiele von PRATT aus dem Bereich der Programmiersprachen sowie der Beschreibungen von Parallelgrammatiken im Kontext der Zellbiologie (vgl. LINDENMAYER/ROZENBERG (1976)) können wir nicht einmal annähernd beurteilen, inwieweit diese Vorschläge geeignet sind, komplexe Übersetzungszusammenhänge wie etwa die zwischen natürlichen Sprachen und formalen Sprachen zu beschreiben. Bei der Untersuchung einer solchen Eignung müßte insbesondere auch geklärt werden, mit welcher Ausschließlichkeit Parallelgrammatiken ein top-down Vorgehen bei der Erzeugung einer Übersetzung festlegen. Da Parallelgrammatiken ähnlich wie kontextfreie Grammatiken von einem Startsymbol (bzw. einem Startknoten) ausgehen, das sukzessive durch Regelanwendung zu einer terminalen Struktur abgeleitet wird, entstehen Schwierigkeiten, wenn - in der Terminologie von Bäumen gesprochen - bei der Generierung von Teilen des zielsprachlichen Baumes, die in der Nähe des Wurzelknotens liegen, Regelanwendungen davon abhängig sind, welche terminalen Teilstrukturen später "weiter unten" im ausgangssprachlichen Baum erzeugt werden.

Dieses Problem, "daß nämlich in einem frühen Stadium der Übersetzung eine Entscheidung getroffen werden muß, die von der gesamten zu übersetzenden Struktur abhängt, wobei diese beliebig groß sein kann" (BROCKHAUS (1971), p. 60),taucht besonders bei natürlichen Sprachen auf, wie die Arbeit von BROCKHAUS (1971) zeigt. Soweit wir den stark programmierungsorientierten Formalismus von BROCKHAUS verstehen, scheint BROCKHAUS gewisse spätere oder gleichzeitige Entwicklungen aus dem Bereich der Parallelgrammatiken vorweggenommen zu haben. Die bei diesem Vorgehen entstehenden oben skizzierten Probleme versuchte BROCKHAUS mit "Semantemen und semantischen Funktionen" (BROCKHAUS, pp. 62) zu lösen, die eine gewisse Ähnlichkeit mit der Verwendung von Attributen und Attributsfunktionen von KNUTH (1968) haben. Es wäre sicherlich interessant, die Überlegungen von BROCKHAUS auf dem Hintergrund von Attributsgrammatiken und Parallelgrammatiken zu reformulieren und weiterzuentwickeln als eine top-down Alternative zu dem von unserer Übersetzungsgrammatik ermöglichten bottom-up Verfahren.

Es mag überraschen, daß wir im Laufe unserer Arbeit auf keine Überlegungen aus den Forschungen zur "Künstlichen Intelligenz" (KI) Bezug genommen haben (zu einem Überblick über die sprachorientierte KI-Forschung vgl. WULZ/ZIFONUN (1978)). In diesem Bereich gibt es zahlreiche Arbeiten, die formale Repräsentationsmittel für natürlichsprachlich formulierte Informationen zum Gegenstand haben sowie die Programmierung von Systemen, die aufgrund natürlichsprachlicher Eingaben Fragen beantworten oder, genereller ausgedrückt, natürlichsprachlich formulierte Problemstellungen lösen sollen. (Zu Einzeldarstellungen vgl. RUSTIN (1973), SCHANK/COLBY (1973), CHARNIAK/WILKS (1976)). Obwohl in solchen Systemen aus natürlichsprachlichen Formulierungen eine systeminterne Repräsentationsform erzeugt werden muß, fehlt es nach unserer Übersicht bisher an theoretischen Darstellungen der für diesen Erzeugungsprozeß erforderlichen Mittel und Methoden. Damit wollen wir nicht etwa behaupten, daß es an expliziten Vorgehensbeschreibungen fehlt. Solche Beschreibungen liegen sowohl von WINOGRAD (1971), WOODS et al. (1972), RIESBECK (1974) und WILKS (1975) vor, um die Bekanntesten aufzuzählen. Dabei handelt es sich in erster Linie um Erläuterungen von Programmen oder Erklärungen von Beispielen, nicht jedoch um eine theoretische Explikation der zugrundeliegenden axiomatischen Systeme. Auf diesen Mangel an Abstraktion und theoretischer Erklärung scheint uns auch die Kritik von HAYES (1975) zur "4th Int. Joint Conference on Artificial Intelligence" abzuheben:

> "There is a lot of natural language work, much of it organized around whole systems. It was often my impression that people were, unfortunately, more concerned with defending their systems than in analysing its relationships to other systems, or describing it in a sufficiently non-idiosyncratic way as to make such analysis possible. ... It is still, evidently, possible to get papers published which put forward an entirely <u>new</u>, completely <u>universal</u>, system for representing <u>all</u> of human knowledge- and which make

the exciting new observation that the comprehension of natural language depends upon having such a representation (or even, this one in particular) available". (pp. 19)

In gewisser Weise erscheint uns die Situation, in der sich die KI-Forschung zur Erzeugung formaler Darstellungen für natürlichsprachliche Formulierungen befindet, vergleichbar zu sein mit der Forschungsphase der Programmiersprachenübersetzer vor Ende der sechziger Jahre, in der zwar zahlreiche Compiler konstruiert und exemplarisch beschrieben wurden, eine Erklärung durch Compilertheorien jedoch noch ausstand. Insofern sind die bisher bekannt gewordenen Vorschläge aus dem KI-Bereich zur Übersetzung natürlichsprachlicher Formulierungen in formale Sprachen weniger konkurrierende Ansätze zu unserer Übersetzungsgrammatik als vielmehr Prüfsteine, ob mit unserem Formalismus bzw. mit welchen Modifikationen Vorgehensweisen bei einzelnen KI-Systemen beschreibbar sind.

ANHANG A: DEFINITION DER VERWENDETEN GRUNDBEGRIFFE UND SYMBOLE

In den folgenden Abschnitten sind die Grundbegriffe festgelegt, mit deren Hilfe die Formalismen einer Übersetzungsgrammatik expliziert wurden. Die Bestimmungen zu den grundlegenden mathematischen Begriffen (Abschnitt A.1) und zu Alphabeten und Grammatiken (Ab schnitt A.2) wiederholen Bekanntes und dienen ledigli zur Verdeutlichung unseres Sprachgebrauchs sowie zur Vermeidung von Mißverständnissen. Für ausführlichere Darstellungen verweisen wir auf WALL (1972) sowie auf KRATZER et al. (1973).

A.1 Grundlegende mathematische Begriffe

A.1.1 Mengen

Wollen wir festlegen, daß eine Menge M aus den Objekt a, b, c besteht, dann schreiben wir

$$M = \{a,b,c\}$$

d.h. wir zählen die Menge M auf. Wollen wir eine Meng nicht durch Aufzählung festlegen sondern durch die An gabe von Eigenschaften der Objekte in dieser Menge, dann schreiben wir

$$M = \{\, x \mid x \text{ hat die Eigenschaft } P \,\}$$

wobei P irgendeine definierte Eigenschaft sei.

Enthält eine Menge M keine Objekte, dann bezeichnen w M als die <u>leere Menge</u> und schreiben

$$M = \emptyset$$

Ist $M = \{a_1, a_2, \ldots, a_n\}$, dann heißen die Objekte a_i Elemente von M und wir schreiben

$$a_i \in M$$

wir sagen dann auch, a_i liegt in M. Ist kein Mißverständnis möglich, dann schreiben wir auch

$$a_1, a_2, \ldots, a_n \in M$$

anstatt $a_1 \in M$, $a_2 \in M$, ..., $a_n \in M$.

Ist ein Objekt x kein Element der Menge M, schreiben wir

$$x \notin M$$

Sind N und M Mengen, so daß die Elemente von N auch Elemente von M sind, dann sagen wir, N ist eine Teilmenge von M, geschrieben

$$N \subseteq M \quad \text{oder} \quad M \supseteq N$$

Ist N Teilmenge von M und gibt es mindestens ein Element von M, das nicht Element von N ist, dann ist N eine echte Teilmenge von M, geschrieben

$$N \subset M \quad \text{oder} \quad M \supset N$$

Sind die Elemente von N auch Elemente von M und gibt es kein Element von M, das nicht auch Element von N ist, d.h. gilt $N \subseteq M$ und $M \subseteq N$, dann heißen die Mengen M und N gleich, geschrieben

$$N = M$$

Sind zwei Mengen nicht gleich, schreiben wir

$$N \neq M$$

Für die <u>Vereinigung</u> von zwei Mengen schreiben wir

$M \cup N$

mit der Festlegung

$M \cup N = \{x \mid x \in M \text{ oder } x \in N\}$

d.h. die Vereinigung von zwei Mengen ist diejenige Menge, in der sowohl die Elemente von M als auch die Elemente von N liegen.

<u>Bemerkung</u>

Im Rahmen von Definitionen benutzen wir "oder" als <u>einschließendes</u> "oder". D.h. am Beispiel

$x \in M \quad \text{oder} \quad x \in N$

erklärt: Es muß einer der drei Fälle gelten:

$x \in M \quad \text{und} \quad x \notin N$
$x \notin M \quad \text{und} \quad x \in N$
$x \in M \quad \text{und} \quad x \in N$

Für das <u>ausschließende</u> "oder", für das genau einer der beiden ersten Fälle gelten darf, schreiben wir "entweder ... oder ..." wie z.B

entweder $x \in M$ oder $x \in N$

Für den <u>Durchschnitt</u> von zwei Mengen M und N schreiben wir:

$M \cap N$

mit der Festlegung

$M \cap N = \{x \mid x \in M \text{ und } x \in N\}$

d.h. der Durchschnitt der Mengen M und N ist diejenige Menge, deren Elemente sowohl in M als auch in N liegen.

Ist der Durchschnitt zweier Mengen M und N leer, d.h. gilt $M \cap N = \emptyset$, dann heißen M und N <u>disjunkte</u> Mengen; m.a.W. M und N haben keine gemeinsamen Elemente.

Die <u>Differenz</u> zweier Mengen M und N, geschrieben

$$M \setminus N$$

mit der Festlegung

$$M \setminus N = \{x \mid x \in M \text{ und } x \notin N\}$$

ist die Menge, die diejenigen Elemente von M enthält, die nicht Elemente von N sind.

Das (<u>kartesische</u>) <u>Produkt</u> zweier Mengen M und N, geschrieben

$$M \times N$$

mit der Festlegung

$$M \times N = \{(x,y) \mid x \in M \text{ und } y \in N\}$$

ist diejenige Menge, die als Elemente (geordnete) Paare enthält, deren erste Komponente in M und deren zweite Komponente in N liegt.

Für das Produkt von n Mengen $M_1, M_2, \ldots, M_n$ gilt:

$$M_1 \times M_2 \times \ldots \times M_n = \{(x_1, x_2, \ldots, x_n) \mid x_i \in M_i \text{ für } i \leq n\}$$

Ein Element des Produkts von n Mengen nennen wir auch ein <u>n-Tupel</u>.

<u>Bemerkung</u>

Wollen wir beliebige Objekte x_i, $i \leq n$, aufzählen, dann schreiben wir $x_1,\ldots,x_n$. Im Gegensatz dazu schreiben wir ein n-Tupel a mit den Komponenten x_i mit runden Klammern: $a = (x_1,\ldots,x_n)$. Während bei der Aufzählung von Objekten die Reihenfolge unerheblich ist, ist in einem Tupel aufgrund des Zusammenhangs mit einem Produkt von Mengen die Reihenfolge der Komponenten festgelegt. Soll jedoch bei einer Aufzählung ausgedrückt werden, daß Objekte $x_1,\ldots,x_n$ in der Reihenfolge der Aufzählung geordnet sind, dann sagen wir auch "die Folge $x_1,\ldots,x_n$", was gleichbedeutend ist mit "das n-Tupel $(x_1,\ldots,x_n)$".

Die Menge, die als Elemente alle Teilmengen einer Menge M enthält, bezeichnen wir als <u>Potenzmenge</u> von M, geschrieben

$\mathbb{P}(M)$

mit der Festlegung

$\mathbb{P}(M) = \{N \mid N \text{ ist eine Teilmenge von } M\}$

Das Symbol $\mathbb{N}$ verwenden wir zur Bezeichnung der Menge der natürlichen Zahlen:

$\mathbb{N} = \{\ 1,\ 2,\ 3,\ \ldots\ \}$

A.1.2 Relationen und Funktionen

Relationen definieren wir als Teilmengen von Produkten von Mengen. Ist $R \subseteq M_1 \times M_2$ eine Relation, dann sagen wir:

(1) R ist eine <u>Relation in</u> M, wenn $M_1 = M_2$;
(2) R ist eine <u>Relation zwischen</u> M_1 und M_2, wenn $M_1 \neq M_2$;
(3) R trifft auf (a,b) zu bzw. (a,b) erfüllt R, wenn $(a,b) \in R$.

Ist $R \subseteq M_1 \times M_2 \times \ldots \times M_n$ eine Relation, dann bezeichnen wir R auch als n-stellig.

Eine Relation P in einer Menge M ist der transitive Abschluß einer Relation R in M , wenn für alle Paare $(a,b) \in P$ gilt:

> es gibt Paare (x_1,x_2), $(x_2,x_3),\ldots,$ (x_{n-1},x_n), die Elemente von R sind, und $a = x_1$ sowie $b = x_n$.

Zur Bezeichnung des transitiven Abschlusses einer Relation R schreiben wir auch R^*.

Funktionen oder Abbildungen sind Spezialfälle von Relationen. Eine Funktion φ von einer Menge M auf (bzw. in) eine Menge N, geschrieben

$$\varphi : M \longrightarrow N$$

ist bestimmt durch eine Relation $F \subseteq M \times N$, wobei gilt: für jedes $a \in M$ gibt es genau ein $b \in N$, so daß $(a,b) \in F$. Für $(a,b) \in F$ schreiben wir

$$\varphi(a) = b$$

und nennen $\varphi(a)$ den Funktionswert von a.

A.1.3 Logische Ausdrücke

Zur Spezifizierung von Mengen und zur Formulierung von Bedingungen in Definitionen verwenden wir die Verknüpfungen der Aussagenlogik und die Quantoren der Prädikatenlogik mit der üblichen wahrheitsfunktionalen Festlegung. Soweit es uns sinnvoll erscheint, stellen wir unsere Definitionen nicht durchgehend in streng logischer Notation dar, sondern verwenden dazu vor allem natürlichsprachliche Paraphrasen, die - soweit als möglich - in jeweils vorangehenden Definitionen vereinbart wurden.

Logische Verknüpfungen

Für die logischen Verknüpfungen verwenden wir folgende Notation:

Konjunktion:	$\wedge$	und
Disjunktion:	$\vee$	oder
Negation	$\neg$	nicht; es gilt nicht
Konditional	$\longrightarrow$	wenn ... dann
Bikonditional	$\longleftrightarrow$	genau dann wenn; gdw.

Quantoren

Die Quantoren verwenden wir in folgender Darstellung:

Allquantor	$\forall$	(für) alle
Existenzquantor	$\exists$	es gibt (ein)
	$\exists!$	es gibt genau ein

Bei Quantifizierungen verwenden wir folgende Abkürzungen:

statt $(\forall x)\ x \in M \ldots$ schreiben wir $(\forall x \in M)\ldots$
$(\forall x_i)\ i \leq n \ldots$ $(\forall x_i,\ i \leq n)\ldots$
$(\forall i)\ i \leq n \ldots$ $(\forall i \leq n)$

Die gleichen Schreibkonventionen gelten auch für Existenzquantifizierungen.

Bei Konjunktionen, die in Definitionen einzelne Bedingungen $B_1, B_2, \ldots, B_n$ verknüpfen, schreiben wir statt $B_1 \wedge B_2 \wedge \ldots \wedge B_n$ auch

(1) B_1
(2) B_2
...
(n) B_n

Soweit es der Übersichtlichkeit und Lesbarkeit dient, führen wir Klammerungen ein, wo dies nicht erforderlich wäre. So schreiben wir z.B.

$(a \in A) \quad (b \in B) \quad (c \in C)$ statt $a \in A \quad b \in B \quad c \in C$.

A.2 Alphabete und Grammatiken

A.2.1 Wörter und Alphabete

Ein <u>Alphabet</u> (Vokabular, Zeichenvorrat) ist eine endliche Menge von Symbolen (Zeichen).

Ein <u>Wort</u> (Kette) über einem Alphabet ist eine endliche Folge von Symbolen aus einem Alphabet. Das leere Wort ist dasjenige Wort, das aus keinem Symbol besteht. Wir bezeichnen das leere Wort mit ε.

Ist V ein Alphabet, dann bezeichnen wir mit V^* die Menge aller Wörter, die aus Symbolen von V gebildet sind, eingeschlossen das leere Wort ε. Mit V^+ bezeichnen wir die Menge $V^* \setminus \{\varepsilon\}$.

Sind $a_1,\ldots,a_n$ Symbole aus V und ist α ein Wort aus V^*, das aus der Folge $a_1,\ldots,a_n$ besteht, dann schreiben wir $\alpha = (a_1,\ldots,a_n)$ oder einfacher auch $\alpha = a_1\ldots a_n$.

Ist α ein Wort mit $\alpha = a_1\ldots a_n$, dann heißt ein Kette $a_i a_{i+1}\ldots a_{i+j}$ mit $i \leq n$ und $i+j \leq n$ <u>Teilwort</u> von α. Ist das Wort δ die Verkettung der Teilwörter α, β, γ, dann schreiben wir auch $\delta = \alpha\beta\gamma$.

Ist $\alpha = a_1\ldots a_n$ ein Wort, dann sagen wir auch, a_i ist ein <u>Symbol von</u> α oder a_i ist in α <u>enthalten</u> für $i \leq n$.

Im Zusammenhang mit Bäumen, deren Knoten mit Symbolen markiert sind, ist es in bestimmten Fällen erforderlich, sich mit Hilfe der Symbole auf die Knoten zu beziehen. Können mehrere Knoten mit dem gleichen Symbol markiert sein, ist der Bezug nicht mehr eindeutig. Es werden deshalb indizierte Symbole eingeführt:

<u>Def.:</u> (indiziertes Alphabet, indiziertes Symbol) (A1)

Sei A ein Alphabet; sei $\mathbb{N}$ die Menge der natürlichen Zahlen.

Wir nennen die Menge

$$V = A \times \mathbb{N}$$

das <u>indizierte</u> <u>Alphabet</u> <u>von</u> A und schreiben dann für V auch $A^{(\mathbb{N})}$.

Ein Paar $(a,i) \in A^{(\mathbb{N})}$ nennen wir <u>indiziertes</u> <u>Symbol</u> und schreiben für (a,i) auch $a^{(i)}$ bzw. a_i, wenn kein Mißverständnis möglich ist (z.B.: NP_1, VP_1 statt $NP^{(1)}$, $VP^{(1)}$).

<u>Def.:</u> (Indexbeseitigung) (A2)

Sei $A = V \cup V^{(\mathbb{N})}$ ein Alphabet.

Eine <u>Indexbeseitigung</u> ist eine Abbildung

$$\varphi: A \longrightarrow V$$
$$\varphi(x) = y \quad \text{wobei} \quad (1)\ y = a \text{ wenn } x = a^{(i)}$$
$$(2)\ y = x \text{ wenn } x \in V$$

φ wird erweitert auf A^* durch die Festlegung

$$\varphi(x_1 \ldots x_n) = \varphi(x_1) \ldots \varphi(x_n) \quad .$$

<u>Erläuterung</u>

Durch die Indexbeseitigung wird einem indizierten Symbol das ihm entsprechende nichtindizierte Symbol zugeordnet, während einem nichtindizierten Symbol das gleiche nichtindizierte Symbol zugeordnet wird.

A.2.2 Grammatiken

In der Folge verstehen wir unter einer Grammatik ein Regelwerk, zu dem ein Verfahren angebbar ist, mit dem die formal richtigen Formulierungen (Sätze) einer Sprache aufgezählt (erzeugt) werden können. Solche Grammatiken werden deshalb auch als Regelgrammatiken oder generative Grammatiken bezeichnet. Da diese Grammatiken nur die formal richtigen Formulierungen einer Sprache beschreiben, findet sich auch die Bezeichnung Syntax (vgl. KRATZER et al. 1973).
Daneben finden sich Namen wie CHOMSKY-Grammatiken, Phrasenstrukturgrammatiken oder Konstituentenstrukturgrammatiken. Aufgrund ihrer Eigenschaften lassen sich diese Grammatiken in verschiedene Typen einteilen (vgl. CHOMSKY 1959). Wir beschränken uns in der Folge im wesentlichen auf kontextfreie Grammatiken ohne auf andere Typen einzugehen.

<u>Def.:</u> (kontextfreie Grammatik) (A3)

Eine <u>kontextfreie Grammatik</u> G ist ein Viertupel

$$G = (VN, VT, S, P)$$

mit folgenden Komponenten:

VN und VT sind Alphabete mit $VN \cap VT = \emptyset$, wobei
VN das <u>nichtterminale Vokabular von</u> G
VT das <u>terminale Vokabular von</u> G

S ist ein ausgezeichnetes Symbol aus VN, das <u>Startsymbol von</u> G

P ist eine Relation, $P \subseteq VN \times A^*$ mit $A = VN \cup VT$, die (Produktions-)Regeln von G. Für ein Paar $(a,\beta) \in P$ wird auch geschrieben $a \longrightarrow \beta$. Im Hinblick auf die Verwendung der Regeln wird $a \longrightarrow \beta$ auch gelesen als "ersetze a durch β".

Zur Bezeichnung der Alphabete von G verwenden wir auch die Schreibweise VN(G), VT(G) und A(G), wobei

$$A(G) = VN(G) \cup VT(G)$$

das Alphabet von G heißt.

Allgemeiner linguistischer Terminologie folgend bezeichnen wir die Symbole aus VN auch als syntaktische Kategorien, die Symbole aus VT auch als Lexeme.

Def.: (Überführung, Ableitung) (A4)

Sei G = (VN,VT,S,P) eine kontextfreie Grammatik; seien $\alpha,\alpha_1,\ldots,\alpha_n,\beta,\gamma$ Wörter über A(G); sei a ein Symbol aus VN.

Das Wort $\alpha a \gamma$ heißt direkt überführbar (in G) in das Wort $\alpha\beta\gamma$, geschrieben $\alpha a \gamma \vdash_G \alpha\beta\gamma$, gdw.:

es gibt eine Produktionsregel $p \in P$, so daß $p = a \longrightarrow \beta$

Eine Folge $\alpha_1,\ldots,\alpha_n$ ist eine Überführung von α_1 nach α_n (in G), gdw.:

für alle $i \leq (n-1)$ gilt:
$\alpha_i \vdash_G \alpha_{i+1}$

Ein Wort α ist (in G) überführbar in β, geschrieben $\alpha \vdash_G^* \beta$, gdw.:

es gibt eine Überführung von α nach β in G.

Eine Überführung $\alpha_1, \ldots, \alpha_n$ in G ist eine Ableitung (in G), gdw.:

$\alpha_1 = S$

Ein Wort α heißt terminal abgeleitet (in G), gdw.:

es gibt eine Ableitung $\alpha_1, \ldots \alpha_n$, so daß gilt:
(1) $\alpha = \alpha_n$
(2) $\alpha \in VT^+$

Wir sagen dann auch, α ist ein terminales Wort (von G).

Def.: (Sprache) (A5)

Sei G = (VN,VT,S,P) eine kontextfreie Grammatik.

Eine Wortmenge L, $L \subseteq A(G)^*$, heißt die von G erzeugte Sprache oder kürzer die Sprache von G, gdw:

L ist die größte Teilmenge von $A(G)^*$, so daß
für alle $\alpha \in L$ gilt:
α ist ein terminales Wort von G

d.h. L ist die Menge der Wörter aus VT^*, die vom Startsymbol S terminal abgeleitet sind.

Ist L(G) eine natürliche Sprache bzw. eine formale Sprache, dann nennen wir die Elemente von L(G) auch Sätze bzw. Formeln.

Komplexe Notation für nichtterminale Symbole

Beim Schreiben der Regeln für eine kontextfreie Grammatik kann das praktische Problem auftauchen, daß Teilmengen von Regeln entstehen, die untereinander sehr ähnlich sind und die man im Interesse der Beschreibungsökonomie und der Durchschaubarkeit zu wenigen Regeln bzw. genaugenommen zu einem Regelschema zusammenfassen möchte. D.h. an einem Beispiel erklärt:

Sollen in einer kontextfreien Grammatik des Deutschen Nominalphrasen (NP) unterschieden werden nach Numerus, Genus und Kasus - auf den Sinn einer solchen Unterscheidung wollen wir hier nicht eingehen -, wäre eine größere Zahl von nichtterminalen Symbolen (Kategorien) für eine solche Unterscheidung erforderlich wie etwa:

NPNOMSF	für "NP im Nominativ,		Singular,	feminin"	
NPNOMPF	"NP	"	Plural	"	"

usw.

Dies wirkt sich auch entsprechend auf die Zahl der Regeln aus, wie etwa:

NPNOMSF $\longrightarrow$ DETNOMSF NNOMSF
für "ersetze NPNOMSF durch ein Determinans im Nominativ, Singular, feminin und ein Nomen im Nominativ, Singular, feminin".

NPNOMPF $\longrightarrow$ DETNOMPF NNOMPF

usw.

Durch Einführung einer "komplexen Notation" (vgl. BROCKHAUS (1971) pp 74, sowie KRATZER et al. (1974) pp 5) können solche "verwandten" Symbole und Regeln zusammengefaßt werden. Dazu werden zu einem nichtterminalen Symbol Mengen von Attribut-Wert-Paaren definiert, wie etwa:

Attribut	Wert
KASUS	1 (für Nominativ)
	2 (für Genitiv)
	...
GENus	MASkulin
	FEMinin
	...
NUMerus	SINGular
	PLURal

so daß die Symbole für "Nominalphrase im Singular ..." usw. geschrieben werden können als

NP/KASUS=1,GEN=FEM,NUM=SING
NP/KASUS=1,GEN=FEM,NUM=PLUR

Man findet für NP auch die Bezeichnung "Hauptkategorie", für Attribut "Merkmal" und für Wert "Unterkategorie" (vgl. BROCKHAUS a.a.O.).

Führt man Variable für die möglichen Werte eines Attributs ein, dann kann man die o.a. Produktionsregeln sowie die dazu zu ergänzenden in dem Regelschema

NP/KASUS=i,GEN=j,NUM=k $\longrightarrow$ DET/KASUS=i,GEN=j,NUM=k
N/KASUS=i,GEN=j,NUM=k

zusammenfassen.

Wir beschränken uns hier auf eine rein notationelle Betrachtungsweise dieser Vorschläge und untersuchen im Rahmen diser Arbeit nicht näher, welche formalen Grundlagen dieser Notation zugrunde zu legen sind. Wir verweisen dazu auf BRAUN (1971) sowie auf die Anwendung von BRAUN in SCHWIND (1977).

Festlegung

Wird ein Symbol a notiert als Zeichenfolge

$$A/m_1=w_1,\ldots,m_n=w_n$$

dann nennen wir A eine Hauptkategorie (von a), m_i ein Attribut oder ein Merkmal (von a) und w_i den (Attributs- oder Merkmals-) Wert von m_i, i = n.

Statt "das Symbol a" sagen wir dann auch verkürzt: "das (komplex notierte) Symbol A" oder "das komplexe Symbol A".

A.3 Bäume, Listen, Ableitungen

Zur Darstellung von Ableitungen werden häufig Baumdiagramme verwendet.

Beispiel:

Sei G = (VN,VT,S,P) eine kontextfreie Grammatik mit

VN = S,NP,VP,DET,N,V,PRAEP, ...
VT = Hund, Hof, der, bellt, im, ...
P = S $\longrightarrow$ NP VP,
NP $\longrightarrow$ DET N,
NP $\longrightarrow$ PRAEP N,
VP $\longrightarrow$ V NP,
N $\longrightarrow$ Hund,
N $\longrightarrow$ Hof,
DET $\longrightarrow$ der,
V $\longrightarrow$ bellt,
PRAEP $\longrightarrow$ im,
...

(Es dürfte klar sein, daß es sich hier um eine sehr naive Grammatik handelt, die lediglich Beispielzwecken dienen soll!)

Die Ableitung des terminalen Wortes α = "der Hund bellt im Hof" kann dargestellt werden mit dem Baumdiagramm in Abb. A.1. Man bezeichnet einen solchen Baum auch als (Konstituenten-)Strukturbaum von α bzw. Phrasemarker, Struktur(beschreibung), Ableitungsgraph u.ä.

In der Literatur werden verschiedene Verfahren beschrieben, die Ableitungen in Zusammenhang bringen mit Baumdiagrammen (vgl. WALL (1972) pp. 214; KRATZER et al. (1973) pp. 91). Von WALL werden auch die Bedingungen aufgezeigt, unter denen Ableitungsbäume für Ableitungen in kontextsensitiven Grammatiken konstruiert werden können.

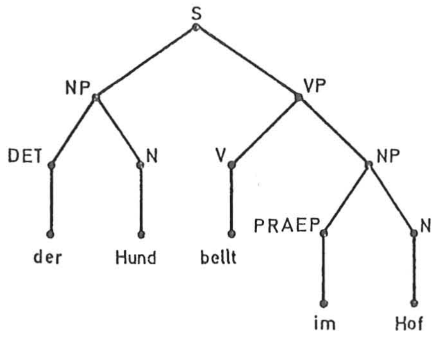


Abbildung A.1: Ableitungsbaum

Für bestimmte Zwecke ist es erforderlich, Baumdiagramme zu linearisieren. Eine mögliche Form für die Linearisierung des Baums von Abb. A.1 ist der Klammerausdruck:

```
(S(NP(DET der)(N Hund))(VP(V bellt)(NP(PRAEP im)
(N Hof))))
```

bzw. in einer besser lesbaren Schreibweise, die die Struktur der Klammerung besser erkennen läßt:

```
(S
   (NP
       (DET der)
       (N Hund))
   (VP
       (V bellt)
       (NP
           (PRAEP im)
           (N Hof))))
```

Einen solchen Klammerausdruck nennen wir Liste oder, wenn er eine Ableitung beschreibt, Ableitungsliste.

Die Ableitungslisten zu einer gegebenen Grammatik G sind Wörter über einem Alphabet $V = A(G) \cup \{(,)\}$ und es liegt nahe, in Abhängigkeit von G eine Grammatik G' zu definieren, deren terminale Wörter Ableitungslisten von Ableitungen in G sind.

In der Folge gehen wir so vor, daß wir zuerst die Begrifflichkeit für Bäume und Listen definieren, um danach den Begriff der Ableitungslisten und der sie erzeugenden Grammatiken zu präzisieren.

A.3.1 Bäume

In den folgenden Definitionen wird auf eine ausführliche Explikation des Baumbegriffs mit graphentheoretischer Fundierung verzichtet. Umfassendere Darstellungen finden sich in KNUTH (1968a) sowie in der dort besprochenen Literatur. Bei unseren Definitionen folgen wir mit einigen Modifikationen BRESZTOWSZKI et al. (1974).

Def.: (Baum) (A6)

Ein Baum ist ein 6-Tupel

$u = (Kn, Ma, x_o, D, L, M)$

mit folgenden Komponenten:

Kn ist eine Menge von Knoten, die Knotenmenge von u

Ma ist eine Menge von Symbolen, die Markierungsmenge oder Markierungen von u

x_o ist ein ausgezeichnetes Element aus Kn, der Wurzelknoten von u

D ist eine Relation in Kn, die Dominanzrelation von u

L ist ebenfalls eine Relation in Kn, die Linksorientierung von u

M ist eine Relation zwischen Ma und Kn, die Markierungsrelation von u.

Zur Bezeichnung der Komponenten von u schreiben wir für Kn, Ma, x_o, D, L, M auch Kn(u), Ma(u), $x_o(u)$, D(u), L(u), M(u).

Wir führen folgende Terminologie ein:

gilt:	dann sagen wir auch:
$(k_1,k_2) \in D(u)$	k_1 dominiert k_2 (in u) (direkt)
$(k_1,k_2) \in L(u)$	k_1 liegt links von k_2 (in u)
	k_2 liegt rechts von k_1 (in u)
	k_1 bzw. k_2 ist linker bzw. rechter Bruder von k_2 bzw. k_1
$(a,k) \in M(u)$	a ist die Markierung von k
	k ist mit a markiert.

Def.: (Dominanzrelation) (A7)

Sei u ein Baum; sei Kn die Knotenmenge von u; sei x_o der Wurzelknoten von u.

Eine Relation $D \subseteq Kn \times Kn$ ist die Dominanzrelation von u, geschrieben D(u), gdw.:

für alle Knoten k, k_1, $k_2 \in$ Kn gilt:

(1) $(k_1,k) \in D \wedge (k_2,k) \in D \longrightarrow k_1 = k_2$

d.h. zwei verschiedene Knoten können nicht denselben Knoten dominieren.

(2) $(k,x_o) \notin D$

d.h. es gibt keinen Knoten, der den Wurzelknoten dominiert.

(3) $(k_1,k_2) \in D^* \longrightarrow (k_2,k_1) \notin D^*$

wobei

$$D^* = \{(k_1,k_2) \mid (\exists\, x_1,\dots,x_n \in Kn)$$
$$(\forall i,\ 2 \leq i \leq n)$$
$$(1)\ k_1 = x_1$$
$$(2)\ k_2 = x_n$$
$$(3)\ (x_{i-1},x_i) \in D\} \quad \text{für } n = 2$$

$$(4)\ k \neq x_o \longrightarrow (x_o,k) \in D^*$$

Die Relation D^* von (3), i.e. der transitive Abschluß von D, nennen wir <u>mittelbare</u> <u>Dominanzrelation</u> und sagen, ein Knoten k_1 <u>dominiert</u> den Knoten k_2 <u>mittelbar</u> (<u>in</u> u), gdw.:

$$(k_1,k_2) \in D^*$$

<u>Erläuterung</u>

Bedingung (3) legt mit Hilfe der mittelbaren Dominanzrelation fest, daß ein Baum "schleifenfrei" sein muß, d.h. ein Knoten k_1 kann nicht einen Knoten k_2 mittelbar dominieren und gleichzeitig wieder von k_2 mittelbar dominiert werden.

(4) schreibt vor, daß alle Knoten eines Baumes, die ungleich dem Wurzelknoten sind, von diesem mittelbar dominiert werden.

<u>Def.:</u> (Linksorientierung) (A8)

Sei u ein Baum; sei Kn die Knotenmenge von u; sei x_o der Wurzelknoten von u; sei D die Dominanzrelation von u.

Eine Relation $L \subseteq Kn \times Kn$ ist die <u>Linksrelation</u> <u>von</u> u, geschrieben L(u), gdw.:

für alle Knoten k, k_1, $k_2 \in Kn$ gilt:

(1) $(k_1,k) \in L \wedge (k_2,k) \in L \longrightarrow k_1 = k_2$

d.h. zwei verschiedene Knoten können nicht gleichzeitig (unmittelbar) links von einem gleichen Knoten liegen.

(2) $(k_1,k_2) \in L \longrightarrow (k_2,k_1) \notin L$

d.h. wenn ein Knoten k_1 links von einem Knoten k_2 liegt, dann kann k_2 nicht gleichzeitig auch links von k_1 liegen.

(3) $(k_1,k_2) \in L^* \longrightarrow (\exists\ k)\ (k,k_1) \in D \wedge (k,k_2) \in D$

wobei L^* der transitive Abschluß von L

d.h. wenn ein Knoten mittelbar links von einem anderen Knoten liegt, dann gibt es genau einen Knoten, der beide dominiert.

<u>Def.:</u> (totale Linksorientierung) (A9)

Sei u ein Baum; sei Kn die Knotenmenge von u.

Eine Relation $L_{total} \subseteq Kn \times Kn$ ist die <u>totale Linksoriehtierung von</u> u, gdw.:

für alle $(x,y) \in L_{total}$ gilt:
es gibt Knoten k_1, $k_2 \in Kn$, so daß

(1) $(k_1,k_2) \in L^*$

(2) $(k_1,x) \in D^* \vee k_1 = x$

(3) $(k_2,y) \in D^* \vee k_2 = y$

<u>Erläuterung</u>

Mit Hilfe der mittelbaren Dominanzrelation und der mittelbaren Linksorientierung L^* werden in der totalen Linksorientierung Knotenpaare zusammengefaßt, die entweder in L^* liegen oder aber deren erste Komponente von einem Knoten mittelbar dominiert wird, der mittelbar links von einem Knoten liegt, der die zweite Komponente des Paares mittelbar dominiert.

Beispiel:

Der Baum in Abb. A.2 ilustriert die Definition am Beispiel eines Knotenpaares $(x,y) \in L_{total}$.

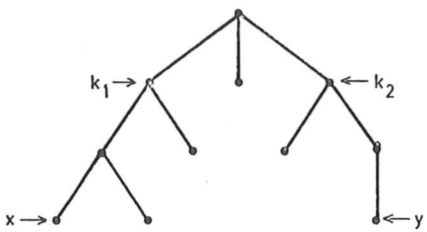


Abbildung A.2

Def.: (Markierungsrelation) (A10)

Sei u ein Baum; sei Kn die Knotenmenge von u; sei Ma die Markierungsmenge von u.

Eine Relation $M \subseteq Ma \times Kn$ ist die Markierungsrelation von u, geschrieben M(u), gdw.:

(1) für alle Knoten $k \in Kn$ gilt:
$(!\exists a)\ (a,k) \in Ma$
d.h. zu jedem Knoten gibt es genau eine Markierung.

(2) für alle Markierungen $a \in Ma$ gilt:
$(\exists k \in Kn)\ (a,k) \in Ma$
d.h. zu jeder Markierung gibt es mindestens einen Knoten, der mit dieser Markierung markiert ist.

Def.: (terminaler, nichtterminaler, verzweigender Knoten; Kante, Vater, Sohn, Geschwister) (A11)

Sei u ein Baum; sei Kn die Knotenmenge von u; sei D die Dominanzrelation von u.

Ein Knoten $k \in Kn$ ist ein terminaler Knoten (von u), gdw.:

$(\forall k_1 \in Kn)\ (k,k_1) \notin D$

d.h. es gibt keinen Knoten, den k dominiert.

k ist ein nichtterminaler Knoten (von u), gdw.:

$(\exists k_1 \in Kn)\ (k,k_1) \in D$

d.h. es gibt einen Knoten, der von k dominiert wird.

k ist ein verzweigender Knoten (von u), gdw.:

$(\exists k_1, k_2 \in Kn)\ (k,k_1) \in D \wedge (k,k_2) \in D \wedge k_1 \neq k_2$

d.h. es gibt mindestens zwei Knoten, die von k dominiert werden.

Ein Paar (k_1,k_2) mit $k_1,k_2 \in Kn$ ist eine Kante (von u), gdw.:

$(k_1,k_2) \in D$

Zwei Kanten (k_1,k_2) und (t_1,t_2) von u heißen zusammenhängend, gdw.:

$k_1 = t_1 \vee k_2 = t_1 \vee t_2 = k_1$

Ein Knoten $k_1 \in Kn$ ist Vater von $k_2 \in Kn$ (in u) bzw. k_2 ist Sohn von k_1 (in u), gdw.:

$(k_1,k_2) \in D$

Ein Knoten k_1 ist (linker bzw. rechter) Bruder von k_2

(<u>in</u> u), gdw.:

$$(k_1,k_2) \in L \quad \text{bzw.} \quad (k_2,k_1) \in L$$

Zwei Knoten $k_1,k_2 \in Kn$ sind <u>Geschwister</u>(<u>knoten</u>) (<u>in</u> u), gdw.:

$$(k_1,k_2) \in L^* \vee (k_2,k_1) \in L^*$$

<u>Def.:</u> (leerer Baum) (A12)

Einen Baum u nennen wir den <u>leeren Baum</u>, gdw.:

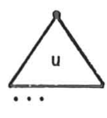
$Kn(u) = \emptyset$

d.h. die Knotenmenge des leeren Baumes ist die leere Menge.

<u>Zeichnerische Darstellung von Bäumen</u>

Die zeichnerische Darstellung eines Baumes, der durch das Tupel $u = (Kn,Ma,x_o,D,L,M)$ gegeben ist, erscheint uns klar zu sein, so daß wir auf die Angabe eines Zeichenverfahrens verzichten.

Wollen wir einen beliebigen Baum u darstellen, ohne seine Struktur näher zu spezifizieren, dann verwenden wir die in Abb. A.3 dargestellte Form.



<u>Abbildung A.3:</u> Darstellung eines unspezifizierten Baumes u.

Def.: (Baumausschnitt, Teilbaum) (A13)

Seien u und v Bäume.

v ist ein Baumausschnitt (Zweig) von u, gdw.:

(1) $Kn(v) \subseteq Kn(u)$

d.h. die Knotenmenge von v ist eine Teilmenge der Knotenmenge von u;

(2) $Ma(v) \subseteq Ma(u)$

d.h. die Markierungsmenge von v ist eine Teilmenge der Markierungsmenge von u;

(3) $D(v) = D(u) \cap (Kn(v) \times Kn(v))$

d.h. die Dominanzrelation von v enthält diejenigen Paare von D(u), deren Komponenten Knoten von v sind;

(4) $L(v) = L(u) \cap (Kn(v) \times Kn(v))$

d.h. die Linksrelation von v enthält diejenigen Paare von L(u), deren Komponenten auch Knoten von v sind;

(5) $M(v) = M(u) \cap (Ma(v) \times Kn(v))$

d.h. die Markierungsrelation von v enthält diejenigen Paare aus M(u), deren erste Komponente eine Markierung von v und deren zweite Komponente ein Knoten von v ist;

(6) $(\forall k_1, k_2 \in Kn(v))$

$((\exists k_3 \in Kn(u))\ (k_3, k_2) \in L(u) \wedge$

$(k_1, k_2) \in L^*(v)) \longrightarrow k_3 \in Kn(v)$

d.h. wenn ein Knoten k_1 mittelbar links von k_2 im Baumausschnitt liegt und wenn es in u einen Knoten k_3 gibt, der links von k_2 in u liegt, dann liegt k_3 auch im Baumausschnitt v.

v ist ein Teilbaum von u, gdw.:

(1) v ist ein Baumausschnitt von u

(2) $(\forall k_1, k_2, k_3 \in Kn(u))$
$(k_1, k_2) \in D(v) \wedge (k_1, k_3) \in D(u)$
$\longrightarrow k_3 \in Kn(v)$

d.h. liegt ein Knoten k_2 im Teilbaum v und wird er von einem Knoten k_1 in v dominiert, dann liegen auch diejenigen Knoten k_3 im Teilbaum v, die von k_1 in u dominiert werden. M.a.W. liegt ein Knoten von u im Teilbaum v von u, dann liegen auch die Geschwisterknoten von u im Teilbaum v.

Erläuterung

Ein Baumausschnitt v eines Baumes u kann aus beliebigen, in u zusammenhängenden Knoten bestehen mit der Einschränkung,daß die zwischen zwei Geschwisterknoten liegenden Knoten ebenfalls zum Baumausschnitt gehören, falls die beiden Geschwisterknoten im Baumausschnitt liegen. Deshalb ist der Baum v_1 von Abb. A.4 ein Baumausschnitt von u, während dies für v_2 dann nicht gilt, wenn die gestrichelten Kanten mit den Knoten C und H wegfallen.

Für einen Teilbaum v von u gilt zusätzlich die Einschränkung, daß - verkürzt gesagt - nicht nur die in u zwischen Geschwisterknoten von v liegenden Knoten im Teilbaum liegen müssen, falls erstere im Teilbaum liegen, sondern daß, wenn ein Knoten im Teilbaum liegt, auch seine Geschwisterknoten von u im Teilbaum liegen müssen. Deshalb ist auch Baum v_2 von Abb. A.5 im Gegensatz zu v_1 kein Teilbaum von u, wenn die gestrichelten

Kanten mit den Knoten wegfallen, die mit B, G und H markiert sind.

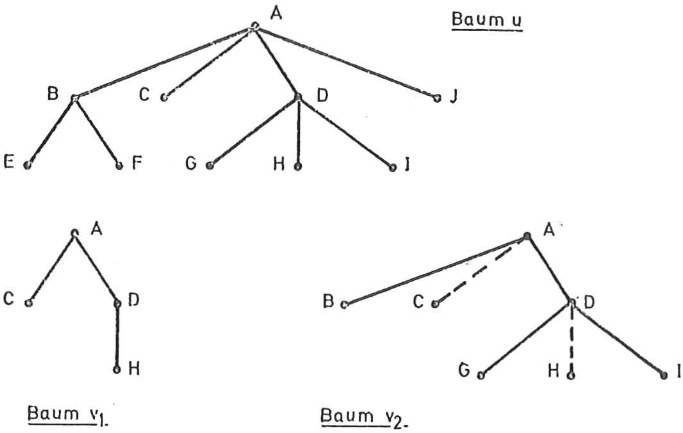


Abbildung A.4: Baumausschnitte von u

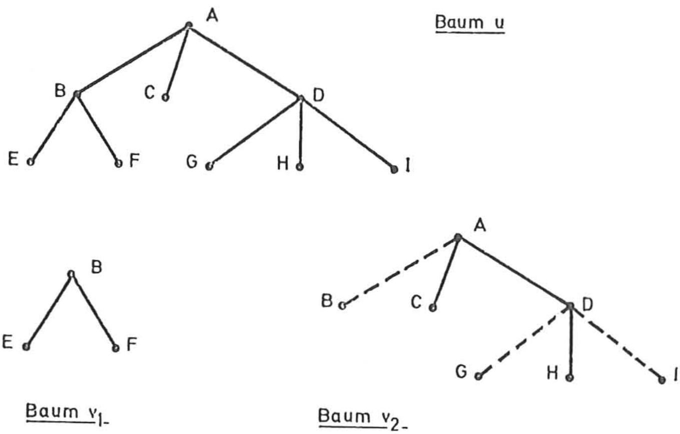


Abbildung A.5: Teilbäume von u

Def.: (koradikal, terminal) (A14)

Seien u, v Bäume.

v ist koradikaler Baumausschnitt bzw. koradikaler Teilbaum von u, gdw.:

(1) v ist Baumausschnitt bzw. Teilbaum von u

(2) $x_o(u) = x_o(v)$

d.h. u und v haben gleiche Wurzelknoten.

v ist terminaler Baumausschnitt bzw. terminaler Teilbaum von u, gdw.:

(1) v ist Baumausschnitt bzw. Teilbaum von u

(2) für alle Knoten $k \in Kn(v)$ gilt:
wenn k terminaler Knoten von v ist, dann ist k auch terminaler Knoten von u.

Erläuterung

Zur Erläuterung vgl. Abb. A.4: Baum v_1 ist sowohl koradikaler als auch terminaler Baumausschnitt von u.

In Abb. A.5 ist v_1 terminaler, v_2 koradikaler Teilbaum von u.

Veränderung von Bäumen

Im Hinblick auf die Regeln, die wir für eine Übersetzungsgrammatik definieren, beschränken wir uns auf zwei Möglichkeiten, Bäume zu verändern: die Ersetzung von terminalen Teilbäumen durch andere Bäume und die Erweiterung eines Baumes durch Hinzufügen eines anderen Baumes rechts oder links von einer vorhandenen Kante.

Def.: (Ersetzung) (A15)

Seien u, t_1, v nicht-leere Bäume; sei t_2 ein Baum.

v ist die Ersetzung von t_1 durch t_2 in u, gdw.:

Fall 1: t_2 ist ein nicht-leerer Baum

(1) t_2 ist koterminaler Teilbaum von v

(2) t_1 ist koterminaler Teilbaum von u

(3) $Kn(v) = (Kn(u) \setminus Kn(t_1)) \cup Kn(t_2)$

(4) $Ma(v) = (Ma(u) \setminus Ma(t_1)) \cup Ma(t_2)$

(5) $x_o(v) = x_o(u)$

(6) $$L(v) = ((L(u) \setminus L(t_1)) \setminus \{(k_1,k_2) \mid (k_1,k_2) \in L(u) \wedge (k_1 = x_o(t_1) \vee k_2 = x_o(t_1))\}) \cup L(t_2) \cup \{(k_1,k_2) \mid (k_1 = x_o(t_2) \wedge (x_o(t_1),k_2) \in L(u)) \vee (k_2 = x_o(t_2) \wedge (k_1,x_o(t_1)) \in L(u))\}$$

(7) $M(v) = (M(u) \setminus M(t_1)) \cup M(t_2)$

Fall 2: t_2 ist der leere Baum

(1) es gilt (2) - (5), (7) von Fall 1

(2) $$L(v) = (L(u) \setminus L(t_1)) \setminus \{(k_1,k_2) \mid (k_1,k_2) \in L(u) \wedge (k_1 = x_o(t_1) \vee k_2 = x_o(t_1))\}$$

Wir sagen dann auch, v entsteht aus u durch Ersetzung von t_1 durch t_2.

Ist t_2 der leere Baum, dann sagen wir auch, v entsteht aus u durch Tilgung von t_1.

Erläuterung

Zur Erläuterung vgl. Abb. A.6!

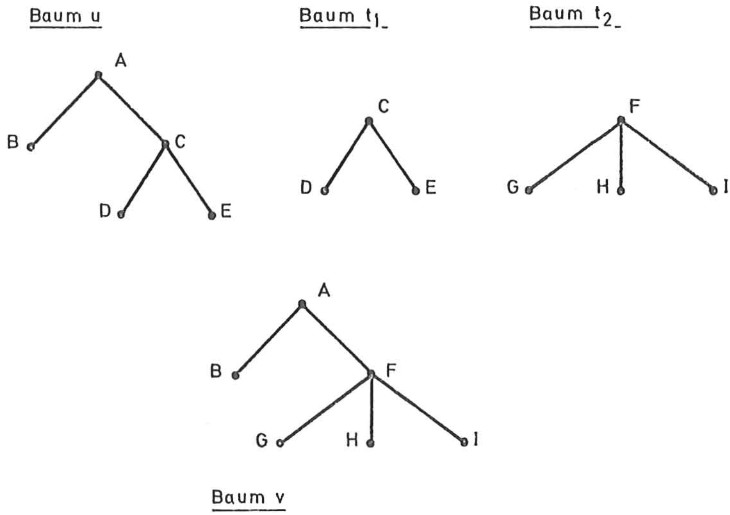


Abbildung A.6: Ersetzung von t_1 durch t_2

Def.: (Rechtserweiterung, Linkserweiterung) (A16)

Seien u, v, t nicht-leere Bäume; sei k ein Knoten von u.

v ist die Linkserweiterung (bzw. Rechtserweiterung) von u um t neben dem Knoten k, gdw.:

(1) $Kn(v) = Kn(u) \cup Kn(t)$

(2) $Ma(v) = Ma(u) \cup Ma(t)$

(3) $M(v) = M(u) \cup M(t)$

(4) $D(v) = D(u) \cup D(t) \cup \{(x,x_o(t)) \mid (x,k) \in D(u)\}$

(5) <u>bei Linkserweiterung</u>:

$$L(v) = (L(u) \setminus \{(x,k) \mid (x,k) \in L(u)\}) \cup \{(x_o(t),k)\} \cup L(t) \cup \{(x,x_o(t)) \mid (x,k) \in L(u)\}$$

<u>bei Rechtserweiterung</u>:

$$L(v) = L(u) \setminus \{(k,x) \mid (k,x) \in L(u)\} \cup \{(k,x_o(t))\} \cup L(t) \cup \{(x_o(t),x) \mid (k,x) \in L(u)\}$$

Wir sagen dann auch, v <u>entsteht</u> <u>aus</u> u <u>durch</u> <u>Linkserweiterung</u> <u>von</u> u <u>um</u> t <u>neben</u> dem Knoten k.

Wird der Knoten k, neben dem erweitert wird, direkt dominiert vom Wurzelknoten von u, und gibt es keinen Knoten k' von u, der links (bzw. rechts) von k liegt, d.h. gilt:

(6) $(x_o(u),k) \in D(u)$

(7) $\neg(\exists k' \in Kn(u))$
$(k',k) \in L(u)$ bzw. $(k,k') \in L(u)$

dann sagen wir auch einfacher, v <u>entsteht</u> <u>aus</u> u <u>durch</u> <u>Linkserweiterung</u> (bzw. <u>Rechtserweiterung</u>) <u>von</u> u <u>um</u> t.

<u>Erläuterung</u>

Zur Erläuterung vgl. Abb. A.7. Für den dort abgebildeten Baum v gilt:

v entsteht aus u durch Linkserweiterung von u um t neben dem mit D markierten Knoten

oder

v entsteht aus u durch Rechtserweiterung von u um t neben dem mit C markierten Knoten.

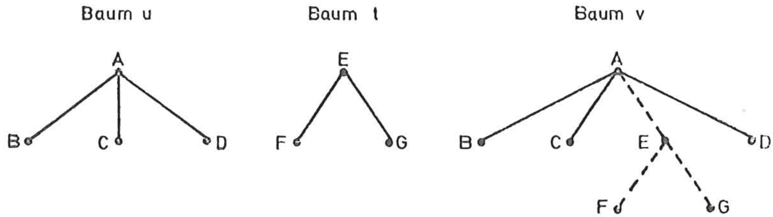


Abbildung A.7: Erweiterung von u um t

Bäume und Alphabete

Def.: (Wort eines Baumes, Endkette) (A17)

Sei u ein Baum; sei L_{total} die totale Linksorientierung von u; sei $\alpha = a_1 \ldots a_n$ ein Wort.

α ist das Wort von u, gdw.:

$(\exists k_1, \ldots, k_n \in Kn(u))\ (\forall i,j \leq n)$

(1) k_i ist terminaler Knoten von u

(2) $(a_i, k_i) \in M(u)$

(3) $(k_i, k_j) \in L_{total} \longrightarrow i < j$

Das Wort eines Baumes u nennen wir auch Endkette von u.

Erläuterung

Das Wort eines Baumes ist die Verkettung der Markierungen seiner terminalen Knoten, wobei die Verkettung in der Reihenfolge erfolgt, die der totalen Linksorientierung von u entspricht. Abb. A.8 zeigt ein Beispiel für das Wort eines Baumes.

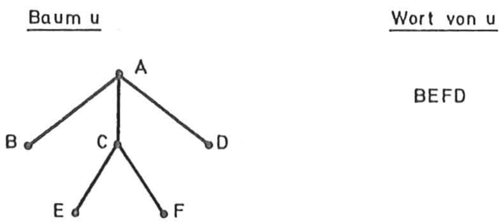


Abbildung A.8

Def.: (Baum über einem Alphabet) (A18)

Sei u ein Baum; sei A ein Alphabet.

u ist ein Baum über dem Alphabet A, geschrieben u(A), gdw.:

$Ma(u) \subseteq A$

d.h. alle Markierungen von u sind Symbole aus dem Alphabet A.

Eine Menge B nennen wir die Baummenge über dem Alphabet A, geschrieben $B^{(A)}$, gdw.:

$$B^{(A)} = \{u \mid u \text{ ist ein Baum über dem Alphabet } A\}$$

A.3.2 Linearisierung von Bäumen

Zur linearisierten Darstellung von Bäumen verwenden wir Klammerausdrücke, die wir als Listen bezeichnen. In der Folge schränken wir jedoch den allgemeinen Begriff der Liste, wie er etwa in der Programmiersprache LISP (vgl. McCARTHY et al. 1965) verwendet wird, ein auf solche Listen, die sich als Bäume darstellen lassen.

Def.: (Liste) (A19)

Listen (über dem Alphabet A) definieren wir rekursiv:

1. Symbole aus A sind Listen.
2. Die leere Liste $l = ()$ ist eine Liste.
3. Ist a ein Symbol aus dem Alphabet A und ist $l_1,\dots,l_n$ eine Folge von Listen mit $n \geq 0$, dann ist auch $l = (a,l_1,\dots,l_n)$ eine Liste.
4. Andere Ausdrücke aus A^* sind keine Listen.

Die Menge aller Listen über dem Alphabet A bezeichnen wir dann mit $L^{(A)}$.

Darstellung von Listen

In 3. von Definition A19 haben wir die Liste l als m-Tupel, $m = n+1$, notiert, wobei die erste Komponente des m-Tupels ein Symbol ist und die anderen Komponenten Listen sind, d.h. also wiederum Tupel oder auch Symbole. Dabei haben wir die runden Klammern als metasprachliche Symbole zur Kennzeichnung von Tupeln verwendet. Da l nicht definiert wurde als Verkettung von Symbolen sondern als ein geordnetes Gebilde, ist bei der Darstellung von Listen auf diese Ordnung zu achten. M.a.W. als Verkettung von Symbolen betrachtet ist eine Liste über dem Alphabet A kein Wort aus A^* sondern ein Wort über dem Alphabet $A' = A \cup \{(,)\}$.

D.h. am Beispiel erklärt: Die Liste $l = (a,l_1,l_2)$ mit $l_1 = (b,c)$ und $l_2 = (e,f)$ ist zu schreiben als $l = (a,(b,c),(e,f))$ und nicht als Verkettung von a, l_1, l_2, was dann als Wort $w = abcef$ zu schreiben wäre.

Wir vereinbaren jedoch als Schreibkonvention eine Einsparung der Kommata, wenn eine Liste explizit aufgezählt wird, und schreiben für $l = (a,(b,c),(e,f))$ auch $l = (a(b\ c)(e\ f))$.

In bestimmten Fällen ist es erforderlich, lediglich die Folge von Symbolen zu betrachten, die sich aus denjenigen Symbolen einer Liste bilden läßt, die ungleich den Klammersymbolen sind. Wir definieren deshalb den Begriff der Symbolfolge einer Liste:

<u>Def.:</u> (Symbolfolge einer Liste) (A20)

Seien $l, l_1, \ldots, l_n \in (A \cup \{(,)\})^*$ Listen über dem Alphabet A; sei $\lambda \in A^*$ ein Wort.

Die <u>Symbolfolge</u> <u>von</u> l definieren wir rekursiv:

1. Ist $l = a$, dann ist ein Wort λ die Symbolfolge von l, gdw.: $\lambda = a$.

2. Ist $l = (a, l_1, \ldots, l_n)$ und gibt es Wörter $\lambda_1, \ldots, \lambda_n \in A^*$, so daß λ_i die Symbolfolge von l_i, $i \leq n$, dann ist das Wort $\lambda = a\lambda_1 \ldots \lambda_n$ die Symbolfolge von l.

3. Andere Wörter aus A^* sind keine Symbolfolgen von l.

Def.: (Baum einer Liste) (A21)

Sei u ein Baum; sei l eine Liste über dem Alphabet A, $l_1, \ldots, l_n$ eine Folge von Listen über A; sei $a \in A$ ein Symbol.

u ist der Baum von l, gdw. einer der folgenden Fälle zutrifft:

Fall 1

u ist der leere Baum und l die leere Liste.

Fall 2

(1) $l = a$

(2) $Kn(u) = \{x_o(u)\}$

d.h. u hat genau einen Knoten

(3) $(a, x_o(u)) \in Ma(u)$

d.h. der Wurzelknoten von u ist mit a markiert.

Fall 3

es gibt eine Folge von terminalen Teilbäumen $v_1, \ldots, v_n$ von u, so daß

$(\forall i, i \leq n)(\forall j, 2 \leq j \leq n)$

(1) $l = (a, l_1, \ldots, l_n)$

(2) $(a, x_o(u)) \in Ma(u)$

d.h. der Wurzelknoten von u ist mit a markiert

(3) $(x_o(u), x_o(v_i)) \in D(u)$

d.h. der Wurzelknoten von u dominiert die Wurzelknoten der Teilbäume $v_1, \ldots, v_n$

(4) $(x_o(v_{j-1}), x_o(v_j)) \in L(u)$

d.h. die Wurzelknoten von je zwei in der Folge benachbarten Teilbäumen liegen entsprechend der Aufzählung der Folge in u nebeneinander

(5) v_i ist der Baum von l_i

Erläuterung

Anstelle einer verbalen Erläuterung geben wir in Abb. A.9 einige Beispiele für Listen und ihre Bäume.

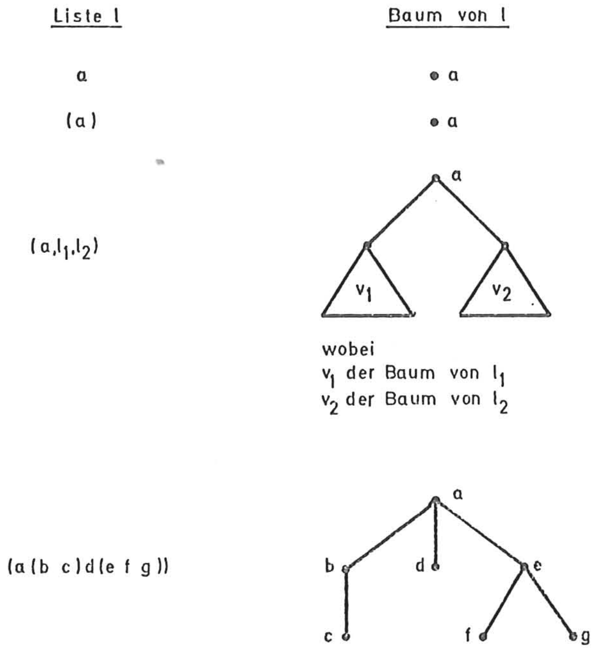


Abbildung A.9: Beispiele für Bäume von Listen

A.3.3 Darstellung von Ableitungen

Während üblicherweise so vorgegangen wird, daß zu einer Ableitung in einer kontextfreien Grammatik ein formales oder informales Verfahren angegeben wird zur Erzeugung entsprechender Ableitungsbäume und dann eine linearisierte Darstellung von Bäumen durch Klammerausdrücke festgelegt wird, gehen wir umgekehrt vor. Wir geben zuerst ein Verfahren an, wie zu einer kontextfreien Grammatik G_1 eine Grammatik G_2 konstruiert werden kann, so daß G_2 Listen erzeugt. Die Bäume dieser Listen sind dann Ableitungsbäume von G_1.

<u>Def.:</u> (Ableitungslistengrammatik) (A22)

Seien $G_1 = (VN_1, VT_1, S_1, P_1)$ und $G_2 = (VN_2, VT_2, S_2, P_2)$ kontextfreie Grammatiken; seien [und] Hilfssymbole zur Darstellung von Listen; sei _ ein Hilfssymbol zur Markierung von Symbolen.

G_2 ist eine <u>Ableitungslistengrammatik</u> <u>zu</u> G_1, gdw.:

(1) $VN_2 = \{\varphi(x) | x \in VN_1\}$

wobei φ eine Markierungsfunktion

$\varphi: A(G_1) \longrightarrow A(G_2) \times \{\_\}$

$\varphi(a) = (a,\_)$ wenn $a \in VN_1$

$\varphi(a) = a$ wenn $a \in VT_1$

Ein Paar $(a,\_)$ schreiben wir als markiertes Symbol: $\underline{a}$

φ wird erweitert auf $(VN_1 \cup VT_1)^*$ durch

$\varphi(a_1 \ldots a_n) = \varphi(a_1) \ldots \varphi(a_n)$

(2) $VT_2 = VN_1 \cup VT_1 \cup \{[\,,\,]\}$

(3) $S_2 = \varphi(S_1)$

(4) $P_2 = \{\psi(p) \mid p \in P_1\}$

wobei ψ eine Abbildung

$$\psi : VN_1 \times A(G_1)^+ \longrightarrow VN_2 \times (VN_1 \cup A(G_2))^+$$

$$\psi(a,\beta) = (\varphi(a), [a\ \varphi(\beta)])$$

Erläuterung

φ ist eine Markierungsfunktion, die nichtterminale Symbole von G_1 mit dem Hilfssymbol $\_$ markiert. Statt der üblichen Markierung durch Abbildung auf die Menge $\{0,1\}$ kennzeichnen wir aus Darstellungsgründen mit dem Unterstreichungsstrich $\_$.

Beispiel: $\varphi(NP) = \underline{NP}$
$\varphi(DET\ N) = \underline{DET}\ \underline{N}$
$\varphi(Hund) = Hund$

ψ verändert eine Produktionsregel $p \in P_1$, $p = (a \longrightarrow \beta)$ in der Weise, daß eine Regel p' entsteht, nach der das markierte a, also $\underline{a}$, ersetzt wird durch einen mit [ ] geklammerten Ausdruck, der aus dem nichtmarkierten a und dem markierten β besteht.

Beispiel: $\psi(NP \longrightarrow DET\ N) = \underline{NP} \longrightarrow [NP\ \underline{DET}\ \underline{N}]$ (R1)
$\psi(DET \longrightarrow der) = \underline{DET} \longrightarrow [DET\ der]$ (R2)
$\psi(N \longrightarrow Hund) = \underline{N} \longrightarrow [N\ Hund]$ (R3)

Beispiel für die Anwendung der Regeln R1, R2, R3:

R1 auf NP angewandt: $l_1 = [NP\ \underline{DET}\ \underline{N}]$
R2 auf l_1 angewandt: $l_2 = [NP\ [DET\ der]\ \underline{N}]$
R3 auf l_2 angewandt: $l_3 = [NP\ [DET\ der]\ [N\ Hund]]$

Durch die Funktion ψ werden also - in einer prozeduralen Redeweise gesprochen - die Regeln von G_1 so verändert, daß die erzeugten Wörter Listen sind, die sich dann als Bäume darstellen lassen. Abb. A.10 zeigt den Baum von Liste l_3 des o.a. Beispiels .

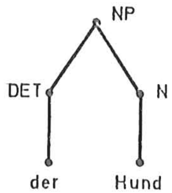


Abbildung A.10: Baum der Liste l_3

Def.: (Ableitungsliste, Ableitungsbaum, Struktur) (A23)

Sei G_1 eine kontextfreie Grammatik; sei G_2 eine Ableitungslistengrammatik zu G_1.

Eine Liste l ist eine (terminale) Ableitungsliste von G_1, gdw.:

l ist (terminal) abgeleitet in G_2

Den Baum von l nennen wir dann auch (terminalen) Baum oder Ableitungsbaum zu G_1.

Ist α das Wort eines Ableitungsbaumes u von G_1, dann heißt u bzw. die Liste von u auch Struktur von α (in G_1).

ANHANG B: INDEX DER WICHTIGSTEN BEGRIFFE UND DEFINITIONEN

ANHANG C: MATHEMATISCHE UND LOGISCHE SYMBOLE

C.1 Mengentheoretische Symbole

$\in$	Element von	$\cup$	Vereinigung
$\notin$	kein Element von	$\cap$	Durchschnitt
$\subseteq$	Teilmenge von	$\setminus$	Differenz
$\subset$	echte Teilmenge von	$\times$	(kartesisches) Produkt
$\neq$	ungleich	$\mathbb{P}$	Potenzmenge
		$\mathbb{N}$	Menge der natürlichen Zahlen

(Zur näheren Erläuterung der vorstehenden Symbole vgl. S. 203 - 206!)

C.2 Logische Symbole

$\wedge$	Konjunktion	$\longleftrightarrow$	Bikonditional
$\vee$	Disjunktion	gdw.	"
$\neg$	Negation	$\forall$	Allquantor
$\longrightarrow$	Konditional	$\exists$	Existenzquantor
		$\exists!$	'es gibt genau ein'

(Zur Verwendung der logischen Symbole vgl. S. 208!)

L i t e r a t u r v e r z e i c h n i s

Aho, A.V./Ullman, J.D. (1973), The Theory of Parsing, Translation and Compiling, Englewood Cliffs/N.J.: Prentice Hall.

Berry-Rogghe, G.L., Lutz, H.D., Saukko, K. (1978), Das Informationssystem PLIDIS, in: Kolvenbach/ Lötscher/Lutz (1978).

Berry-Rogghe, G.L./Wulz, H. (1977), An overview of PLIDIS. A Problem Solving Information System with German as Query Language, in: Proc. of the workshop on 'Natural Language for Interaction with Data Bases', Laxenburg/Austr., 10-14 Jan. 1977.

Braun, S. (1971), Eigenschaften strukturierter Symbole in formalen Sprachen, Habilitationsschrift, TU München.

Bresztowszky, U./Günther, A./Leiss, H./Lutz-Hensel, M. (1974), Linguistische Operationen in der generativen Transformationsgrammatik (= Forschungsberichte des Instituts für Kommunikationsforschung und Phonetik der Universität Bonn, 47), Hamburg: Buske.

Brockhaus, K. (1971), Automatische Übersetzung. Untersuchungen am Beispiel der Sprachen Englisch und Deutsch, Braunschweig: Vieweg.

Charniak, E./Wilks, Y. (eds.) (1976), Computational Semantics. An Introduction to Artificial Intelligence and Natural Language Comprehension, Amsterdam: North-Holland.

Chauché, J. (1974), Transducteurs et arborescences. Etudes et réalisations de systèmes appliquées aux grammaires transformationelles, Diss. Grenoble.

Chomsky, N. (1959), On Certain Formal Properties of Grammars, in: Information and Control 2, 137-167.

Chomsky, N. (1969), Aspekte der Syntax-Theorie, Frankfurt a.M.: Suhrkamp.

Creswell, M.G. (1973), Logics and Languages, London: Methuen.

Dilger, W./Zifonun, G.(1978), Eine Regelgrammatik für KS, Mannheim: IdS, Abt. LDV, unveröff. Arbeitspapier.

Droop, H.G. (1977), Das präpositionale Attribut. Darstellung und Korpusanalyse (= Forschungsberichte des Instituts für deutsche Sprache Mannheim, 34), Tübingen: Narr.

Ehrig, H./Kreowski, H.-J. (1976), Parallel Graph Grammars, in: Lindenmayer/Rozenberg (1976), 425-468.

Ginsburg, S./Partee, B. (1969), A Mathematical Model of Transformational Grammars, in: Information and Control 15, 297-334.

Gries, D. (1971), Compiler Construction for Digital Computers, New York: Wiley.

v. Hahn, W. (1978), Probleme der Simulationstheorie und Fragepragmatik bei der Simulation natürlichsprachlicher Dialoge, - Vortrag bei der Arbeitstagung "Kognitive Psychologie", Hamburg.

Hayes, P. (1975), Impressions of the 4th IJCAI, Tbilisi, Georgia, in: AISB-Newsletters 21, 19-20.

Kade, O. (1968), Kommunikationswissenschaftliche Probleme der Translation (= Grundfragen der Übersetzungswissenschaft, Beihefte zur Zeitschrift Fremdsprachen, II), Leipzig: Verlag Enzyklopädie.

Knuth, D.E. (1968), Semantics of Context - Free Languages, in: Mathematical Systems Theory 2, 127-145.

Knuth, D.E. (1968a), The Art of Computer Programming. Fundamental Algorithms, (Vol. 1), Reading/Mass.: Addison-Wesley.

Kolvenbach, M./Lötscher, A./Lutz, H.D. (Hrsg.), Künstliche Intelligenz und natürliche Sprache. Sprachverstehen und Problemlösen mit dem Computer (= Forschungsberichte des Instituts für deutsche Sprache, 42), Tübingen: Narr.

Kratzer, A./Pause, E./v. Stechow, A. (1973/1974), Einführung in Theorie und Anwendung der generativen Syntax (= Schwerpunkte Linguistik und Kommunikationswissenschaft, 7/I und 7/II) Frankfurt/M.: Athenäum.

Lewis, D.K. (1970), General Semantics, in: Synthese 22, 18-67.

Lindenmayer, A./Rozenberg, G. (eds) (1976), Automata, Languages and Development. At the Crossroads of Biology, Mathematics and Computer Science, Amsterdam: North-Holland.

Lötscher, A. (1977), Syntaktische Analyse in PLIDIS. Benutzerbeschreibung von PASS1. Korrigierte und überarbeitete Version von H.D. Lutz, Mannheim: IdS, Abt. LDV, unveröff. Arbeitspapier.

Lötscher, A. (1978), Automatische syntaktische Analyse des Deutschen mit Übergangsnetzwerken, in: Kolvenbach/Lötscher/Lutz (1978).

Lötscher, A./Kolvenbach, M. (1978), Morphosyntaktische Analyse in einem Frage-Antwort-System, in: Kolvenbach/Lötscher/Lutz (1978).

McCarthy, J. et al. (1965), LISP 1.5 Programmer's Manual, Cambridge/Mass.: M.I.T.

Menzel, W. (1974), Berechenbarkeit, Vorlesungsskriptum der Fakultät für Informatik, Universität Karlsruhe.

Montague, R. (1970), Universal Grammar, in: Theoria 36, 373-396.

Neubert, A. (1968), Semantik und Übersetzungswissenschaft, in: Ruzicka, R. (Hrsg.), Probleme der strukturellen Grammatik und Semantik, Leipzig: Karl-Marx-Universität.

Pause, E. (1974), Einzelsprache und Interlingua. Einige Aspekte zum Aufbau einer Übersetzungsgrammatik, in: Linguistische Berichte 32, 1-14.

Pause, E. (1976), Zur Theorie transformationeller Syntaxen. Generative Kraft - Entscheidbarkeit - Analyse (= Linguistische Forschungen, 14), Wiesbaden: Athenaion.

Pratt, T.W. (1971), Pair Grammars, Graph Languages and String-to-Graph Translations, in: Journal of Computer and System Sciences 5, 560-595.

Riesbeck, Ch.K. (1974), Computational Understanding. Analysis of Sentences and Context, Castagnola: Istituto per gli Studi Semantici e Cognitivi.

Rosen, K.R. (1973), Tree-Manipulating Systems and Church-Rosser Theorems, in: Journ. ACM 20, 160-187.

Rustin, R. (ed.) (1973), Natural Language Processing, New York. Algorithmic Press.

Schank, R.C./Colby, K.M. (eds.) (1973), Computer Models of Thought and Language, San Francisco: Freeman.

Schreiber, P.P. (1976), Baum-Transduktoren, Diss. TU Berlin.

Schütte, K. (1977), Proof Theory, Berlin: Springer.

Schwind, C. (1977), Ein Formalismus zur Beschreibung der Syntax und Bedeutung von Frage-Anwort-Systemen, Diss. TU München, FB Informatik, TUM-INFO 7710.

Suppes, P. (1973), Semantics of Context-free Fragments of Natural Languages, in: Hintikka, K.J.J. et al. (eds.), Approaches to Natural Language, Dordrecht/ Holl.: Reidel.

Thomason, R. (1972), A Semantic Theory of Sortal Incorrectness, in: Journal of Philosophical Logic 1, 209-258.

Wall, R. (1972), Introduction to Mathematical Linguistics, Englewood Cliffs/New Jersey: Prentice Hall. Deutsche Ausgabe: Einführung in die Logik und Mathematik für Linguisten (= Scriptor Taschenbücher, S12/S13), Kronberg/Ts: Scriptor.

Wilks, Y. (1975), An Intelligent Analyzer and Understander of English, in: Comm. ACM 18, 264-274.

Wilss, W. (1977), Übersetzungswissenschaft. Probleme und Methoden, Stuttgart: Klett.

Winograd, T. (1971), Procedures as a Representation for Data in a Computer Programm for Understanding Natural Language, MIT: Ph.D. Thesis, 1971.

Woods, W.A. (1970), Transition Network Grammars for Natural Language Analysis, in: Comm. ACM 13, 591-606.

Woods, W.A./Kaplan, R.M./Nash-Webber, B. (1972), The Lunar Science Natural Language Information System. Final Report, Cambridge/Mass.: BBN-Report No. 2378.

Wulz, H. (1976), Was hat künstliche Intelligenz mit natürlicher Sprache zu tun?, in: ZGL 4, 356-366.

Wulz, H./Zifonun, G. (1978), Automatische Problemlösung und Sprachverstehen als Forschungsgegenstände, in: Kolvenbach/Lötscher/Lutz (1978).

Zifonun, G. (1976), Die Konstruktsprache KS, Mannheim: IdS, Abt. LDV, Arbeitspapier LDV III/8/76.

Zifonun, G. (1978), Interne Repräsentation, in: Kolvenbach/Lötscher/Lutz (1978).