Aus: P. Eisenberg (Hrsg.): *Semantik und künstliche Intelligenz. Beiträge zur automatischen Sprachbearbeitung II.* Berlin, New York 1977. S. 59 - 85.

PETER HELLWIG

Ein Computermodell für das Folgern in natürlicher Sprache[1]

1. Problemstellung

Für den Linguisten sind unter den verschiedenen Unternehmungen der automatischen Sprachbearbeitung jene besonders interessant, in denen versucht wird, einen Ausschnitt des menschlichen Gebrauchs der Sprache mit Hilfe des Computers zu simulieren. Das Gesamt aus Eingabe, Programmabläufen und Ausgabe hat hier den Status eines Modells. Programme und Daten zusammen sind nichts anderes als eine Theorie über den entsprechenden sprachlichen Objektbereich. Gegenüber anderen linguistischen Theorien hat die Computersimulation eine Reihe von Vorteilen. Wünschenswerte Eigenschaften von Modellen, wie Widerspruchsfreiheit, Explizitheit, Vollständigkeit und Einfachheit, werden einer rigorosen Prüfung unterzogen. Bei geeigneter Konstruktion des Programmsystems lassen sich Einzelheiten der Theorie leicht manipulieren und so dem Modellobjekt immer mehr annähern. Wenn sprachliches Handeln den Gegenstand der Theorie bildet, dürfte schließlich ein dynamisches Modell, wie es ein Programmsystem darstellt, am genauesten den Anspruch auf Adäquatheit erfüllen können.

Der Gebrauch der Sprache, um den es uns im weiteren gehen soll, ist das Folgern. Folgerungen zu ziehen ist eine wichtige Teilaktivität sprachlicher Kommunikation. Die theoretische Klärung der Folgerungsmechanismen in natürlicher Sprache ist zudem die Voraussetzung für die technische Verwirklichung von Dialogsystemen zwischen Mensch und Maschine. Zwar gibt es schon relativ hochentwickelte Programme zur automatischen Frage-Beantwortung und mechanischen Problemlösung, die über Deduktionskapazitäten verfügen. Die mir bekannten operieren jedoch entweder über Formeln einer mehr oder weniger komplizierten Kunstsprache, oder aber sie sind auf sehr bedingte Verwendungen einer natürlichen Sprache beschränkt und damit weniger leistungsfähig. Bei Verwendung eines

1 Ich danke Klaus Brockhaus, Wolfgang Klein und Robert Maier für die Durchsicht einer früheren Fassung dieser Arbeit und für einige kritische Hinweise.

Deduktionssystems der ersteren Sorte muß der menschliche Bearbeiter selbst seine Fragen in die Sprache des Systems übersetzen. Dabei muß er jedoch bereits eine Einsicht in den Problemlösungsgang besitzen und eine ganze Reihe von Deduktionen vorab selbst vollziehen. Wo es sich nicht um ganz spezielle Anwendungsbereiche handelt, ist der Nutzen dieser Art von Dialogsystemen daher noch gering.[2]

Es gibt noch einen weiteren wichtigen Grund, der automatische Deduktionssysteme auf natürlichsprachiger Grundlage erstrebenswert macht. In den natürlichen Sprachen ist es möglich, über alle nur denkbaren Objektbereiche Aussagen zu machen. Ein Deduktionssystem auf der Basis einer natürlichen Sprache wäre daher maximal flexibel. Eine Kunstsprache von gleicher Stärke, die einfach genug ist, so daß sie ohne großen Aufwand von menschlichen Benutzern erlernt und gehandhabt werden kann, ist nicht in Sicht. Es wäre auch unnötig, eine solche zu entwickeln, wenn man den Prozeß des Folgerns in natürlicher Sprache durch Automaten direkt nachvollziehen könnte. Umfängliche Übersetzungsoperationen, sei es durch einen menschlichen Bearbeiter, sei es durch ein eigenes Programmsystem, wären dann ebenfalls überflüssig. Im folgenden soll gezeigt werden, daß ein effektives Deduktionsverfahren, das über natürlichsprachigen Daten operiert, durchaus im Bereich des Möglichen liegt.

1.1 Deduktives System

Unter einer 'Aussage' verstehen wir einen Satz, der wahr oder falsch sein kann. Eine Aussage, die nicht selbst wieder Aussagen als Bestandteile enthält, heiße 'atomare Aussage', eine Aussage, die aus anderen Aussagen besteht, heiße 'molekulare Aussage'. Wenn zwischen einer Menge von Aussagen und einer weiteren Aussage die Folgebeziehung besteht, so ist, wenn erstere Aussagen wahr sind, auch die letztere Aussage wahr.

Ein 'deduktives System S' ist eine Menge von Aussagen, die in zwei Klassen zerfällt. Die erste Klasse bilden Aussagen, deren Wahrheit vorausgesetzt wird. Wir nennen diese Aussagen 'Axiome von S'. Die zweite Klasse bilden Aussagen, deren Wahrheit aus der Wahrheit der Axiome gefolgert werden kann. Wir nennen diese Aussagen 'Theoreme von S'. Zu einem Computerprogramm, das Folgerungen simuliert, bilden die Axiome eines deduktiven Systems

---

2 Zu Deduktionssystemen auf der Basis von Logikkalkülen siehe Chang/Lee (1973) und die darin enthaltene Bibliographie, zu solchen mit natürlichsprachiger Eingabe siehe den Forschungsbericht Simmons (1970).

die 'Datenbasis', oder kurz: die 'Basis'. Die Theoreme des Systems dagegen sind gerade die Aussagen, die wir als Ausgabe des Programms erwarten.

Der Inhalt der Basis kann vom Benutzer des Programms frei bestimmt werden. Es sind jedoch die folgenden Regeln zu beachten:

1) Alle Sätze der Eingabe müssen den Formationsregeln der Eingabesprache gemäß gebildet sein, so daß kein Satz schon aus formalen Gründen uninterpretierbar ist.
2) Die Sätze der Eingabe müssen allein aufgrund ihrer Form eindeutig sein. Für natürlichsprachige Eingaben bedeutet das in der Praxis, daß sie mit einer desambiguierenden Strukturbeschreibung versehen sein müssen.[3]
3) Nur wahre Aussagen dürfen in die Datenbasis aufgenommen werden. Für die Wahrheit der Eingaben trägt der Benutzer die Verantwortung. Die Wahrheit von Prämissen eines Schlusses reduziert sich für das Deduktionsprogramm auf die Frage, ob die entsprechenden Aussagen in der Basis gegeben sind, und die Falschheit einer Prämisse reduziert sich auf die Frage, ob ihr Negat in der Basis gegeben ist. Es werden also keine Wahrheitswerte zu Aussagen gespeichert, sondern bei Bedarf wird das Gegebensein von Aussagen in der Basis überprüft.
4) Die Basis muß widerspruchsfrei gehalten werden. Ein Widerspruch liegt vor, wenn eine Aussage zugleich mit ihrem Negat in der Basis gegeben ist, oder wenn sich aus den Axiomen der Basis eine Aussage zusammen mit ihrem Negat folgern läßt.

1.2 Ableitungsregeln

Unter den Prämissen eines Schlusses kann man im allgemeinen zwei Sorten von Aussagen unterscheiden. Die eine Sorte bilden Aussagen, die eine Wahrheitswertfunktion beinhalten. Sie mögen kurz 'funktionale Aussagen' heißen. Die andere Sorte von Aussagen enthält keine solche Funktion. Wir nennen diese Aussagen 'elementare Aussagen'.

---

3 Die Automatisierung der Zuordnung einer Strukturbeschreibung in Gestalt von Analyseprogrammen lasse ich hier außer Betracht. Näheres dazu siehe Hellwig (1974). Ein vollständiges natürlichsprachiges Deduktionssystem muß natürlich über ein geeignetes Desambiguierungsverfahren verfügen.

(1) *Die Lampe brennt nur dann, wenn der Strom eingeschaltet ist.*
(2) *Der Strom ist nicht eingeschaltet.*

(1) ist eine funktionale Aussage, (2) ist eine elementare Aussage. Durch die Funktion, die (1) beinhaltet, wird den verschiedenen Möglichkeiten der logischen Konjunktion aus zwei Aussagen bzw. ihren Negaten je ein Wahrheitswert zugeordnet. Die Konjunktionsmöglichkeiten, die den Wert *wahr* erhalten, sollen nach H. Reichenbach die '*truth cases*' der funktionalen Aussage genannt werden.[4] Die *truth cases* von (1) sind:

(3) *((die Lampe brennt) und (der Strom ist eingeschaltet)).*
((die Lampe brennt nicht) und (der Strom ist eingeschaltet)).
((die Lampe brennt nicht) und (der Strom ist nicht eingeschaltet)).

Eine Aussage, die eine Wahrheitswertfunktion beinhaltet, zu behaupten, heißt behaupten, daß ein Objektbereich gegeben ist, in dem die *truth cases* der funktionalen Aussage möglich sind und alle anderen Fälle unmöglich. Durch die Aufnahme einer funktionalen Aussage in die Datenbasis wird also das Modell eines Objektbereiches mit einer Reihe möglicher Zustände errichtet. Natürlich ist der Benutzer bei der Errichtung eines bestimmten Modells in der Datenbasis völlig frei.[5]

Aus den *truth cases* einer funktionalen Aussage ist ablesbar, ob eine Folgebeziehung zwischen den beteiligten Aussagen besteht, und wenn ja, zwischen welchen. Immer wenn zusammen mit einer Aussage *(a)* die Aussage *(b)*, nicht aber das Negat von *(b)* einen der *truth cases* bildet, stehen *(a)* und *(b)* in Folgebeziehung. Da der Fall *((a) und (nicht b))* in dem durch die funktionale Aussage errichteten Modell unmöglich ist, ist, wenn *(a)* gegeben ist, das Gegebensein von *(b)* logisch notwendig.

Stehen zwei Aussagen *(a)* und *(b)* in Folgebeziehung, so können wir eine Regel für die entsprechende Folgerungsoperation formulieren. Wir nennen diese Regel 'Ableitungsregel' und schreiben:

(4) <*(a)*. $\vdash$ *(b)*.>

4 Siehe Reichenbach (1966) 27.

5 Um Mißverständnissen vorzubeugen, sei ausdrücklich darauf hingewiesen, daß ich den Modellbegriff hier und sonst im Sinne der Kybernetik und nicht im Sinne der mathematischen Modelltheorie verwende. Zur vorliegenden Interpretation des Terminus Modell siehe Klaus (1969) 411-426.

Dies ist zu lesen: "Wenn *(a).* gegeben ist, so darf *(b).* erzeugt werden". Die Aussage *(a).* in einer Ableitungsregel wie (4) soll 'Antezedenz', die Aussage *(b).* 'Postzedenz' heißen. Zu unserem Beispiel lassen sich die folgenden beiden Ableitungsregeln aufstellen:

(5) <*(Die Lampe brennt).*⊢ *(Der Strom ist eingeschaltet).*>
<*(Der Strom ist nicht eingeschaltet).*⊢ *(Die Lampe brennt nicht).*>

Die Aussagen (1) und (2) sind Elemente der Objektsprache, d. h. sie finden ihre Interpretation in der Bezugnahme auf außersprachliche Objekte und Sachverhalte. Auf welche Objekte und Sachverhalte sie tatsächlich bezogen werden, steht im Belieben des Benutzers, der die Datenbasis erstellt. Die Ableitungsregeln (5) dagegen sind Elemente der Metasprache unseres Deduktionssystems. Sie beziehen sich auf mögliche, objektsprachliche Aussagen in der Basis.[6] Es ist erlaubt, Ableitungsregeln in die Datenbasis aufzunehmen. Die Axiomenmenge unseres deduktiven Systems kann also sowohl objektsprachliche wie metasprachliche Aussagen enthalten.

Der Zusammenhang zwischen (1) und (5) läßt es wünschenswert erscheinen, auch über Regeln für das Ableiten von Ableitungsregeln zu verfügen. Demjenigen, der eine Datenbasis erstellt, steht es frei, solche zu formulieren. Er legt sich dadurch jedoch auf einen bestimmten Gebrauch seiner Objektsprache fest. Nimmt er z. B. eine Regel in die Basis auf, der zufolge aus einer Aussage der Form *((α) nur dann, wenn (β)).* die Ableitungsregeln <*(α).*⊢ *(β).*> und <*(nicht α).*⊢ *(nicht β).*> ableitbar sind, so hat er damit den Gebrauch des Ausdrucks *nur dann, wenn* im Kontext von *(α)* und *(β)* bestimmt und ist im weiteren an diesen Gebrauch gebunden. Der Zusammenhang zwischen Antezedenz und Postzedenz einer Regel zur Ableitung von Ableitungsregeln ist also ein analytischer. Da analytische Folgerungen sowohl theoretisch wie praktisch eine besondere Rolle spielen, sehen wir für diese Art von Ableitungsregeln ein besonderes Symbol vor, nämlich '⊩'. Zu lesen ist dies jedoch ganz so wie '⊢', nur daß die Argumente beider Operationen von unterschiedlicher Art sind.

6 Die metasprachlichen Ableitungsregeln dürfen nicht verwechselt werden mit objektsprachlichen Behauptungen der materiellen Implikation. Erstere sind Anweisungen für eine Folgerungsoperation, letztere bezeichnen eine bestimmte Konstellation von *truth cases* in einem Objektbereich.

Aus programmtechnischen Gründen werden wir unter den analytischen Ableitungsregeln noch eine bestimmte Menge besonders auszeichnen. Das geschieht durch das Symbol '$\underset{o}{\Vdash}$'. Im Gegensatz zu den analytischen Ableitungsregeln sollen die zuerst besprochenen Regeln mit dem Symbol '$\vdash$' 'empirische Ableitungsregeln' heißen, denn sie haben Gültigkeit nur in bezug auf das in der Datenbasis errichtete Modell eines Objektbereiches. Für die Theorie der Folgerungen ist es wichtig festzuhalten, daß Deduktionen nur möglich sind, wenn sowohl analytische wie empirische Ableitungsregeln vorhanden sind.

1.3 Ableitungsprozedur

Die Handlung des Folgerns kann nun auf einfache Weise als das Befolgen von Ableitungsregeln definiert werden. Es muß nur festgelegt werden, wie das Befolgen von Ableitungsregeln vor sich zu gehen hat. Bevor wir dazu nähere Angaben machen, müssen wir noch die Unterscheidung von Aussagen und Aussageformen einführen. 'Aussageformen' sind Formeln, die Variable enthalten und zu Aussagen werden, wenn für alle Variablen passende Konstanten substituiert werden. Eine Konstante ist passend, wenn sie vom selben Typ ist wie die Variable, und wenn sie identisch ist mit allen Konstanten, die innerhalb derselben Formel bereits für die Variable substituiert wurden. Der Leser wird in diesen Festlegungen die Substitutionsregel erkennen, die in der symbolischen Logik allgemein gebräuchlich ist. Eine Aussage, die entsprechend dieser Regel aus einer Aussageform konstruiert worden ist, soll eine 'Instanz' der Aussageform heißen. Darüber hinaus wollen wir auch zwei identische Aussagen als Instanzen voneinander betrachten. Es ist praktisch, Aussageformen in Ableitungsregeln zuzulassen. Durch sie können Instanzen zusammengefaßt werden, für die analoge Ableitungsmöglichkeiten gelten sollen.

Folgerungen lassen sich nun als reine Symbolmanipulationen formulieren und sind leicht zu programmieren. Die Ableitungsprozedur besteht aus folgenden Schritten:

1: Programmstart.
Eine gegebene oder zuvor abgeleitete Aussage wird eingelesen.
Die erste gegebene Ableitungsregel wird eingelesen.

2: Es wird überprüft, ob die eingelesene Aussage eine Instanz des Antezedens der eingelesenen Ableitungsregel ist. Ist das der Fall, so wird zu 3: übergegangen, andernfalls zu 5:.

3: Es wird eine Instanz zum Postzedenz der eingelesenen Ableitungsregel gebildet, wobei alle Variablen, die mit Variablen des Antezedenz identisch sind, auf dieselbe Weise ersetzt werden, wie in der Instanz des Antezedenz.

4: Die erzeugte Instanz des Postzedenz der Ableitungsregel wird als Theorem ausgegeben.

5: Ist in der Basis keine weitere Ableitungsregel mehr vorhanden, so Programmstop. Andernfalls wird die nächste gegebene Ableitungsregel gelesen und zu 2: übergegangen.

Im Rahmen des deduktiven Gesamtprogramms sehen wir vor, daß die obige Ableitungsprozedur für jede Aussage aufgerufen wird, die neu in die Datenbasis aufgenommen werden soll, jedoch mit der Einschränkung, daß dabei nur analytische Ableitungsregeln mit dem Symbol '$\Vdash_{o}$' benutzt werden dürfen.

Wir haben mit den obigen Ausführungen die Grundzüge eines deduktiven Systems umrissen, das auf einer Regellogik im Unterschied zu einer Satzlogik beruht.[7] Eine Satzlogik hat zum Ziel, tautologisch wahre Aussagen zu ermitteln. Die Gesetze der Satzlogik sagen jedoch nichts darüber aus, was zu tun ist, wenn man von einer bestimmten Aussage auf eine andere schließen will. Allerdings sind im Rahmen der Satzlogik Verfahren entwickelt worden, nach denen sich berechnen läßt, ob eine gegebene Menge von Aussagen Tautologien oder Widersprüche enthält.[8] Dies machen sich viele formale Fragebeantwortungs- und Problemlösungssysteme zunutze. Läßt sich z. B. aus der Hinzufügung einer Aussage zu einer Menge von Aussagen ein Widerspruch errechnen, so führt die Hinzufügung des Negats der Aussage zu einer Tautologie. Dadurch ist bewiesen, daß das Negat der Aussage ein Theorem des entsprechenden deduktiven Systems darstellt.

Ein Deduktionssystem auf der Basis einer Regellogik funktioniert viel direkter. Die Datenbasis enthält hier nicht nur objektsprachliche Aussagen, sondern Regeln für das Folgern selber. Theoreme können durch Anwendung dieser Regeln aus Axiomen direkt abgeleitet werden und nicht über den Umweg der Berechnung von Widersprüchen oder Tautologien innerhalb der Vereinigungsmenge der betreffenden Aussagen. Für natürliche Sprachen dürfte es auch sehr schwierig sein, ein geeignetes Berechnungsverfahren zu entwickeln, und nicht weniger schwierig ist es, natürlichsprachige Aussagen in einen satzlogi-

---

7 Zur Unterscheidung von Satzlogik und Regellogik und zur Übersetzbarkeit der ersteren in letztere siehe Bochenski/Menne (1965) § 0.83 und § 9.

8 Ein häufig eingesetztes Berechnungsverfahren ist das 'resolution principle' von J. A. Robinson. Vgl. Robinson (1965).

gischen Kalkül, der entsprechende Berechnungen erlaubt, zu übersetzen. Dagegen macht es verhältnismäßig wenig Schwierigkeiten, für natürlichsprachige Eingaben analytische Ableitungsregeln zu formulieren, aus denen sich dann bei Gegebensein eines bestimmten Modells in der Basis empirische Folgerungsmöglichkeiten ergeben.

2. Aussagen

Als Beispiel einer natürlichen Sprache wählen wir für die folgenden Kapitel das Deutsche. Wir werden freilich nur einen Ausschnitt berücksichtigen, der gerade groß genug ist, um die Prinzipien unseres Deduktionsmodells deutlich zu machen. Dabei beschränken wir uns auch bewußt auf einen Bereich, der einen Vergleich mit den bisher üblichen satzlogischen Deduktionssystemen erlaubt, obwohl sich die eigentlichen Vorzüge der regellogischen Deduktion erst bei der Behandlung von schwierigeren Fällen des natürlichsprachigen Folgerns zeigen. Durch Aufnahme weiterer Regeln ließe sich der Ausschnitt des Deutschen, der in den Folgerungsprozeduren Verwendung finden kann, jedenfalls rasch erweitern.

2.1 Analyse in L-Konstituenten

Um Ableitungsregeln formulieren zu können, muß man sich auf eine Formbeschreibung der natürlichsprachigen Äußerungen beziehen können, da aus der reinen Folge der Ausdrücke nicht ohne weiteres die Struktur der Äußerung ersichtlich ist. Der Zweck, nämlich in Ableitungen verwendet zu werden, wirkt auf die Art der Strukturbeschreibung zurück. Nicht jede Zerlegung in Konstituenten, durch die Wohlgeformtheit und Eindeutigkeit von Aussagen definiert werden kann, ist geeignet. Vielmehr benötigt man eine Analyse in genau die syntaktischen Einheiten, die in Folgerungsoperationen eine Rolle spielen. Solche Einheiten sollen 'logische Konstituenten' oder kurz 'L-Konstituenten' heißen.[9] Es ist hier nicht der Platz, Einzelheiten der Heuristik darzulegen. Nur soviel sei gesagt: L-Konstituenten erhält man, wenn man Kommutationen, statt sie unter dem Gesichtspunkt der Wohlgeformtheit und nur im Rahmen von Einzelsätzen durchzuführen, unter Wahrung der Folgebeziehung und im Rahmen von Prämissen und Konklusion von Schlußfiguren vornimmt. Für Klassen von L-Konstituenten kann man Variable einführen. Um die Entsprechungen von Variablen und Konstanten zu regeln, muß man zu beiden Kategorienanga-

9 Der Terminus stammt von P. Hinst. Siehe Hinst (1974) 49 ff.

ben hinzufügen. Das Ergebnis der Analyse läßt sich durch etikettierte Klammerungen festhalten, die sich in die Form von Baumgraphen bzw., zum Zwecke der maschinellen Verarbeitung, in Listenstrukturen überführen lassen.

Zu den Beispielen der folgenden Kapitel geben wir nur sehr rudimentäre Strukturbeschreibungen, da es uns hier nicht so sehr auf sie ankommt. Kategorien haben wir fortgelassen und Klammern spärlicher benutzt, als es eigentlich notwendig wäre. Wo die Eindeutigkeit in Gefahr gerät, werden wir Erläuterungen geben. Es mögen folgende Vereinbarungen gelten: Alle unmittelbar in runden Klammern stehenden, also von keinem weiteren Klammerpaar umgebenen Ausdrücke bilden eine L-Konstituente. Den Ausdruck *nicht* betrachten wir als eine Art Vorzeichen zum unmittelbar folgenden Ausdruck. Er bildet mit letzterem zusammen eine L-Konstituente. Enthält ein Klammerpaar wiederum Klammerpaare, so ist die L-Konstituente, die ersterem entspricht, den L-Konstituenten übergeordnet, die letzteren entsprechen. Die Strukturbeschreibung, die daraus resultiert, ist mit der einer Dependenzgrammatik vergleichbar, außer daß Variablen nicht nur für ein einzelnes abhängiges Element stehen können, sondern u. U. für ganze Bäume. Damit entsprechen die Variablen in unseren Aussageformen eher den Konstituenten in einer Phrasenstrukturgrammatik.

2.2 Ableitung von Ableitungsregeln

In den folgenden Abschnitten sollen uns zunächst molekulare Aussagen des Deutschen beschäftigen. Wir beginnen damit, eine Datenbasis einzurichten, indem wir die folgenden analytischen Ableitungsregeln als Axiome einführen. Die griechischen Buchstaben in den Regeln sind Aussagevariablen. Das Symbol '.' kennzeichnet eine Aussage als abgeschlossen.

(R 1) <(*wenn* (α) *dann* (β)). $\Vdash_{o}$ <(α). $\vdash$ (β).>>
(R 2) <((α) *oder*$_{D}$ (β)). $\Vdash_{o}$ <(*nicht* α). $\vdash$ (β).>>
(R 3) <((α) *oder*$_{E}$ (β)). $\Vdash_{o}$ <(α). $\vdash$ (*nicht* β).>>
(R 4) <(*nur wenn* (α) *dann* (β)). $\Vdash_{o}$ <(*nicht* α). $\vdash$ (*nicht* β).>>
(R 5) <((α) *dann und nur dann wenn* (β)). $\Vdash_{o}$ <(α). $\vdash$ (β).> <(*nicht* α). $\vdash$ (*nicht* β).>>[10]
(R 6) <(*entweder* (α) *oder* (β)). $\Vdash_{o}$ <(α). $\vdash$ (*nicht* β).> <(*nicht* α). $\vdash$ (β).>>
(R 7) <<(α). $\vdash$ (β).> $\Vdash_{o}$ <(*nicht* β). $\vdash$ (*nicht* α).>>

---

10 Die logische Konjunktion von empirischen Ableitungsregeln im Postzedenz analytischer Regeln bleibt unbezeichnet.

Wenn wir, wie Wittgenstein in den "Philosophischen Untersuchungen", unter der Bedeutung von sprachlichen Zeichen ihren Gebrauch verstehen, dann wird durch die obigen Regeln zumindest ein Teil der Bedeutung einiger Zeichen unserer Beispielsprache, nämlich ihr Gebrauch in Schlußfolgerungen, beschrieben. Die Aufstellung von analytischen Ableitungsregeln wie den obigen fällt somit in die Domäne der Semantik der ausgewählten Objektsprache. Sie ist zugleich die Voraussetzung dafür, daß das Datenverarbeitungssystem über eine deduktive Kapazität für empirisch wahre, also ein bestimmtes Modell betreffende, Ableitungen verfügt. Was im einzelnen empirisch gefolgert werden kann, hängt natürlich davon ab, welche Aussagekonstanten bei der Ableitung an die Stellen der Variablen in unseren Regeln treten.

Empirische Ableitungsregeln, wie sie dem Postzedenz von (R 1) bis (R 6) entsprechen, können nur abgeleitet werden, wenn eine abgeschlossene Aussage die Instanz des Antezedenz bildet. Eine 'abgeschlossene Aussage' ist eine solche, die selbständig behauptet worden ist. Per definitionem können abgeschlossene Aussagen nicht Teil anderer Aussagen sein. Die obigen Regeln sind also nicht zu verwechseln mit Ersetzungsregeln, nach denen Aussagen oder Teile von Aussagen umgeformt werden können, und die wir in 2.3 besprechen werden.

Die Regeln (R 1) bis (R 6) haben wir als Beispiele ausgewählt, weil sie genau die Menge aller möglichen Ableitungen von einer Aussage (α). auf eine Aussage (β). einschließlich der Negate umfassen. Die Konstanten in den Aussageformen haben, wie man sieht, dieselbe Bedeutung wie die aussagelogischen Junktoren Implikation, Disjunktion, Exklusion, Replikation, Äquivalenz und Kontravalenz.

Der Ausdruck *oder* ist im Deutschen mehrdeutig. Er kann Disjunktion, Exklusion oder Kontravalenz meinen. Es ist jedoch für unser Deduktionssystem praktisch, neben *entweder oder* für die Kontravalenz auch je einen eindeutigen Ausdruck für die Disjunktion und die Exklusion zu besitzen. Wir haben daher für die interne Repräsentation den Ausdruck *oder* durch Subskripte desambiguiert. Im gewöhnlichen Deutsch geschieht die Desambiguierung durch Umschreibungen. Mit Hilfe von Umformungsregeln - siehe (R 8) und (R 9) in 2.3 - sind aus diesen Umschreibungen die von uns eingeführten Ausdrücke $oder_D$ und $oder_E$ ableitbar.

Man vermißt vielleicht Regeln für die Aussageformen *((α) und (β))*. und *(weder (α) noch (β))*.. Die durch diese ausgedrückten Wahrheitswertfunktionen zeichnen sich, neben einigen anderen, dadurch aus, daß sie nur einen einzigen *truth case* spezifizieren. Tritt die Konjunktion *und* als Prädikat einer abgeschlossenen Aussage auf, so ist damit genau das Gegebensein der Aussagen in den Argumentpositionen behauptet und nichts sonst. Von einer gegebenen Aussage auf eine ebenfalls gegebene Aussage zu folgern, erübrigt sich jedoch. Funktionale Aussagen mit nur einem *truth case* scheiden daher für die Ableitung von Ableitungsregeln aus.

Durch (R 1) bis (R 6) haben wir dafür gesorgt, daß Ableitungen von *(α)*. bzw. *(nicht α)*. nach *(β)*. bzw. *(nicht β)*. möglich sind. Jede der zugrunde liegenden Menge von *truth cases* erlaubt aber auch eine bestimmte Folgerung von *(β)*. bzw. *(nicht β)*. nach *(α)*. bzw. *(nicht α)*.. Diese zweite Folgebeziehung steht mit der ersten in einem regelmäßigen Zusammenhang. Immer wenn *(ß)*. aus *(α)*. abgeleitet werden kann, kann *(nicht α)*. aus *(nicht β)*. abgeleitet werden. In der Satzlogik entspricht dem das Gesetz der Kontraposition der Implikation. Die zugehörige Regel ist in (R 7) formuliert. Mit Hilfe von (R 7) wird aus den im Postzedenz von (R 1) bis (R 6) stehenden Regeln je eine weitere Ableitungsregel erzeugt.

Die obigen analytischen Regeln werden im Rahmen der Ableitungsprozedur jedesmal angewendet, wenn eine Instanz des Antezedenz in die Datenbasis insertiert wird. Gegeben seien die folgenden Axiome:

(6) *(wenn (a) dann (b))*.
(entweder (nicht b) oder (c)).
(a).

Um nicht umfangreiche Sätze des Deutschen aufführen zu müssen, auf deren innere Struktur es im Augenblick nicht ankommt, sei vereinbart, daß die Buchstaben *a*, *b*, *c* ... Aussagekonstanten des Deutschen darstellen. Der Leser kann zur Probe dafür tatsächliche deutsche Aussagen setzen. Aus den ersten beiden abgeschlossenen Aussagen in (6) resultieren nach Anwendung von (R 1), (R 6) und (R 8) die folgenden empirischen Ableitungsregeln:[11]

(7) <*(a)*. ⊢ *(b)*.> <*(nicht b)*. ⊢ *(nicht a)*.>
<*(nicht b)*. ⊢ *(nicht c)*.> <*(c)*. ⊢ *(b)*.>
<*(b)*. ⊢ *(c)*.> <*(nicht c)*. ⊢ *(nicht b)*.>

Ruft man nun die Ableitungsprozedur für die in (6) gegebene elementare Aussage *(a)*. auf, so wird zunächst nach der ersten Regel in (7) das Theorem *(b)*. erzeugt. Macht man dieses zum Gegenstand einer weiteren Ableitung, so erhält man nach der fünften Regeln von (7) das Theorem *(c)*..

---

11 Dabei ist auch die weiter unten angeführte Ersetzungsregel (R19) schon benutzt worden.

2.3 Ersetzungsregeln

Ist eine funktionale Aussage gegeben, deren Argumentstellen wiederum von funktionalen Aussagen eingenommen werden, so reichen die Regeln (R 1) bis (R 7) in 2.2 nicht aus, um alle empirischen Ableitungsregeln direkt zu erzeugen, die aus einer solchen Aussage tatsächlich folgen. Nur für das Prädikat der abgeschlossenen funktionalen Aussage ist ja bisher eine analytische Ableitungsregel vorhanden.

(8) *(wenn (a) dann ((nicht b) oder*$_D$ *((c) und (d)))).*

Aus (8) können wir durch Anwendung von (R 1) die folgende Regel ableiten:

(9) <*(a).* ⊢ *((nicht b) oder*$_D$ *((c) und (d))).*>

Zwar wäre, falls *(a).* zu den Axiomen zählt, nach (9) die Aussage

(10) *((nicht b) oder*$_D$ *((c) und (d))).*

als Theorem ableitbar, und wir könnten daraus in einem weiteren Schritt mit Hilfe von (R 2) deduzieren:

(11) <*(b).* ⊢ *((c) und (d)).*>

Ein solches Verfahren würde jedoch bedeuten, daß wir bei jeder Hinzufügung eines elementaren Axioms zur Basis überprüfen müßten, ob sich nun Theoreme ableiten lassen, aus denen neue empirische Ableitungsregeln erzeugt werden können. Wünschenswert wäre dagegen, der Aussage (8) direkt alle empirischen Ableitungsregeln entnehmen zu können, ungeachtet der Frage, welche Aussagen sonst noch gegeben sind.

Es ist bekannt, daß tautologisch wahre Äquivalenzen der Satzlogik dazu benutzt werden können, gegebene Formeln in andere umzuformen. Es gilt die Regel, daß ein Ausdruck in einer Formel jederzeit durch einen anderen Ausdruck ersetzt werden darf, der mit ersterem tautologisch äquivalent ist. Steuert man die Umformungen in geeigneter Weise, so kann man komplexe Aussagen in eine bestimmte Normalform überführen. Die Form, die uns interessiert, ist die konjunktive Normalform. Eine 'konjunktive Normalform' ist eine Aussage, die nur aus Konjunktionen von Disjunktionen besteht. Für aussagenlogische Kalküle gilt, daß sich jede Aussage in eine ihr äquivalente Normalform umformen läßt.[12] Wir versuchen im folgenden, für die natürliche Sprache denselben Weg zu beschreiten.

---

12 Vgl. Bochensky/Menne (1965) § 10.32.

Ersetzungsregeln sind analytische Ableitungsregeln, in denen sowohl das Antezedenz wie das Postzedenz aus objektsprachlichen Ausdrücken besteht. Sie können nicht nur auf abgeschlossene Aussagen angewendet werden, sondern auf jede Ausdrucksfolge, die eine Instanz des jeweiligen Antezedenz ist. Um konjunktive Normalformen herstellen zu können, nehmen wir die folgenden Ersetzungsregeln in die Basis auf:

(R 8) <((entweder (α) oder (β)) oder (beides)) $\Vdash_o$
(((α) oder$_D$ (β))>
(R 9) <((entweder (α) oder (β)) oder (keins von beiden)) $\Vdash_o$
((α) oder$_E$ (β))>
(R10) <(wenn (α) dann (β)) $\Vdash_o$ ((nicht α) oder$_D$ (β))>
(R11) <((α) oder$_E$ (β)) $\Vdash_o$ ((nicht α) oder$_D$ (nicht β))>
(R12) <(nur wenn (α) dann (β)) $\Vdash_o$ ((α) oder$_D$ (nicht β))>
(R13) <((α) dann und nur dann wenn (β)) $\Vdash_o$
(((nicht α) oder$_D$ (β)) und ((α) oder$_D$ (nicht β)))>
(R14) <(entweder (α) oder (β)) $\Vdash_o$
(((nicht α) oder$_D$ (nicht β)) und ((α) oder$_D$ (β)))>
(R15) <((α) oder$_D$ (((β) und (γ)))) $\Vdash_o$
(((α) oder$_D$ (β)) und ((α) oder$_D$ (γ)))>
(R16) <(((α) und (β)) oder$_D$ ((α) und (γ))) $\Vdash_o$
((α) und ((β) oder$_D$ (γ)))>
(R17) <(((α) oder$_D$ (β)) oder$_D$ (γ)) $\Vdash_o$
((α) oder$_D$ ((β) oder$_D$ (γ)))>
(R18) <(((α) und (β)) und (γ)) $\Vdash_o$ ((α) und ((β) und (γ)))>
(R19) <(nicht(nicht α)) $\Vdash_o$ (α)>
(R20) <(nicht((α) oder$_D$ (β))) $\Vdash_o$ ((nicht α) und (nicht β))>
(R21) <(nicht((α) und (β))) $\Vdash_o$ ((nicht α) oder$_D$ (nicht β))>
(R22) <((α) oder$_D$ (β)) $\Vdash_o$ ((β) oder$_D$ (α))>
(R23) <((α) und (β)) $\Vdash_o$ ((β) und (α))>

Unser Beispiel (8) läßt sich nun folgendermaßen in eine konjunktive Normalform überführen:

(12) (wenn (a) dann ((nicht b) oder$_D$ ((c) und (d)))).
((nicht a) oder$_D$ ((nicht b) oder$_D$ ((c) und (d)))).
nach (R10),
((nicht a) oder$_D$ (((nicht b) oder$_D$ (c)) und
((nicht b) oder$_D$ (d)))).
nach (R15),
(((nicht a) oder$_D$((nicht b) oder$_D$ (c))) und
((nicht a) oder$_D$ ((nicht b) oder$_D$ (d)))).
nach (R15).

Die letzte Zeile von (12) ist die gesuchte Normalform.

Da wir alle abgeschlossenen Aussagen in der Datenbasis als Konjunkte voneinander auffassen, ohne diese Konjunktion zu repräsentieren, kann auch jede Konjunktion von Aussagen in selbständige Aussagen aufgelöst werden. Andererseits bilden aus demselben Grund zwei abgeschlossene Aussagen der Basis eine Instanz der Konjunktion aus beiden. Es gelten also die Regeln:

(R24) <((α) *und* (β)). |⊢ (α). (β).>
(R25) <(α). (β). |⊢ ((α) *und* (β)).>

Statt der letzten Zeile von (12) schreiben wir nun nach (24):

(13) *((nicht a) oder*$_D$ *((nicht b) oder*$_D$ *(c))).*
((nicht a) oder$_D$ *((nicht b) oder*$_D$ *(d))).*

Aus diesen beiden Axiomen lassen sich nun mit Hilfe von (R 2) die folgenden empirischen Ableitungsregeln erzeugen, wobei allerdings mit (R22) die Reihenfolge der Disjunkte verschiedentlich verändert, mit (R20) die Negation zu den atomaren Aussagen transportiert und mit (R19) die doppelte Verneinung beseitigt werden muß:

(14) <((b) *und* (nicht c)). ⊢ (nicht a).>
<((a) *und* (nicht c)). ⊢ (nicht b).>
<((a) *und* (b)). ⊢ (c).>
<((b) *und* (nicht d)). ⊢ (nicht a).>
<((a) *und* (nicht d)). ⊢ (nicht b).>
<((a) *und* (b)). ⊢ (d).>

Angenommen, die Aussagen *(a).* und *(b).* zählen zu den Axiomen. Nach (R25) ist dann auch *((a) und (b)).* gegeben. Letztere Aussage ist eine Instanz des Antezedenz der dritten und der sechsten Regel von (14). Aus *(a).* und *(b).* sind demnach die Theoreme *(c).* und *(d).* ableitbar. Der Fall, auf dem (11) basiert, ist also durch (14) neben mehreren anderen Möglichkeiten berücksichtigt, obwohl wir bei der Aufstellung von (14) das Gegebensein von *(a).* nicht zur Voraussetzung gemacht haben.[13]

3. Zielorientierte Ableitung

Sowohl in Dialogen zwischen menschlichen Kommunikationspartnern als auch in Systemen der automatischen Frage-Beantwortung und Problemlösung geht es normalerweise nicht darum, aus irgendwelchen oder gar allen für wahr gehaltenen Aussagen alle möglichen Folgerungen zu ziehen. Vielmehr steht meist die Wahrheit einer bestimmten Aussage in Frage, und Folgerungen werden vollzogen, um die betreffende Aussage mit Hilfe anderer zu verifizieren oder zu widerlegen. Es geht also darum festzustellen, ob die betreffende Aussage aus vorgegebenen

---

13 Werden immer zuerst Ersetzungsregeln angewendet, können übrigens die Regeln (R 1) und (R 3) bis (R 6) eingespart werden.

Axiomen ableitbar ist oder nicht. Das Gesamt der Operationen, die dabei vorgenommen werden, heißt 'Beweis'. Beweisabläufe kann man anschaulich anhand eines Frage-Antwort-Modells demonstrieren. Für den Rest unserer Ausführungen werden wir uns daher im Rahmen eines solchen Modells bewegen.

Eine 'Frage' sei eine Aussage, deren Status als Theorem eines deduktiven Systems noch nachzuweisen ist. Fragen sind ebenfalls abgeschlossene Aussagen, jedoch kennzeichnen wir ihren Status, indem wir sie durch das Symbol '?' an Stelle des für gegebene Aussagen verwendeten Symbols '.' abschließen.

Das Hauptproblem eines Beweises liegt darin, aus der eventuell großen Menge von Axiomen in der Datenbasis diejenigen herauszufinden, die als Prämissen für die Ableitung des fraglichen Theorems in Betracht kommen. Die Methode, probeweise aus beliebigen Axiomen Theoreme abzuleiten, in der Hoffnung, das fragliche zu finden, führt in zu viele Sackgassen. Ausgeschlossen ist es auch, alle deduzierbaren Theoreme selbst in die Datenbasis aufzunehmen, da dadurch zu bald die Grenzen der Speicherkapazität der Basis erreicht würden. Die Lösung des Problems liegt darin, daß alle Deduktionen direkt von den fraglichen Theoremen und nicht von den Axiomen aus gesteuert werden. Ein solches Ableitungsverfahren heißt 'zielorientiert'.[14] In den folgenden Abschnitten werden wir ein Modell zielorientierter Ableitung entwickeln.

3.1 Disjunktive Expansion einer Frage

Der direkteste Beweis einer Frage *(a)?* besteht darin, daß man die Aussage *(a)*. als zu den Axiomen zählend ermittelt. Stößt man in der Basis auf eine Ableitungsregel <*(b)*. ⊢ *(a)*.>, so ließe sich *(a)*. auch aus *(b)*. folgern. Das Beweisziel wäre also erreicht, wenn man ein Axiom *(a)*. oder ein Axiom *(b)*. fände und, falls die Datenbasis teilweise redundant sein sollte, natürlich auch, wenn beides der Fall wäre. Der weiteren Durchmusterung der Basis kann man daher als neue Fragestellung die Disjunktion *((a) oder*$_D$ *(b))?* zugrunde legen. Fände sich nun noch eine Regel <*(c)*. ⊢ *(b)*.>, so könnte man die Menge der gesuchten Axiome um *(c)*. erweitern, denn

14 Ein zielorientiertes Deduktionssystem für Problemlösungen wird in Hewitt (1972) entwickelt. T. Winograd benutzt das Deduktionssystem von C. Hewitt in seinem Modell eines natürlichsprachigen Dialogs mit einem 'Roboter', vgl. Winograd (1972). Den Arbeiten von Hewitt und Winograd verdanke ich wertvolle Anregungen.

aus *(c)*. folgt das unter anderem gesuchte *(b)*.. Die neue Fragestellung lautete also *(((a) oder*$_D$ *(b)) oder*$_D$ *(c))?*.

Aus diesen Überlegungen ergibt sich die Grundidee der disjunktiven Expansion: Immer wenn eine Ableitungsregel gefunden wird, deren Postzedenz mit einem fraglichen Theorem übereinstimmt, kann das Antezedenz der Regel als Disjunkt mit der ursprünglichen Frage verknüpft werden.[15] Auf diese Weise ist garantiert, daß nur Axiome berücksichtigt werden, die potentiell zum Beweisziel führen. Läßt sich im weiteren irgend ein Disjunkt der expandierten Frage als in der Basis gegeben verifizieren, so ist damit der Beweis für die ursprüngliche Frage erbracht.[16]

Es mögen unter anderen die folgenden Ableitungsregeln in der Basis vorhanden sein:

(15) $\langle(a).\ \vdash (b).\rangle$ $\langle(b).\ \vdash (c).\rangle$ $\langle(h).\ \vdash (g).\rangle$
$\langle(c).\ \vdash (d).\rangle$ $\langle(g).\ \vdash (d).\rangle$ $\langle(d).\ \vdash (e).\rangle$

Es sei die Frage *(e)?* zu beantworten.

Unter einer 'Stufe der Expansion' verstehen wir die Expansion aller Disjunkte der vorhergehenden Stufe, für die sich Ableitungsregeln mit passendem Postzedenz finden. Die ursprüngliche Frage bildet die nullte Stufe. Wir schließen die Disjunkte einer Stufe in eckige Klammern ein, verzichten aber in den Beispielen der Übersichtlichkeit wegen auf die runden Klammern der einzelnen Disjungierungen.

Die Expansion unserer Frage vollzieht sich damit auf den folgenden Stufen:

(16) 0. Stufe: [*(e)*]?
1. Stufe: [[*(e)*] *oder*$_D$ *(d)*]?
2. Stufe: [[[*(e)*] *oder*$_D$ *(d)*] *oder*$_D$ *(c) oder*$_D$ *(g)*]?
3. Stufe: [[[[*(e)*] *oder*$_D$ *(d)*] *oder*$_D$ *(c) oder*$_D$ *(g)*] *oder*$_D$ *(b) oder*$_D$ *(h)*]?
4. Stufe: [[[[[*(e)*] *oder*$_D$ *(d)*] *oder*$_D$ *(c) oder*$_D$ *(g)*] *oder*$_D$ *(b) oder*$_D$ *(h)*] *oder*$_D$ *(a)*]?

Eine Expansion ist 'abgeschlossen', wenn kein Disjunkt einer Stufe mehr eine Erweiterung auf einer nächsten Stufe findet. Für das einzige Disjunkt auf der 4. Stufe in (16), *(a)*, gibt es keine Regel in (15), deren Postzedenz mit *(a)* übereinstimmt. Die Expansion von *(e)?* ist daher auf der 4. Stufe abgeschlossen.

---

15 Regeln mit dem Symbol '$|\!\vdash_o$' sollen hiervon ausgeschlossen sein. Außerdem sei angenommen, daß das Symbol '?' eine Instanz von '.' bildet.

16 Für die Zulässigkeit des Verfahrens der disjunktiven Expansion und ihrer Evaluierung kann ein formaler Beweis erbracht werden, doch soll dieser hier aus Platzgründen unterbleiben.

Um zirkuläre Expansionen zu vermeiden, müssen wir noch die folgende Ersetzungsregel einführen:

(R26) <((α) *oder*$_D$ (α)) $\Vdash_o$ (α)>

(R26) soll während einer Expansion angewendet werden, wann immer dies möglich ist. Dadurch wird verhindert, daß ein schon vorhandenes Disjunkt ein zweites Mal mit einer Frage verknüpft wird. Gegeben sei z. B. folgendes Axiom nebst den entsprechenden Ableitungsregeln:

(17) *((a) dann und nur dann wenn (b)).*
<(a). $\vdash$ (b).> *<(nicht b). $\vdash$ (nicht a).>*
<(b). $\vdash$ (a).> *<(nicht a). $\vdash$ (nicht b).>*

Es sei die Frage *(a)?* zu expandieren. Die Expansion ist wegen (R26) schon auf der 1. Stufe abgeschlossen:

(18) [[*(a)*] *oder*$_D$ *(b)*]?

Ohne (R26) würden *(b)* um *(a)* und *(a)* um *(b)* fortlaufend expandiert, ohne daß es je zu einem Abbruch käme.

Molekulare Aussagen als Fragen werden, wenn möglich, zunächst in konjunktive Normalform umgeformt. Wir sehen analog zu (R24) vor:

(R24') <((α) *und* (β))? $\Vdash_o$ (α)? (β)?>

Lautet eine Frage

(19) *(wenn (a) dann ((nicht b) oder*$_D$ *((c) und (d))))?* ,

so ergeben sich daraus, ganz so wie (13) aus (8), die beiden selbständigen Fragen:

(20) *((nicht a) oder*$_D$ *((nicht b) oder*$_D$ *(c)))?*
((nicht a) oder$_D$ *((nicht b) oder*$_D$ *(d)))?*

Die Expansion der Disjunkte von (20) erfolgt auf die gewöhnliche Weise, d. h. für jedes einzelne Element werden Ableitungsregeln mit passendem Postzedenz gesucht, und ggf. wird das entsprechende Antezedenz auf der nächst höheren Stufe mit der bisherigen Disjunktion verknüpft. Konjunktive Normalform und disjunktive Expansion ergänzen sich also auf vorteilhafte Weise. (19) ist natürlich nur dann bewiesen, wenn sich sowohl die Expansion der ersten wie die der zweiten Frage in (20) verifizieren läßt.

3.2 Evaluierung der Expansion

Eine Frage kann positiv beantwortet werden, wenn ihre Expansion verifiziert werden kann. '■' sei das Symbol für den Wert *wahr* und '□' das Symbol für den Wert *falsch*. Läßt sich aus einer fraglichen Disjunktion das Symbol '■' ableiten, so gilt das der Expansion zugrunde liegende fragliche Theorem als bewiesen. Zur Evaluierung einer fraglichen Disjunktion stehen folgende Regeln zur Verfügung:

(R27) <*(α)* $\Vdash_{\circ}$ ■, wenn *(α).* in der Basis>
(R28) <*((α) oder*$_D$ *(nicht α))* $\Vdash_{\circ}$ ■>
(R29) <*((α) und (nicht α))* $\Vdash_{\circ}$ □>
(R30) <*((α) oder*$_D$ ■*)* $\Vdash_{\circ}$ ■>
(R31) <*((α) und* ■*)* $\Vdash_{\circ}$ *(α)*>
(R32) <*((α) oder*$_D$ □*)* $\Vdash_{\circ}$ *(α)*>
(R33) <*((α) und* □*)* $\Vdash_{\circ}$ □>

Die Regeln (R27) bis (R33) sind Ersetzungsregeln. Sie können also auf Teile der fraglichen Disjunktion angewendet werden. Die Regel (R27) gibt an, welchen Wert die Evaluierung eines fraglichen Disjunkts erhält, wenn ein entsprechendes Axiom in der Basis vorhanden ist. Die Form von (R27) fällt aus dem Rahmen des Definierten. Wir haben die Regel nur der Übersichtlichkeit wegen hier eingeführt. In der Praxis wird das, was (R27) besagt, am besten in Gestalt einer Match-Routine implementiert. Es ist eine Angelegenheit der Optimierung, ob schon während der Expansion auf jeder Stufe eine Evaluierung versucht wird, oder erst nachdem die Expansion der Frage abgeschlossen ist.

Wir nehmen noch einmal das Beispiel (6) auf. Gegeben sind die Axiome:

(6) *(wenn (a) dann (b)).*
(entweder (nicht b) oder (c)).
(a).

Aus (6) sind, wie wir nachgewiesen haben, die Ableitungsregeln (7) erzeugbar:

(7) <*(a).* ⊢ *(b).*> <*(nicht b).* ⊢ *(nicht a).*>
<*(nicht b).* ⊢ *(nicht c).*><*(c).* ⊢ *(b).*>
<*(b).* ⊢ *(c).*> <*(nicht c).* ⊢ *(nicht b).*>

Fraglich sei *(c)?*. Die Expansion der Frage lautet dann:

(21) [*(c)*]?
[[*(c)*] *oder*$_D$ *(b)*]?
[[[*(c)*] *oder*$_D$ *(b)*] *oder*$_D$ *(a)*]?

Mit der letzten Zeile von (21) ist die Expansion von *(c)?* abgeschlossen. Die Aussage *(a).* ist eines der Axiome in (6). Nach den Regeln (R27) und (R30) läßt sich daher aus der letzten Zeile von (21) das Symbol '■' ableiten. Damit ist *(c).* bewiesen, und die Frage *(c)?* kann durch Assertierung des Theorems *(c).* beantwortet werden.

Gegeben seien die folgenden Axiome:

(22) *(wenn (c) dann (nicht d)).*
(nur wenn (nicht a) dann (nicht d)).
(entweder (b) oder (nicht e)).
(nicht e).

Aus den ersten drei Aussagen in (22) können die folgenden empirischen Ableitungsregeln abgeleitet werden:

(23) <*(c).* ⊢ *(nicht d).*> <*(d).* ⊢ *(nicht c).*>
<*(a).* ⊢ *(d).*> <*(nicht d).* ⊢ *(nicht a).*>
<*(b).* ⊢ *(e).*> <*(nicht e).* ⊢ *(nicht b).*>
<*(nicht b).* ⊢ *(nicht e).*><*(e).* ⊢ *(b).*>

Eine Frage laute:

(24) *(wenn ((a) oder*$_{D}$ *(b)) dann (nicht c))?*

Die konjunktive Normalform von (24), aufgelöst in selbständige Fragen, ist:

(25) *((nicht a) oder*$_{D}$ *(nicht c))?*
((nicht b) oder$_{D}$ *(nicht c))?*

Die disjunktive Expansion der beiden Teilfragen ergibt:

(26) [[*((nicht a) oder*$_{D}$ *(nicht c)*] *oder*$_{D}$ *(nicht d) oder*$_{D}$ *(d)*]*?*
[[*((nicht b) oder*$_{D}$ *(nicht c)*] *oder*$_{D}$ *(nicht e) oder*$_{D}$ *(d)*]*?*

Die Expansionen in (26) sind nicht abgeschlossen, doch können die fraglichen Disjunktionen bereits evaluiert werden. Aus den Disjunkten *(nicht d) oder*$_{D}$ *(d)* der ersten Teilfrage ist nach (R28) '■' ableitbar und daher für die gesamte Expansion nach (R30) wiederum '■'. Die Expansion der zweiten Teilfrage läßt sich nach (R27) und (R30) zu '■' evaluieren, da eines ihrer Disjunkte, nämlich *(nicht e)*, zu den Axiomen in (22) zählt.

Es fällt auf, daß bei der Evaluierung von (26) nur die elementare Aussage *(nicht e).* in der Basis aufgesucht werden muß, nicht mehr jedoch die funktionalen Aussagen, aus denen Ableitungsregeln erzeugt worden sind. Da die Ableitungsregeln bereits für die Expansion herangezogen worden sind, werden die ursprünglichen Axiome bei der Evaluierung nicht mehr benötigt. Daß dies generell gilt, läßt sich wie folgt beweisen: Angenommen, jede junktorenlogisch komplexe Aussage sei in eine konjunktive Normalform überführbar. Aus jedem Konjunkt der Form *((α) oder*$_{D}$ *(β))* können dann zwei empirische Ableitungsregeln abgeleitet werden, nämlich <*(nicht α).* ⊢ *(β).*> und <*(nicht β).* ⊢ *(α).*>. Ist nun eine Disjunktion *((α) oder*$_{D}$ *(β))?* fraglich, so ergibt sich aus der Anwendung der beiden Regeln die Expansion [[*(α) oder*$_{D}$ *(β)*] *oder*$_{D}$ *(nicht β) oder*$_{D}$ *(nicht α)*]*?*. Letztere ist allein mit Hilfe von (R27) und (R30), ohne Rückgriff auf das ursprüngliche Axiom, zu '■' evaluierbar.

Wir ziehen aus dieser Tatsache die folgende Konsequenz: Funktionale Aussagen werden nicht in die Basis aufgenommen, sondern statt dessen die aus ihnen abgeleiteten Ableitungsregeln. Die Axiomenmenge unseres deduktiven Systems besteht also nur aus elementaren Aussagen und Algorithmen. Der Vorteil einer solchen Basis für die automatische Verarbeitung liegt auf der Hand.

Läßt sich ein fragliches Theorem nicht beweisen, so gibt die disjunktive Expansion desselben Auskunft darüber, unter welchen Bedingungen ein Beweis möglich wäre, nämlich dann, wenn eines der Disjunkte der Expansion in die Basis aufgenommen werden könnte. In einem als Dialog zwischen Mensch und Maschine konzipierten Frage-Antwort-System kann man den Computer rückfragen lassen, ob der Benutzer etwa das Gegebensein eines der fraglichen Disjunkte bestätigen kann. Ist dies der Fall, kann die ursprüngliche Frage noch nachträglich beantwortet werden. Auch für den Unterhalt einer Datenbank ist es vorteilhaft zu wissen, welche Information noch eingegeben werden muß, um von den Benutzern gestellte Fragen in Zukunft beantworten zu können. Schließlich ist die exakte Auskunft über die noch fehlenden Bedingungen für bestimmte Ableitungen ebenfalls wichtig, wenn das geschilderte Modell als heuristisches Mittel im Rahmen einer logisch orientierten linguistischen Semantik verwendet wird.

3.3 Beweisprozeduren

Die Beweisprozedur besteht aus den folgenden Schritten:

1: Programmstart.
Ein fragliches Theorem wird eingelesen und, falls notwendig, in konjunktive Normalform überführt.
Der Zähler für die Expansionsstufen, n, wird auf Ø gesetzt.

2: Die erste für die Expansion in Frage kommende Ableitungsregel wird aus der Datenbasis eingelesen.[17]

3: Es wird für jedes Disjunkt der n-ten Expansionsstufe der Frage überprüft, ob es eine Instanz des Postzedenz der eingelesenen Ableitungsregel ist.
Ist das der Fall, wird zu 4: übergegangen, sonst zu 5:.

4: Es wird eine Instanz zum Antezedenz der eingelesenen Ableitungsregel gebildet, wobei alle Variab-

---

17 In Frage kommen alle Regeln mit den Symbolen '$\vdash$' und '$|\vdash$', nicht jedoch solche mit dem Symbol '$|\vdash_{o}$'.

len, die mit Variablen im Postzedenz identisch sind, auf dieselbe Weise ersetzt werden wie in der Instanz des Postzedenz.[18]
Der gewonnene Ausdruck wird auf der n+1-ten Expansionsstufe mit den bisher gegebenen Disjunkten der Frage disjunktiv verknüpft.

5: Ist in der Basis keine weitere Ableitungsregel mehr vorhanden, wird zu 6: übergegangen.
Andernfalls wird die nächste in Frage kommende Ableitungsregel eingelesen und zu 3: übergegangen.

6: Es wird geprüft, ob auf der n+1-ten Stufe irgendein Disjunkt erzeugt worden ist.
Ist das der Fall, wird der Zähler n um 1 erhöht und zu 2: übergegangen.
Andernfalls wird zu 7: übergegangen.

7: Es wird versucht, mit Hilfe der Regeln (R27) bis (R33) das Symbol '■' abzuleiten.
Gelingt dies, so ist das fragliche Theorem bewiesen. Programmstop.
Gelingt dies nicht, so bleibt das Theorem fraglich. In Gestalt der erzeugten disjunktiven Expansion werden die Bedingungen ausgegeben, unter welchen ein Beweis möglich wäre. Programmstop.

Bei Verwendung der Beweisprozedur im Rahmen eines automatischen Frage-Antwort-Systems wird man zunächst die positive Beantwortung einer Frage anstreben, indem man versucht, sie als Theorem zu verifizieren. Ist dies nicht möglich, so wird man das Negat der Frage bilden und für dieses die Beweisprozedur aufrufen. Gelingt ein Beweis jetzt, so kann die Frage negativ beantwortet werden. Ist ein Beweis wiederum unmöglich, so lautet die Antwort *ungewiß*.

4. Terme

Die molekularen Aussagen des Deutschen, die wir bisher besprochen haben, waren von den Formeln einer aussagenlogischen Kunstsprache nicht sehr verschieden und ließen sich leicht in solche überführen. Sobald die innere Struktur atomarer Aussagen für das Folgern eine Rolle spielt, scheiden sich jedoch die Wege. Eine Übersetzung des Deutschen in einen der herkömmlichen prädikatenlogischen Kalküle ist so schwierig, daß sie nicht ratsam erscheint.

18 Wir haben bisher bei der Expansion die Anwendung von Regeln, die Variablen enthalten, nicht vorgesehen. Wir werden jedoch in 4.2 ein Beispiel für solche Regeln geben.

Unter der Voraussetzung, daß es Pudel gibt[19], bildet das folgende Beispiel eine gültige Schlußfigur:

(27) *Alle Pudel sind Hunde.*
Kein Pudel beißt.
———————————
Nicht alle Hunde beißen.

Die erste Prämisse von (27) ist die Grundlage dafür, daß, wenn auch die zweite Prämisse gegeben ist, die Konklusion von (27) gefolgert werden kann. Wir zählen deshalb die erste Prämisse von (27) zu den funktionalen Aussagen. Nun handelt es sich aber bei dieser im Deutschen nicht um eine molekulare Aussage, die so in Ableitungsregeln überführt werden könnte, daß direkt aus der Aussage *Kein Pudel beißt.* die Aussage *Nicht alle Hunde beißen.* ableitbar wäre. Die herkömmliche Prädikatenlogik löst das Problem durch Quantoren und gebundene Variable sowie eine Form, in der eine Aussage wie *Alle Pudel sind Hunde.* tatsächlich eine molekulare Aussage bildet. Im Deutschen kommen jedoch so gut wie keine gebundenen Variablen vor, und es widerspräche auch unserer Definition, den Satz *Alle Pudel sind Hunde.* als molekulare Aussage zu betrachten. Es ist daher unsere Aufgabe zu zeigen, daß die bisher geschilderten Prinzipien der Ableitung auch anwendbar sind, wenn man nur atomare Prämissen zur Verfügung hat, und daß man dabei gebundene Variablen vermeiden kann. Zur Demonstration wählen wir den Ausschnitt des Deutschen, den die Syllogistik behandelt. Da die Syllogistik im modernen Gewand als ein Teil der Klassenlogik aufgefaßt werden kann, sollen unsere Beispiele zugleich nachweisen, daß ein syntaktisch orientiertes Verfahren wie das unsere nicht von vorneherein schwächer ist als eine auf der Grundlage der Klassenlogik konzipierte Semantik.

4.1 Termableitungsregeln

Alle L-Konstituenten, aus denen Aussagen bestehen, sollen 'Terme' heißen. Wir versehen (27) mit folgender rudimentärer Strukturbeschreibung:

(28) *((Alle (Pudel)) sind (Hunde)).*
((Kein (Pudel)) beißt).
((Nicht alle (Hunde)) beißen).

Flexionsunterschiede wie *beißt* und *beißen* werden wir im Folgenden übergehen. In interner Repräsentation fallen sie weg. Die Strukturbeschreibung in (28) ist unzureichend, weil der Typ der Konstituenten nicht eindeutig ist.

(29) *((Alle (Pudel)) sind (zahm)).*

In (29) haben wir dieselbe Klammerung wie in der ersten Aussage von (28). Bei der genaueren Analyse müssen deshalb den Klammerpaaren Kategorienindices hinzugefügt werden. Wie bisher, wollen wir jedoch im Rahmen dieser Arbeit darauf verzichten. Die Klammerungen der folgenden Beispiele mögen analog zu (28) interpretiert werden.

Es sei nun das System der Ableitungsregeln um folgende Konvention erweitert: Läßt sich aus jeder Aussage $(\alpha).$ eine Aussage $(\beta).$ ableiten, wobei $(\alpha).$ und $(\beta).$ sich nur durch die Terme (δ) in $(\alpha).$ und (τ) in $(\beta).$ voneinander unterschieden, so soll eine Termableitungsregel $<(\delta) \vdash (\tau)>$ aufgestellt werden dürfen. Auf diesem Grundsatz, wenn auch meist nicht in dieser expliziten Weise, basieren bisher schon linguistische Gliederungen des Wortschatzes in Hyponyme, Synonyme, Antonyme usw..

Welche Termableitungen möglich sein sollen, muß in einem deduktiven System axiomatisch festgelegt werden. Dies geschieht durch bestimmte funktionale Aussagen, die in die Datenbasis aufgenommen werden, und durch analytische Ableitungsregeln für diese. Ist eine Objektsprache vorgegeben, so gilt es, die funktionalen Aussageformen zu ermitteln, aus denen Termableitungsregeln analytisch abgeleitet werden können.

Für den ausgewählten Bereich des Deutschen nehmen wir folgende analytische Ableitungsregeln in die Basis auf:

Ableitung von Ableitungsregeln:

(R34) $<((\mathit{alle}\ (\alpha))\ \mathit{sind}\ (\beta)).\ \underset{\circ}{\Vdash}\ <(\mathit{alle}\ (\beta))\ \vdash\ (\mathit{alle}\ (\alpha))>$
$<(\mathit{ein}_E\ (\alpha))\ \vdash\ (\mathit{ein}_E\ (\beta))>>$

(R35) $<((\mathit{ein}_E\ (\alpha))\ \mathit{ist}\ (\beta)).\ \underset{\circ}{\Vdash}\ <(\mathit{alle}\ (\alpha))\ \vdash\ (\mathit{ein}_E\ (\beta))>$
$<(\mathit{alle}\ (\beta))\ \vdash\ (\mathit{ein}_E\ (\alpha))>>$

(R36) $<<(\alpha)\ \vdash\ (\beta)>\ \underset{\circ}{\Vdash}\ <(\mathit{nicht}\ \beta)\ \vdash\ (\mathit{nicht}\ \alpha)>>$

Ersetzungsregeln:

(R37) $<(\mathit{wenigstens\ ein}\ (\alpha))\ \underset{\circ}{\Vdash}\ (\mathit{ein}_E\ (\alpha))>$
(R38) $<(\mathit{nicht\ ein}_E\ (\alpha))\ \underset{\circ}{\Vdash}\ (\mathit{kein}\ (\alpha))>$
(R39) $<((\mathit{alle}\ (\alpha))\ \mathit{nicht}\ \beta\ (\_\gamma))\ \underset{\circ}{\Vdash}\ ((\mathit{kein}\ (\alpha))\ \beta\ (\_\gamma))>$
(R40) $<((\mathit{ein}_E\ (\alpha))\ \mathit{nicht}\ \beta\ (\_\gamma))\ \underset{\circ}{\Vdash}\ ((\mathit{nicht\ alle}\ (\alpha))\ \beta\ (\_\gamma))>$
(R41) $<((\mathit{nicht\ alle}\ (\alpha))\ \mathit{nicht}\ \beta\ (\_\gamma))\ \underset{\circ}{\Vdash}\ ((\mathit{ein}_E\ (\alpha))\ \beta\ (\_\gamma))>$
(R42) $<((\mathit{kein}\ (\alpha))\ \mathit{nicht}\ \beta\ (\_\gamma))\ \underset{\circ}{\Vdash}\ ((\mathit{alle}\ (\alpha))\ \beta\ (\_\gamma))>$
(R43) $<(\mathit{alle}\ (\alpha))\ \underset{\circ}{\Vdash}\ (\mathit{ein}_E\ (\alpha))>$
(R44) $<(\mathit{kein}\ (\alpha))\ \underset{\circ}{\Vdash}\ (\mathit{nicht\ alle}\ (\alpha))>$
(R45) $<((\mathit{ein}_E\ (\alpha))\ \mathit{ist}\ (\beta)).\ \underset{\circ}{\Vdash}\ ((\mathit{ein}_E\ (\beta))\ \mathit{ist}\ (\alpha)).>$
(R46) $<((\mathit{kein}\ (\alpha))\ \mathit{ist}\ (\beta)).\ \underset{\circ}{\Vdash}\ ((\mathit{kein}\ (\beta))\ \mathit{ist}\ (\alpha)).>$

Alle Variablen in (R34) bis (R46) sind freie Variablen, d. h. sie werden in der Ableitungsprozedur nach dem allgemeinen Substitutionsprinzip durch Term-Konstanten ersetzt. In Gestalt von '$\_\gamma$' haben wir allerdings eine weitere Sorte von Variablen eingeführt. Hängen von einer übergeordneten L-Konstituente mehrere L-Konstituenten ab, so interessieren davon häufig nicht alle. Es ist daher praktisch, sie zusammenzufassen. Das Symbol '_' vor einer Variablen soll andeuten, daß eine beliebige Folge von L-Konstituenten der gleichen Stufe eine Instanz dieser Variablen bilden kann. Da die Typen nachgeordneter L-Konstituenten ohnehin von der dominierenden L-Konstituente determiniert werden, brauchen sie nicht angegeben werden.

Bei der Aufstellung von Regeln zur Ableitung von Termableitungsregeln ist darauf zu achten, daß die Kontraposition weiterhin gilt. Die Kontrapositionsregel für Termableitungen ist in (R36) formuliert. Mit ihrer Hilfe kann aus den empirischen Termableitungsregeln in (R34) und (R35) je eine weitere Regel erzeugt werden. Weiter ist es wichtig, die richtigen L-Konstituenten zu finden, für die die Termableitungsregeln gelten. So sind im vorliegenden Fall Ableitungsregeln für Terme ohne die quantifizierenden Ausdrücke *alle* und *ein*$_E$ nicht möglich, ein Faktum, daß m. E. die strukturelle Lexikologie bisher übersehen hat.[20]

Durch (R37) wird der im Deutschen mehrdeutige Ausdruck *ein* für unsere Zwecke auf eine bestimmte Weise definiert. Den desambiguierten Ausdruck haben wir mit dem Subskript 'E' versehen. (R39) bis (R42) dienen zur Überführung von Aussagen mit negiertem Prädikat in eine Normalform, in der diese Negation zugunsten einer anderen Quantifizierung eines der Terme aufgegeben ist. (R43) bis (R46) werden in Beweisprozeduren angewendet. Dies ist erlaubt, weil das Symbol '$\Vdash$' und nicht das Symbol '$\Vdash_o$' in ihnen vorkommt.

---

20 Nach unserem Modell ist übrigens *Alle Pudel sind Hunde.* keine analytisch wahre Aussage, sondern ein empirisches Axiom der Basis. Würde man behaupten *Alle Pudel sind Fische.*, so hätte man in keiner Weise eine Konvention der deutschen Sprache verletzt, sondern man hätte offensichtlich ein mangelndes Wissen davon, was in unserem üblicherweise vorausgesetzten Objektbereich Pudel und Fische sind. Analytisch ist in diesem Zusammenhang nur das Verhältnis zwischen einer Aussage mit dem Prädikat *ist*, seinen Argumenten einschließlich der Ausdrücke *alle*, *nicht alle*, *ein*$_E$, *kein* und bestimmten Folgerungsmöglichkeiten. Wenn man der Meinung ist, daß es Aufgabe der strukturellen Lexikologie sei, Relationen wie z. B. die Hyponymie von *Pudel* in bezug auf *Hund* festzustellen, womöglich gar dem Ausdruck *Pudel* das Merkmal *Hund* zuzuordnen, so setzt man damit voraus, daß sich die Lexikologie auf ein ganz bestimmtes Weltmodell festlegen müsse.

4.2 Disjunktive Expansion von Termen

Der Einführung von Termableitungsregeln entspricht die disjunktive Expansion von Termen eines fraglichen Theorems. Jede L-Konstituente einer Frage kann disjunktiv erweitert werden, wenn sich eine Regel findet, zu deren Postzedenz die Konstituente eine Instanz bildet. Wir müssen zu diesem Zwecke freilich die Definition des Ausdrucks *oder*$_D$ erweitern, so daß er zur Verknüpfung von Termen taugt und nicht nur zur Verknüpfung von Aussagen. Da eine disjunktive Verknüpfung von Termen jedoch nichts anderes ist als die Abkürzung einer disjunktiven Verknüpfung von Aussagen, steht dem nichts im Wege.

Durch Anwendung von (R34), (R36) und (R38) ergeben sich aus der ersten Prämisse in (28) folgende Termableitungsregeln:

(30) <*(alle (Hunde))* $\vdash$ *(alle (Pudel))*>
<*(ein*$_E$ *(Pudel))* $\vdash$ *(ein*$_E$ *(Hund))*>
<*(nicht alle (Pudel))* $\vdash$ *(nicht alle (Hunde))*>
<*(kein (Hund))* $\vdash$ *(kein (Pudel))*>

Ist nun fraglich

(31) *((Nicht alle (Hunde)) beißen)?*,

so verläuft die Expansion folgendermaßen:

(32) 0. Stufe: [*((Nicht alle (Hunde)) beißen)*]?
1. Stufe: [*(*[*Nicht alle (Hunde)) oder*$_D$ *(nicht alle (Pudel))*]*)* *beißen*]?
2. Stufe: [*(*[[*(Nicht alle (Hunde)) oder*$_D$ *(nicht alle (Pudel))*] *oder*$_D$ *(kein (Pudel))*]*)* *beißen*]?
3. Stufe: [*(*[[[*(Nicht alle (Hunde)) oder*$_D$ *(nicht alle (Pudel))*] *oder*$_D$ *(kein (Pudel))*] *oder*$_D$ *(kein (Hund))*]*)* *beißen*]?

Beim Übergang von der 1. zur 2. Stufe wurde (R44) angewendet. Wir haben hier den Fall, daß in der Expansion verwendete Regeln Variablen enthalten. Die Zuordnung einer Instanz zum Postzedenz und die Bildung einer Instanz aus dem Antezedenz werfen jedoch keine neuen Probleme auf. Die disjunktive Expansion von Termen ist programmtechnisch sehr praktisch, da die Operationen dabei von lokaler Natur sind und keine umfangreichen Umspeicherungen notwendig machen.

4.3 Evaluierung der Expansion

Die Evaluierung der Expansion einer Frage, in der auch Terme disjunktiv erweitert wurden, kann grundsätzlich weiterhin nach den Regeln (R27) bis (R33) erfolgen. Allerdings darf man nicht vergessen, daß die Variablen in den genannten Regeln für Aussagen stehen, und wir nur für die Zwecke der vorliegenden Arbeit Kategorienangaben weggelassen haben. Bevor die Regeln (R27) bis (R33) angewendet werden können, muß daher eine Entsprechung zwischen den Disjunktionen von Termen in einer Aussage und den Disjunktionen von Aussagen, für welche erstere nur eine Abkürzung darstellen, hergestellt werden. Es ist einfacher, das durch eine geeignete Match-Routine zu leisten als durch die Aufstellung weiterer Ableitungsregeln.

In (28) ist die Aussage *((Kein (Pudel)) beißt).* eines der Axiome. Schon auf der 2. Stufe von (32) bildet diese Aussage eines der implizierten Aussage-Disjunkte. (R27) und (R30) können daher nacheinander angewendet werden. Das Ergebnis der Evaluierung ist das Symbol '■'. Damit ist bewiesen, daß die Konklusion in (27), die ja in (31) als Frage aufgeworfen worden war, zu Recht besteht. Für das noch folgende Beispiel sei vereinbart, daß die Buchstaben *a*, *b*, *c* ... Termkonstanten des Deutschen darstellen. Der Leser kann, wie früher bei den Aussagen, dafür irgendwelche Ausdrücke, die im Deutschen als entsprechende L-Konstituenten auftreten können, setzen. Das Beispiel ist eine Instanz des syllogistischen Modus Barbara. Gegeben seien die Axiome:

(33) *((alle (a)) sind (b)).*
((alle (b)) sind (c)).

Nach (R34) und (R36) werden daraus die folgenden Regeln abgeleitet:

(34) *<(alle(b)) ⊢ (alle(a))> <(nicht alle (a)) ⊢ (nicht alle (b))>*
<(ein$_{E}$*(a)) ⊢ (ein*$_{E}$*(b))> <(kein (b)) ⊢ (kein (a))>*
<(alle(c)) ⊢ (alle(b))> <(nicht alle (b)) ⊢ (nicht alle (a))>
<(ein$_{E}$*(b)) ⊢ (ein*$_{E}$*(c))> <(kein (c)) ⊢ (kein (b))>*

Fraglich sei das Theorem:

(35) *((alle (a) sind (c))?*

Die Expansion von (35) erfolgt nach den Regeln von (34) und ist mit (36) abgeschlossen:

(36) [*(*[[[*(alle (a))*] *oder*$_{D}$ *(alle (b))*] *oder*$_{D}$ *(alle (c))*]*)* *sind (c))*]?

Eines der Aussagen-Disjunkte, die in (36) enthalten sind, lautet *((alle (c)) sind (c))?*. Um dieses fragliche Theorem zu evaluieren, bedarf es nicht der Durchsicht der Datenbasis. Verfügt man über die folgenden Regeln, ist eine Ableitung von '■' unmittelbar möglich:

(R47) *<((alle (α)) sind (α))* $\Vdash_{\circ}$ ■>
(R48) *<((ein*$_{E}$ *(α)) ist (α))* $\Vdash_{\circ}$ ■>

Damit bestätigt sich, was wir schon in 3.2 angenommen haben: Funktionale Aussagen, die in Gestalt von Ableitungsregeln algorithmisiert worden sind, brauchen in die Datenbasis nicht aufgenommen werden. Die Basis kann weiterhin auf elementare Aussagen und Regeln beschränkt bleiben.

Aus dem Theorem *((alle (a)) sind (c)).*, das wir soeben abgeleitet haben, ließen sich seinerseits wieder Termableitungsregeln erzeugen. Diese wären jedoch redundant, so daß auf sie verzichtet werden kann. Alles, was aus ableitbaren Theoremen folgt, ist mit Hilfe der gegebenen Ableitungsregeln bereits deduzierbar.[21]

Die Konklusionen aller bekannten syllogistischen Modi sowie beliebige Kombinationen aus jenen Schlußfiguren lassen sich mit den Regeln (R34) bis (R48) problemlos verifizieren. Das deutet darauf hin, daß ein regellogisches Deduktionssystem auf der Basis einer natürlichen Sprache einen aussichtsreichen Ansatz darstellt. Darüberhinaus mag das geschilderte Modell auch für eine handlungstheoretisch orientierte Semantik von Interesse sein.

21 Der Beweis hierfür muß aus Platzgründen unterbleiben.