

# HAM-RPM: EIN REDEPARTNERMODELL ALS SIMULATIONS-PROGRAMM

Walther v. Hahn, Dieter Henskes, Wolfgang Hoepfner, Wolfgang Wahlster<sup>1</sup>

## 0. Forschungsgebiet

HAM-RPM leistet als natürlichsprachliches Artificial Intelligence System einen Beitrag zu einem interdisziplinären Forschungsgebiet zwischen Linguistik, Informatik und Psychologie. Dabei greift es besonders auf die Gebiete Computerlinguistik, Künstliche Intelligenz und Psychologie der Informationsverarbeitung zurück. In den letzten Jahren sind besonders in den USA etwa mit dem Ziel einer Anwendung innerhalb von Informationssystemen zahlreiche vergleichbare Systeme entwickelt worden.

Nach der Typologie von WILKS 1975 gehört HAM-RPM zur Klasse der semantisch orientierten Systeme, bei denen das Verstehen natürlicher Sprache im Vordergrund steht. Diese sind von den sogenannten strukturell orientierten Systemen zu unterscheiden, die versuchen, mit syntaktischen Methoden möglichst direkt zu einer problemorientierten Repräsentation zu kommen und dabei natürlichsprachliche Probleme weitgehend unberücksichtigt lassen. Die Verarbeitung dieser internen Repräsentation stellt die eigentliche Leistung dieser frühen Systeme dar.

Nach WINOGRAD 1974 müßte man HAM-RPM zu den Systemen der zweiten Generation rechnen, die sich durch komplexe linguistische Operationen und die Einbeziehung von Erfahrung, Wissen und 2 Problemlösungsmethoden auszeichnen.

## 1. Kommunikationstheoretische Beschreibung

1.1. Während eines Programmlaufs kommuniziert ein menschlicher Dialogpartner (im folgenden MD genannt) mit dem Simulationsmodell über eine Teilwelt. Die Bezeichnung "Mini-" oder "Mikrowelt" wird nicht verwendet, da es sich bei den betrachteten Teilwelten nicht um logisch abgeschlossene, sondern erweiterbare Strukturen ohne Modellcharakter, also um Fragmente handelt.

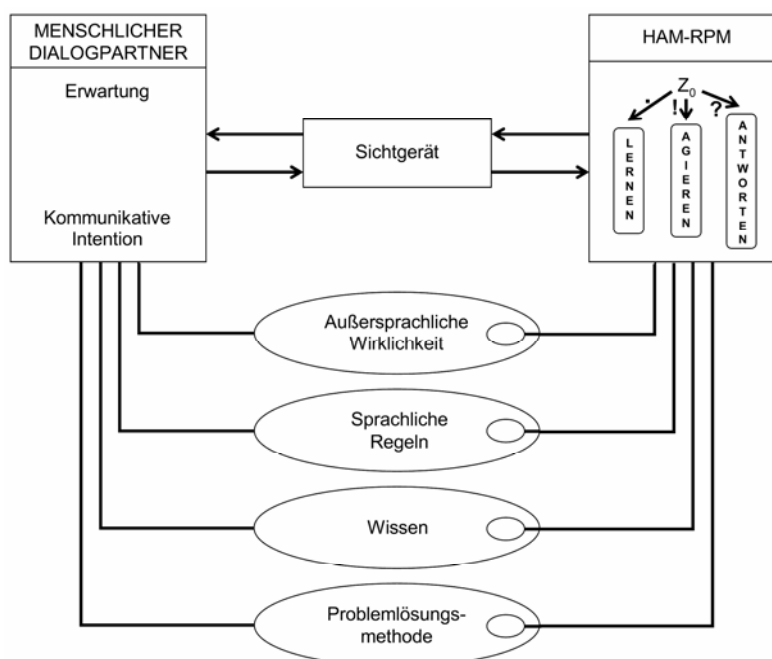


Fig. 1 Kommunikationsmodell

Wie in Fig. 1 dargestellt, dient ein Sichtgerät mit Schreibmaschinentastatur als Kanal. Informationen, die durch Mittel wie Sprechrhythmus, Pausen, Gestik oder Mimik übermittelt werden, können im Gegensatz zu einer natürlichsprachlichen face-to-face-Situation in dem Modell deshalb nicht berücksichtigt werden. Eine sprachliche Verständigung wird dadurch ermöglicht, daß die beiden Kommunikationsteilnehmer über einen nichtleeren Durchschnitt von Codes verfügen. Außerdem sind die jeweils referierten Teile der außersprachlichen Wirklichkeit, das Wissen und die Problemlösungsmethoden des simulierten Dialogpartners (im folgenden MM genannt) echte Teilmengen der entsprechenden Bereiche von MD. Um HAM-RPM zu einem Anwendungsprogramm auszubauen, müßte man das System mit MD nicht bekannten Fakten und Problemlösungsmethoden ausstatten.

Der während eines Dialogs ablaufende Kommunikationsprozeß ist asymmetrisch, da eine kommunikative Intention und Partneereinschätzung nur auf der Seite des MD, nicht aber bei MM vorhanden ist. In einer natürlichen Frage-Situation würde MD eine Fragehandlung vollziehen, weil er x nicht weiß, aber x aus dem Grund y wissen möchte, und daher MM durch seine Frage dazu auffordert, ihm dieses Wissen zu vermitteln. MD verfügt aber aus den oben genannten Gründen stets über alle Informationen, die MM gespeichert hat. Da die spezielle Motiviertheit des MD außerdem lediglich im Test und der Weiterentwicklung des Simulationsmodells beruht, handelt es sich um sog. pervertierte Fragen in Form von Test- und Fangfragen. Eine weitere Einschränkung besteht darin, daß nur MD die Möglichkeit hat, den Dialog zu eröffnen oder abzubrechen, ohne daß ihn dafür seitens MM Sanktionen erwarten.

Vor Beginn des Dialogs ist RPM in einem Grundzustand  $z_0$  (vgl. Fig. 1). Als Reaktion auf die Dialogeröffnung durch eine Sprechhandlung von MD geht RPM in einen der Zustände LERNEN (u.a. Prüfung, ob Nachricht neue, relevante und glaubhafte Information enthält; Konsistenzprüfung mit vorhandenem Wissen; Integration in bestehende Wissensstrukturen), AGIEREN (z.B. kontrollierte Zustandsveränderung innerhalb der Teilwelt) oder ANTWORTEN über. Die Eingabe muß vom Benutzer durch entsprechende Interpunktion eindeutig als ein bestimmter Sprechakt klassifiziert werden. Indirekte Sprechakte werden vom System nicht erkannt.

1.2. Im weiteren wird nur der Zustand ANTWORTEN betrachtet. Bei bestimmten Fragen können allerdings auch Aspekte anderer Zustände eine Rolle spielen. So werden z.B. auf die Frage Wo könnte x stehen? eine Folge von "Pseudoaktionen" wie etwa die systematische Variation der Position von x und MM1 simuliert, um die gesuchten "möglichen Welten" zu finden.

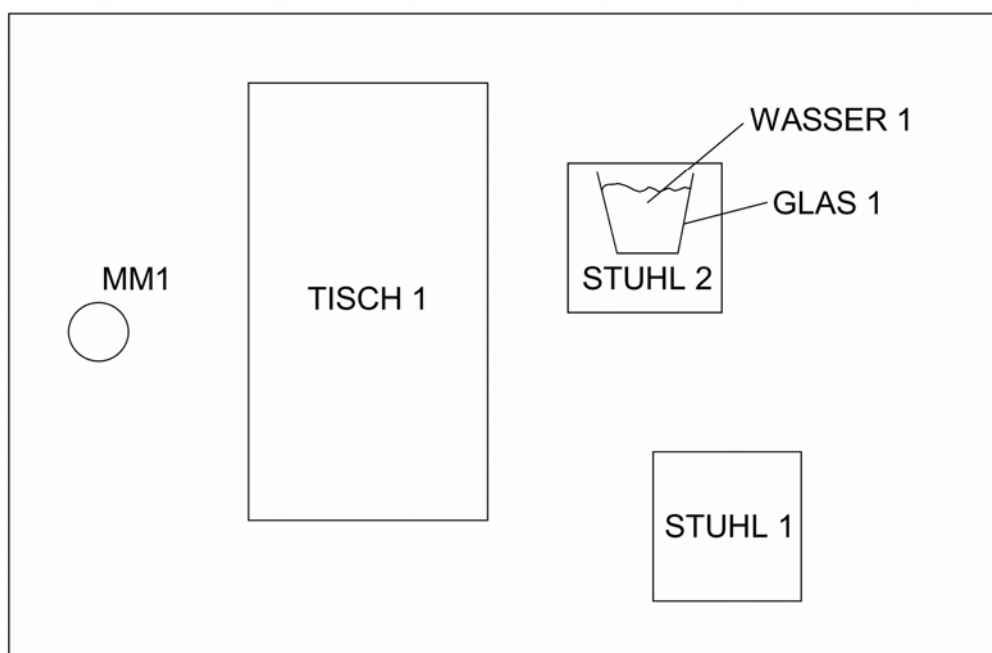


Fig. 2 Räumliche Situation

Für das folgende soll noch eine weitere Festlegung getroffen werden: eine Teilwelt T sei definiert als eine Menge von Referenzen auf physikalische Objekte. Für die weitere Darstellung wählen wir ein spezielles T:=

{RAUM1, MM1, TISCH1, STUHL1, STUHL2, WASSER1, GLAS1, WAND1, BODEN1}.

Die räumlichen Beziehungen zwischen den genannten Objekten werden innerhalb eines geometrischen Modells (GEOMOD) für eine bestimmte Ausgangssituation festgelegt. Eine solche Situation wird in Fig. 2 dargestellt.

```
1 *(IST D- STUHL KAPUTT ?)
  (WELCH- STUHL ?) *STUHL2
  (JA)

2 *(WELCH- GEGENSTAND SIND MOEBEL ?)
  (STUHL TISCH)

3 *(KOENN- DU D- GLAS NICHT GREIF- ?)
  (DOCH !)

4 *(WELCH- AGGREGATZUSTAND HAB- D- RAUM ?)
  (D- RAUM1 HAB- KEIN- AGGREGATZUSTAND)

5 *(WAS IST IN D- WASSER ?)
  (NICHTS)

6 *(WELCH- STUHL IST GASFOERMIG ?)
  (KEIN- STUHL IST GASFOERMIG)

7 *(TRINK- DU OFT BIER ?)
  (WELCHE WORTART HAT)OFT? *(ADV)
  (ES GIBT KEIN- BIER)

8 *(WELCH- MOEBEL SIND FEST ?)
  (JED- MOEBEL SIND FEST)

9 *(WAS STEH- NICHT AUF D- WASSER ?)
  (ALLES)

10 *(KOENN- DU D- GLAS TRINK- ?)
  (NEIN)

11 *(WARUM ?)
  (GLAS1 IST NICHT FLUESSIG !)
```

Fig. 3 Protokoll eines Dialogs in Kurzform

Es soll hier gleich auf eine besondere Schwierigkeit hingewiesen werden. Auf die Frage Wo steht der Tisch? könnte MM in der gegebenen Situation z.B. mit Vor dem Stuhl2 oder Vor dem Glas antworten. Letzteres ist aber nur dann eine gewünschte Antwort, wenn das Glas im laufenden Dialog bereits eine gewisse kommunikative Relevanz erlangt hat. RPM muß also über die kommunikative Kompetenz verfügen, aus einer Menge möglicher Antworten die in der jeweiligen Situation relevanten auszuwählen. Im Simulationsprogramm wird dies durch entsprechende Operationen auf dem Wert einer Variablen EINFLUSS angenähert, die außer den Koordinaten und einer Größenklasse für jedes Objekt definiert ist. So kann z.B. im Lauf eines Dialogs EINFLUSS (GLAS1) so vergrößert werden, daß das Glas bei entsprechenden Vergleichen mit herangezogen wird.

```

*(KOENN- DU D- GLAS AUF D- WASSER STELL- ?)
(ERSETZEN DU-MM)
ENTSCHEIDUNGSFRAGE
((MOD (KOENN-)) (NOM (MM1)) (NOM (GLAS1)) (PRP (AUF))
(NOM (WASSER1)) (VRB (STELL-)))
POTENTIAL
(TRANSFORMATION ;)
((NOM (MM1)) (VRB (STELL-)) (NOM (GLAS1)) (PRP (AUF))
(NOM (WASSER1)))
(SATZ AKZEPTIERT ;)
(S (NP (NOM (MM1))) (VP (VPA (VRB (STELL-)) (NP (NOM
(GLAS1))) (PP (PRP (AUF)) (NOM (WASSER1))))))
S
  NP
  VP
    VPA
      VRB
      NP
      PP
        PRP
        NOM

```

((MM1) (GLAS1) (STELL-))  
(DEKOMPOSITIONSERGEBNIS : (GLAS1) (WASSER1) (AUF) NEGATION: NIL)  
NIL  
(NEIN)  
(12622 MSEC)

Fig. 4 Beispielsatz in Langform

## 2. Performanzbeispiele

Fig. 3 vermittelt einen Eindruck von der Leistung des Programms beim Verstehen und Beantworten von Fragen. Die Ausgabe der Antwort kann in zwei Modi erfolgen:

- nur die Oberflächenstruktur der Antwort bzw. Rückfrage wird ausgegeben (vgl. Fig. 3)
- der gesamte Prozeß des Verstehens und der Beantwortung wird protokolliert (Fig. 4).

Die Zeitspanne zwischen der Eingabe einer Frage und deren Beantwortung kann im Protokoll ausgegeben werden. Es wurde eine Proportionalität zwischen der intuitiven Einschätzung des Schwierigkeitsgrades einer Frage und der vom Programm benötigten Verarbeitungszeit angestrebt.<sup>4</sup>

## 3. Systemtheoretische Beschreibung

3.1. Ein erster Zugang zur Beschreibung des im Dialog (Fig. 3) beobachtbaren Verhaltens ist die Darstellung von HAM-RPM als rückgekoppeltes, dynamisches System (vgl. Fig. 5). Formal können die vom Benutzer explizit manipulierbaren Daten (DATENBASIS) wie die sprachlichen Regeln, die Situationsbeschreibung, das Wissen und die Problemlösungsmethoden, über die MM verfügt, als Eingabe in das System betrachtet werden. Durch Fragen des MD und dessen Antworten auf Rückfragen zum Zeitpunkt  $t$  (vgl. Fig. 3, Satz 7) kann nicht nur eine Antwort, sondern auch eine Änderung der Eingabedaten zum Zeitpunkt  $t+1$  (Rückkopplung) ausgelöst werden.

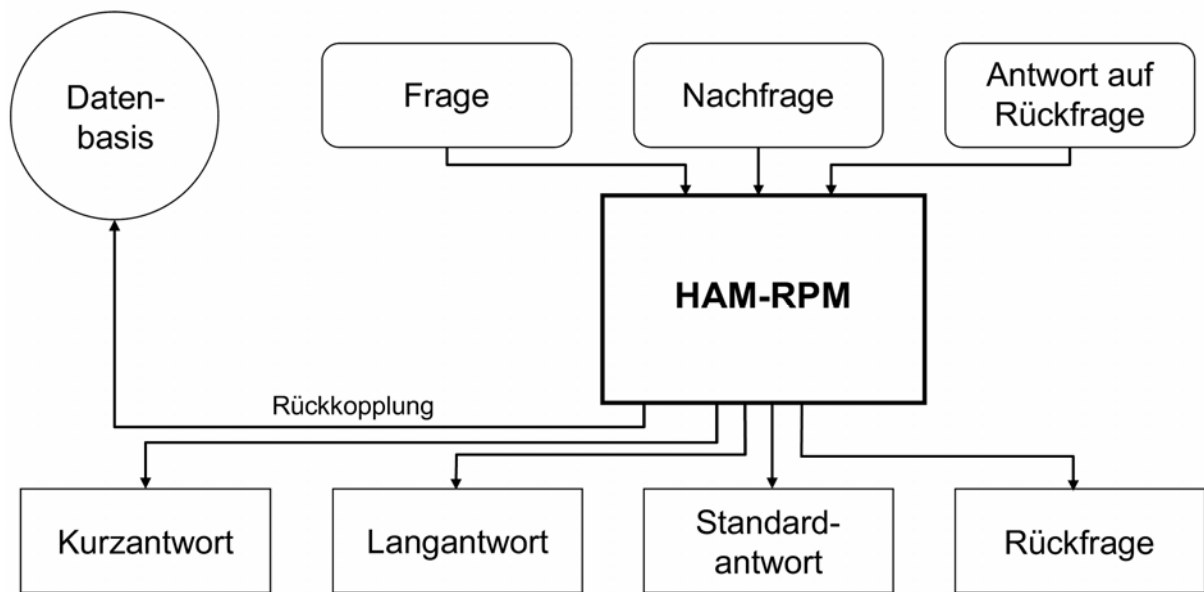


Fig. 5 Systemtheoretisches Modell

So wird erreicht, daß die gleiche Frage in verschiedenen Phasen des Dialogs unterschiedlich beantwortet wird (Dynamik). Ein triviales Beispiel für die Abhängigkeit der Antwort vom Ko-Text ist die Beantwortung von Nachfragen des MD (vgl. Fig. 3, Satz 11), da diese Fragen zur Begründung der vorausgehenden Antwort des MM auffordern.<sup>5</sup>

3.2. Auf eine Frage des MD kann das System mit einer kurzen prägnanten Antwort in Form einer Ellipse (hier Kurzantwort genannt) oder einer ausführlichen Antwort in Form eines Textes (hier Langantwort genannt) reagieren. Das System hat in seiner derzeitigen Ausbaustufe noch nicht die kommunikative Kompetenz zu entscheiden, ob in einer Situation  $x$  zum Zeitpunkt  $t$  eine Kurzantwort oder eine Langantwort kommunikativ adäquat ist. In HAM-RPM wird eine Partnertaktik von MM also nicht berücksichtigt.

Im allgemeinen ist die Generierung einer Kurzantwort aufwendiger als die einer Langantwort, da eine Informationskonzentration und -auswahl nach nur schwer operationalisierbaren Kriterien vorgenommen werden muß, während die Langantwort lediglich eine Strukturierung der Information und die Erzeugung von Kohärenz erfordert.

3.3. In einigen wohldefinierten Situationen generiert das System Standard-Antworten, die zur Metakommunikation gerechnet werden können. Beispiele:

Es gibt kein  $x$  !  
(vgl. Fig. 3, Satz 7)  
Unsinnige Frage

Existenzielle Präsupposition in Eingabe ist verletzt.

Eingabe enthält Widerspruch oder ist nicht sinnvoll interpretierbar.

Bitte Frage nochmal anders formulieren  
Weiß ich nicht !

Eingabe ist syntaktisch nicht wohlgeformt.

Durch Inferenzen auf vorhandenem Wissen nicht beantwortbare Frage.

#### 4. Datenstrukturen

4.1. Auf einer elementaren Ebene gibt es in der Datenbasis des Modells nur die Struktur der Eigenschaftsliste. Faßt man den Begriff der Datenstruktur weiter und bezieht die Menge aller auf einem Objekt definierten Zugriffs- und Verarbeitungsfunktionen in die Definition ein, so bietet HAM-RPM dem Benutzer zahlreiche Datenstrukturen auf syntaktischer, semantischer und pragmatischer Ebene an (vgl. Fig. 6).

Die vielfältigen Manipulationsmöglichkeiten der vorhandenen Datensätze mit Hilfe eines Texteditors ergeben eine im Gegensatz zu den meisten vergleichbaren Systemen hohe Flexibilität und Adaptibilität des Programms, die es als Forschungsinstrument innerhalb der AI und der Linguistik, als Demonstrations- und Übungsprogramm im Hochschulunterricht und für den praktischen Einsatz als Frage-Antwort-System geeignet erscheinen lassen.

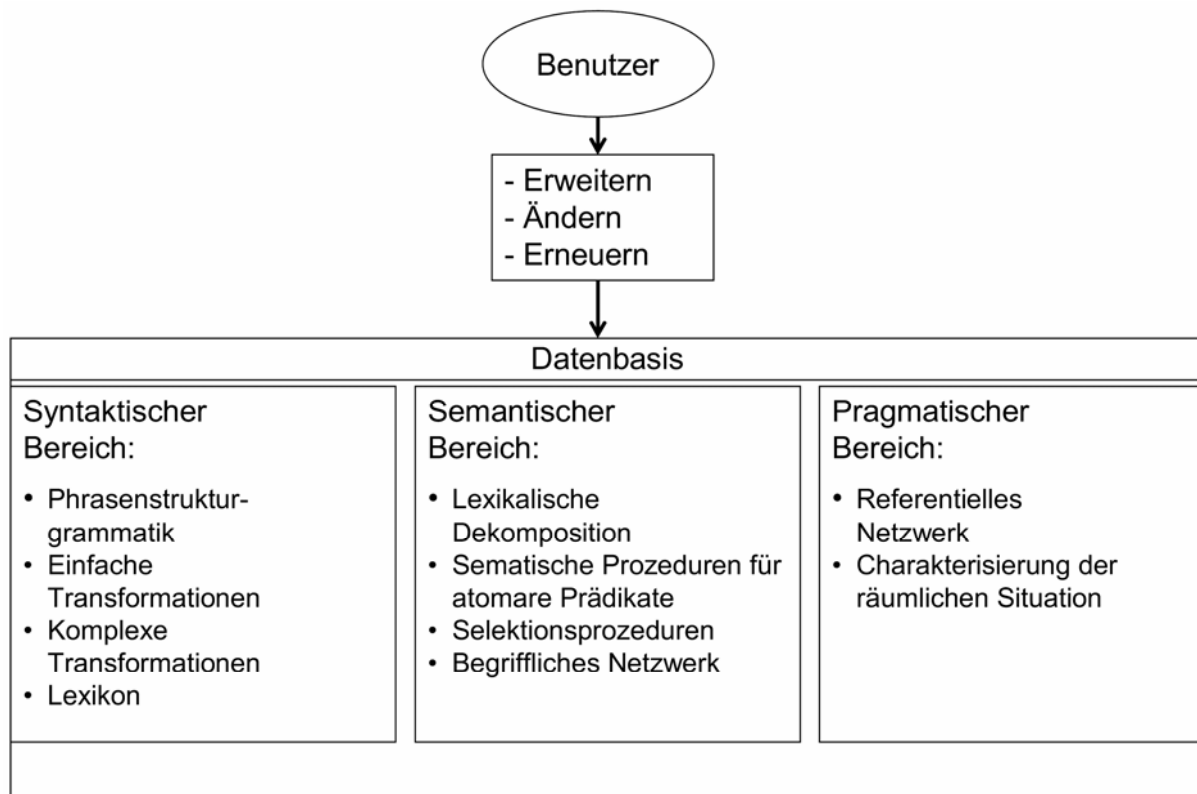


Fig. 6 Struktur der Datenbasis

4.2. Als Basisgrammatik kann der Benutzer beliebige  $\epsilon$ -freie Typ-2-Grammatiken (kontextfreie Grammatiken) in die Datenbasis<sup>7</sup> eingeben. Der bottom-up-Parser arbeitet von links nach rechts auf der präterminalen Kette des zu analysierenden Satzes. Da weder Chomsky- noch Greibach-Normalform benutzt werden müssen, fällt das bei den üblichen Parsern oft auftretende Problem semantisch nicht interpretierbarer Hilfssymbole weg. Eine Produktion der Form  $VPA \rightarrow VRB + NP + PP$  wird in der Datenbasis als (SYNTAX (VPA (VRB NP PP)) VRBNPPP) dargestellt. Einfache Transformationen auf der präterminalen Kette in der Form

SD:	IPRN	VRB	DET	NOM
	1	2	3	4
SC:	3	4	2	1

werden als (TRANSF (3 4 2 1) IPRNVRBDET NOM) beschrieben.

Komplexe Transformationen, die Abbildungen von Strukturbäumen auf Strukturbäume leisten, werden als symbolische Ausdrücke in einer komfortablen Pattern-Matching-Sprache (vgl. WAHLSTER 1975) dargestellt.

Mit Hilfe von Operationen auf einem einfachen Lexikon, das aus Strukturen wie (STUEHLE (WORTART (NOM PL))) aufgebaut ist, wird die präterminale Kette des Eingabesatzes erzeugt.

4.3.1. Im semantischen Bereich kann das in der generativen Semantik entwickelte Verfahren der lexikalischen Dekomposition eingesetzt werden. Die Dekompositionsregeln werden in prozeduraler Form in die Datenbasis eingebettet. Im Lexikon wird für jedes dekomponierbare Lexem unter der

Eigenschaft (Property) DEKOMPOSITION als Wert (Value) der Name der entsprechenden Dekompositionsprozedur aufgelistet. So wird z.B. die Dekomposition von leer in nicht+enthält durch den Lexikoneintrag (LEER (WORTART (ADJ) (DEKOMPOSITION (LEER-DEK))) und die Dekompositionsregel in Form einer, in diesem Fall einfachen Prozedur mit dem Namen LEER-DEK vollständig beschrieben.

Für jedes atomare Prädikat wie ENTHAELT existieren semantische Prozeduren, die die Verarbeitung in GEOMOD anstoßen oder Inferenzen und Suchprozesse auf Wissensspeichern auslösen können. Mit Hilfe der Dekomposition kann die Anzahl der semantischen Prozeduren klein gehalten werden.

Selektionsbeschränkungen im Sinne der interpretativen Semantik kann der Benutzer als Selektionsprozeduren formulieren, die auf semantischen Netzwerken arbeiten. So wird durch die zu trink- gehörige Selektionsprozedur erreicht, daß die Antwort auf Frage 10, Fig. 3 Nein lautet und die Nachfrage mit der Ausgabe der verletzten Selektionsrestriktion beantwortet wird.

4.3.2. Neben den genannten prozeduralen Beschreibungsmöglichkeiten der Semantik von Lexemen werden im semantischen und pragmatischen Bereich Netzstrukturen zur Speicherung von Information eingesetzt. Formal betrachtet, handelt es sich bei diesen Strukturen um bewertete, gerichtete Graphen. Die semantischen Einheiten, die durch die Knoten des Graphen repräsentiert werden, stehen zueinander in verschiedenen semantischen Relationen, mit denen die Kanten bewertet sind. In HAM-RPM werden nur sog. tiefenorientierte semantische Netze verwendet, d.h. als Relationen sind nicht beliebige zweistellige Oberflächenprädikate, sondern nur eine gewisse Zahl atomarer Relationen zugelassen.

Im begrifflichen Netzwerk werden die RPM zur Verfügung stehenden Begriffe (Types) durch die Menge der Relationen charakterisiert, in denen sie mit anderen Begriffen stehen. Die Knoten des begrifflichen Netzwerks werden durch Oberflächenstrings repräsentiert, während die Kanten mit atomaren Relationen ohne direkte Oberflächenentsprechung bewertet sind.

Folgende atomare Relationen sind definiert:

- V ::= hat als möglichen Wert
- E ::= hat als Eigenschaft
- D ::= hat als Denotat
- T ::= hat als Teil
- K ::= ist Teilmenge von
- U ::= ist Obermenge von

Obwohl die gleichzeitige Verwendung von U und K redundant ist, wurde sie aus Gründen der Effektivität von Suchprozessen beibehalten. Im begrifflichen Netzwerk (vgl. Fig. 7) ist also das eher statische Wissen gespeichert. Nur im (noch nicht implementierten) Zustand LERNEN könnte MM1 durch Übernahme von Begriffen des MD oder eigene Abstraktion das begriffliche Netzwerk verändern.

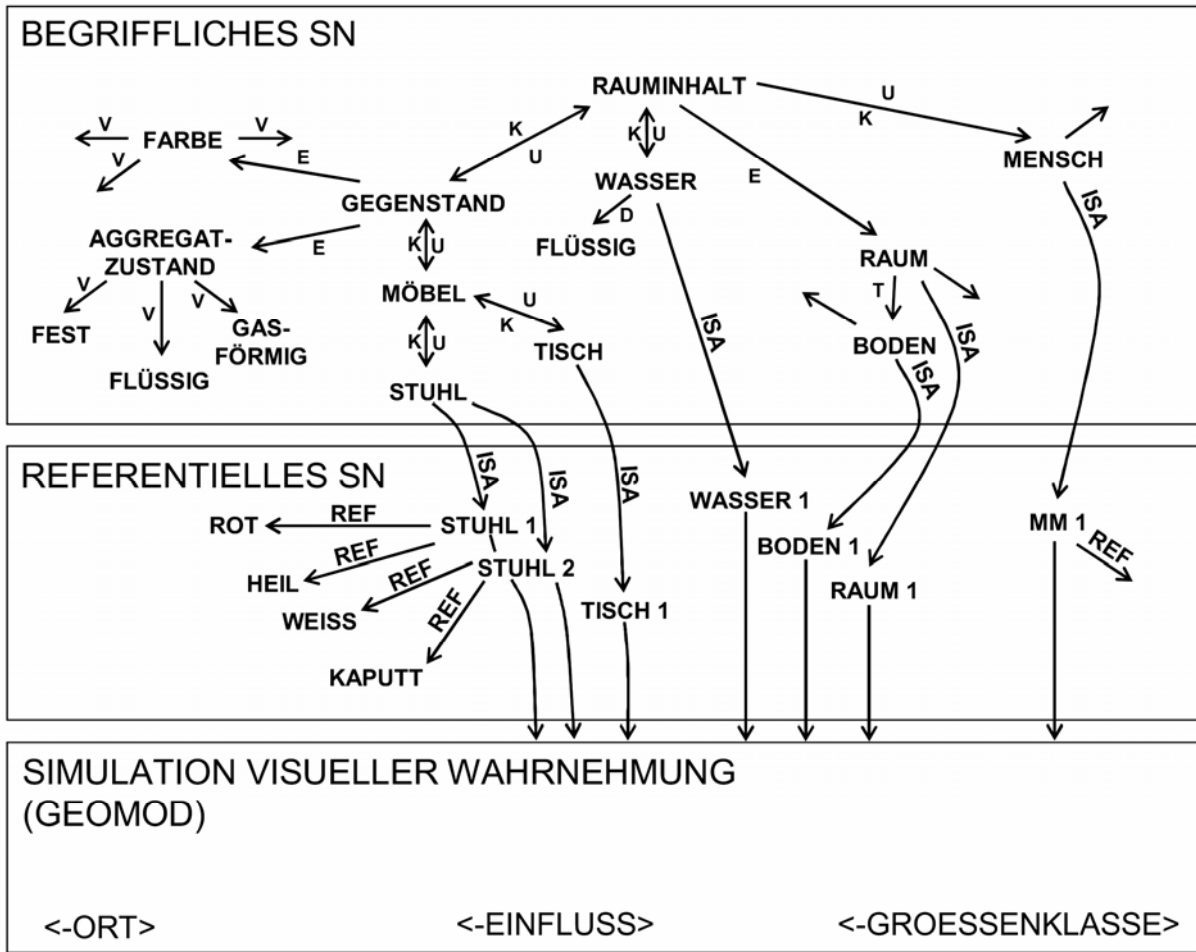


Fig. 7 Netzwerke und topologisches Teilweltmodell

4.3.3. Durch die Relation ISA wird eine Kopplung zwischen dem begrifflichen Netz und einem referentiellen Netz hergestellt (vgl. Fig. 7). ISA ist als die Beziehung zwischen den im begrifflichen Netz auftretenden Types und den als Knoten des referentiellen Netzwerks vorgesehenen Tokens definiert. Die Tokens werden aus mnemotechnischen Gründen als Konkatination der entsprechenden Oberflächenstrings mit einer zur eindeutigen Identifikation notwendigen Ziffer dargestellt. Tokens können durch ihren aktuellen Zustand, der durch die atomare Relation REF gekennzeichnet ist, genauer spezifiziert werden. So wird in den durch Fig. 7 dargestellten Netzfragmenten STUHL1 in der aktuellen Situation durch die Attribute 'rot' und 'heil', STUHL2 dagegen durch die Merkmale 'weiß' und 'kaputt' charakterisiert.

Bei einer mengentheoretischen Interpretation können die im referentiellen Netzwerk auftretenden Tokens als Elemente der mit ihnen im Pfeildiagramm durch ISA verbundenen "fuzzy sets" des begrifflichen Netzwerks aufgefaßt werden.<sup>9</sup>

Das begriffliche Netzwerk wird also referentiell verankert. Im referentiellen Netzwerk werden diejenigen Teile der aktuellen Situation gespeichert, die kurzfristig sprachlich im Gedächtnis (Datenbasis) repräsentiert werden. Da sich die aktuelle Situation durch Aktionen des MM ständig ändern kann (z.B. MM zerstört den STUHL1), ist das referentielle Netzwerk im Gegensatz zum begrifflichen Netzwerk dynamisch.

4.3.4. Die Verbindung jedes Tokens mit nicht natürlichsprachlich codierten Daten, die einem System zur eingeschränkten Simulation visueller Wahrnehmung (GEOMOD) zur Verfügung gestellt werden, repräsentiert die referentielle Beziehung zur außersprachlichen Teilwelt. Die Eigenschaften ORT, EINFLUSS und GRÖSSSNKLASSS, die jedem Token in der Datenbasis über einen numerischen Wert zugeordnet sind, dienen der Beschreibung der jeweils aktuellen räumlichen Situation. Die



Angaben unter ORT und GRÖSSENKLASSE werden zur Bestimmung der geometrischen Beziehungen zwischen Objekten eingesetzt, während der Wert der Variablen EINFLUSS das Antwortverhalten steuert (vgl.1.2.).

4.3.5. Sowohl das begriffliche und referentielle Netzwerk als auch die Daten zur Charakterisierung der räumlichen Situation werden in der Datenbasis als Eigenschaftslisten dargestellt. So erscheinen beispielsweise der Knoten für den Begriff GEGENSTAND und die mit ihm direkt verbundenen Kanten als die Liste (GEGENSTAND (K (RAUMINHALT) U (GLAS MOEBEL) E (FARBE ZUSTAND) D (FEST))).

Beim Entwurf der Datenbasis wurden neben rein deskriptiven Ansätzen (Basisgrammatik, Transformationen, Lexikon, Netzwerke, GEOMOD-Daten) auch die Ideen des "procedural embedding of know-ledge" (lex. Dekomposition, sem. Prozeduren für atomare Prädikate, Selektionsprozeduren) berücksichtigt. Der prozedurale Ansatz, der gegenüber dem deskriptiven den Nachteil hat, daß er höhere Anforderungen an den Benutzer stellt (Beherrschen von LISP 1.6), wurde an den Stellen gewählt, wo es auf erhöhte Flexibilität ankam.<sup>10</sup>

## 5. Funktionsweise

5.1. Im folgenden sollen die Prozesse beschrieben werden, die bei Eingabe der Oberflächenstruktur einer Frage durch verschiedene Operationen auf der oben beschriebenen Datenbasis schließlich die Ausgabe der Oberflächenstruktur der Antwort bewirken (vgl. Fig. 8). Die Grundstruktur des Programms entspricht der eines Interpreters für Programmiersprachen.

Es können drei Verarbeitungsphasen unterschieden werden. Zunächst wird in einem Analyseteil ausgehend von der Oberflächenstruktur der Frage eine semantische Repräsentation der Eingabe erzeugt. Durch die Antwortbasis wird dann zur Tiefenstruktur der Frage eine semantische Repräsentation der entsprechenden Antwort generiert. Schließlich wird in einem Generierungsteil die Tiefenstruktur der Antwort in eine Oberflächenstruktur überführt.

Neben diesem globalen linearen Kontrollfluß wirken in jeder Phase eine Vielzahl von syntaktischen, semantischen und pragmatischen Prozessen zusammen. Um der Interdependenz dieser Prozesse beim menschlichen Verstehen näher zu kommen, wurde beim Entwurf des Systems darauf geachtet, daß sich Syntax, Semantik und Pragmatik in geeigneter Weise verzahnen und überlappen können.

5.2. Dieses Konzept soll am Beispiel der Analysephase genauer erläutert werden. Nach dem Einlesen der Frage, einer Ersetzung von Du durch MM1 und der Generierung der präterminalen Kette wird ermittelt, ob es sich bei der Eingabe um eine W-Frage oder eine Entscheidungsfrage handelt. Die präterminale Kette (vgl. Fig. 4) wird nun semantisch-pragmatisch vorverarbeitet. So werden z.B. die existentiellen Präsuppositionsbedingungen überprüft. In Fig. 3, Beispiel 7 wird festgestellt, daß es in der dem MM bekannten Teilwelt kein Bier gibt. Dies führt unmittelbar zu einer Ausgabe, die den globalen Kontrollfluß unterbricht. Eine Vorverarbeitung im referenz-semantischen Bereich ist die Disambiguierung von Nominalgruppen der Form: bestimmter Artikel +Nomen (vgl. Beispiel 1). Hier fordert MM den MD durch eine Rückfrage zur Disambiguierung der Frage auf. Diesen vorzeitigen Unterbrechungen der Analyse ähnelt das häufig zu beobachtende "in das Wort fallen" beim Auftreten von Unklarheiten in natürlichen Dialogen.

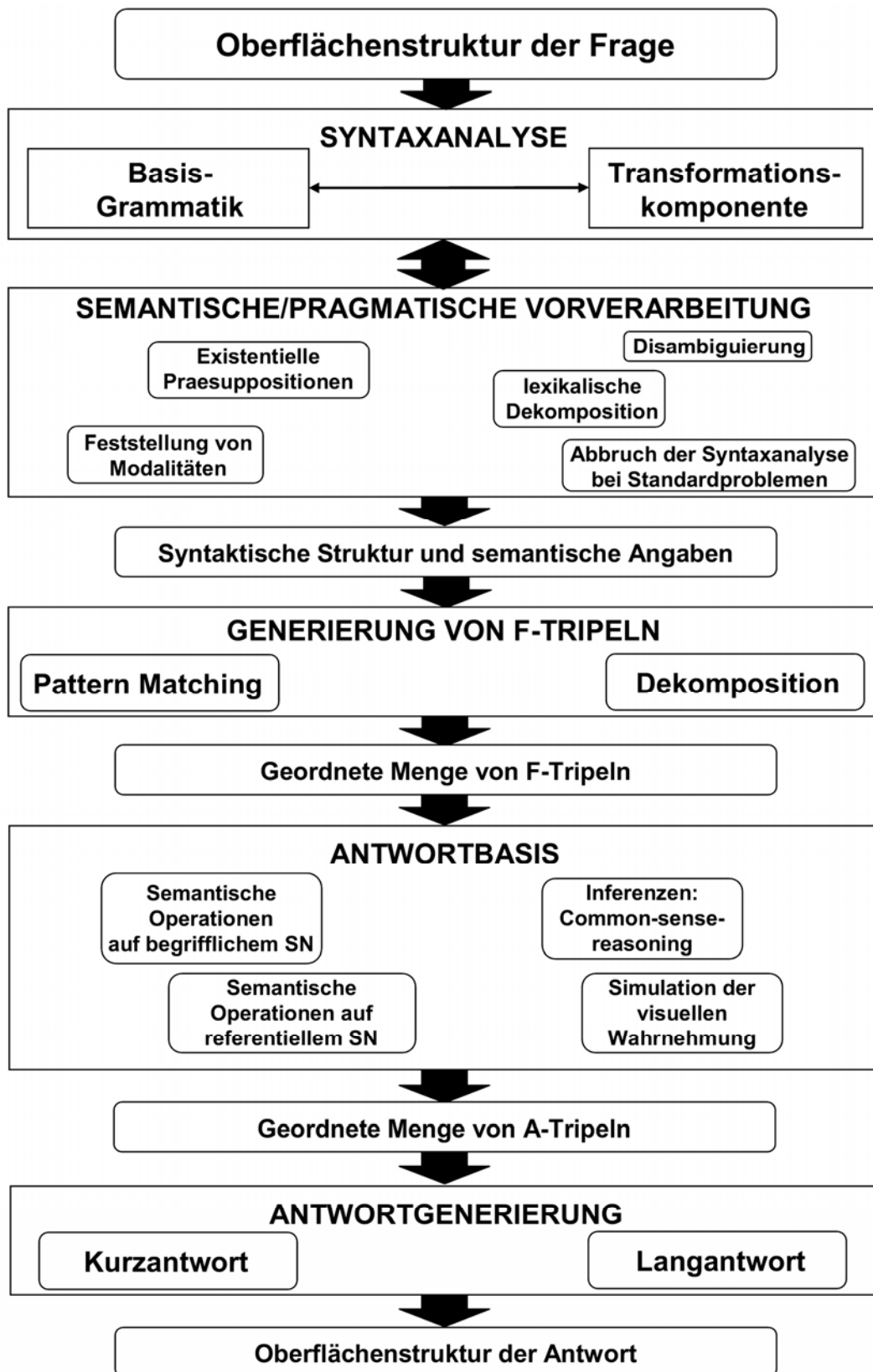


Fig. 8 Verarbeitungsphasen

Nach dem Schema der Fillmore'schen Kasusgrammatik wird der Eingabesatz in eine Modalitätskomponente und eine Proposition aufgeteilt (vgl. Fig. 4). Durch Pattern Matching wird festgestellt, ob ein Fragetyp vorliegt, dessen Semantik nach einem einfachen Schema erschlossen

werden kann (vgl. Welch-Fragen, Beispiel Nr. 2, 4, 6, 8). Solche Fragen können ohne den Aufwand einer weiteren Analyse mit Hilfe einfacher Operationen auf der Datenbasis beantwortet werden.

Einfache Transformationen (vgl. Fig. 4) überführen den propositionalen Teil der Frage in die Struktur eines Aussagesatzes. Auf der erhaltenen Struktur arbeitet der Basisparser, welcher in den verschiedenen Phasen der Reduktion wieder Prozeduren im semantischen und pragmatischen Bereich auslöst. Die Syntaxanalyse wird also nur solange vorangetrieben, bis die Semantik der Eingabe ermittelt ist. Schließlich wird als Zwischencode die Syntaxstruktur in polnischer Notation und als Strukturbaum ausgegeben.

Durch auf den Zwischencode angewandte Pattern-Matching-Prozeduren und Dekompositionsregeln wird nun die semantische Repräsentation der Frage als geordnete Menge von Frage-Tripeln (F-Tripel) erzeugt. Die Positionen eines Tripeis werden durch Transformationen nacheinander aufgefüllt. In Fig. 4 wird als erstes F-Tripel ((MM1) (GLAS1) (STELL-)) und als zweites Tripel ((GLAS1) (WASSER1) (AUF)) generiert.

5.3.1. Die Reihenfolge der F-Tripel untereinander entspricht einer Präsuppositionsabhängigkeit, die in der Antwortbasis zu berücksichtigen ist. Falls MM das Glas in der aktuellen Situation nicht erreichen könnte, würde die Frage schon ohne Bearbeitung des zweiten Tripels negativ beantwortet. Da unter Berücksichtigung der Modalität im Beispiel das erste Tripel positiv beantwortet wird, muß auch das zweite Tripel in der Antwortbasis überprüft werden. Dazu wird GEOMOD aktiviert. Dort wird nach einer "möglichen Welt" gesucht, in der das Glas auf dem Wasser steht. NIL als Ergebnis dieser Suche wird durch eine einfache, nur von der Modalität abhängige Transformation in die Kurzantwort Nein überführt.

5.3.2. GEOMOD, als ein Modell zur Simulation der Wahrnehmung räumlicher Beziehungen, leistet eine Abbildung von F-Tripeln der Form ((LWAS) (LWO) (LWIE)) auf eine geordnete Menge von Antworttripeln der Form (WAS WO WIE). LWAS ist eine Liste von Tokens, über die eine Aussage gemacht werden soll. Die Liste LWO enthält die Objekte, zu denen die Gegenstände aus LWAS in Beziehung gebracht werden sollen. LWIE enthält die topologischen Relationen, die zwischen den Elementen von LWAS und LWO zu überprüfen sind. Als Wert von LWAS, LWO oder LWIE tritt im Fall von W-Fragen das Atom NIL auf. Dabei stellen alle kombinatorischen Möglichkeiten Repräsentationen von sinnvollen Anfragen dar.

Beispiele:

((STUHL1) NIL (VOR))	Vor was steht der Stuhl?
(NIL NIL NIL)	Was steht wo?
((STUHL1) NIL NIL)	Wo steht der Stuhl?
((STUHL1) (TISCH1) NIL)	Wie steht der Stuhl zu dem Tisch?

Zur Beantwortung der Anfrage aktiviert GEOMOD prozedurale Definitionen räumlicher Beziehungen wie vor, hinter, auf, über usw. Diese Prozeduren greifen auf die Koordinaten der entsprechenden Gegenstände zurück. Außerdem entnehmen die Prozeduren dem begrifflichen Netzwerk Informationen über die Gestalt und die physikalische Beschaffenheit der einzelnen Objekte. Es wird also nicht eine Datenbasis abgesucht, die eine komplette sprachliche Beschreibung der räumlichen Situation enthält, sondern es wird jeweils eine Verbalisierung der simulierten visuellen Wahrnehmung vollzogen.

5.3.3. Viele Fragen können ohne Aktivierung von GEOMOD beantwortet werden. Gezieltes Suchen, das als Analogie zum Erinnern gesehen werden kann, und Inferenzen über mehrere Relationen hinweg sind die wesentlichen Operationen innerhalb der Antwortbasis. Anhand folgender Beispiele sollen diese Prozesse erläutert werden:

- (1a) (Welch- Moebel sind fest ?)
- (1b) (Jed-Moebel ist fest.)
- (2a) (Welch- Moebel sind heil ?)
- (2b) (Stuhl1 Tischi)

Die Fragen 1a und 2a stellen für die Analysephase keine Schwierigkeit dar. Bei der Beantwortung von Fragen spielen Erwartungen und sich darauf stützende Strategien eine wesentliche Rolle. So wird zunächst angenommen, daß es sich eventuell um eine triviale Frage handelt, bei der das als Selektor

für eine Untermenge interpretierte Prädikat FEST auf die Menge aller Möbel zutrifft. Eine im Analyseteil aufgerufene Prozedur aktiviert daher eine Funktion, die untersucht, ob im begrifflichen Netz eine D-Kante unmittelbar von MOEBEL zu FEST führt. Da dies nicht der Fall ist, obwohl die Hypothese, daß es sich um eine triviale Frage handelt, noch nicht falsifiziert ist, wird eine weitere Funktion aufgerufen, die nacheinander alle von Obermengen des Begriffs MOEBEL wegführenden D-Kanten daraufhin überprüft, ob sie zu FEST führen. Dieser Fall tritt bei 1a schon beim Knoten GEGENSTAND auf, bei dem FEST als Denotat vermerkt ist. Da es sich um eine allgemeine Eigenschaft von Gegenständen und damit von Möbeln handelt, kann die Frage mit der generalisierenden Antwort 1b beantwortet werden.

Bei 2a wird zunächst die gleiche Strategie angewendet; jedoch ohne Erfolg. In einer Art backtracking-Prozeß wird von einer Prozedur geprüft, ob es zu MOEBEL Untermengen gibt. Da im begrifflichen Netzwerk die Mengen STUHL und TISCH als Untermengen von MOEBEL enthalten sind, wird die Menge aller zu MOEBEL gehörigen Tokens ermittelt. Nacheinander wird im referentiellen Netzwerk geprüft, ob von einem Token REF-Kanten zu HEIL führen. Falls alle in Frage kommenden Tokens im aktuellen Zustand HEIL sind, wird die Frage nun nicht mit einer generalisierenden Antwort wie Jed- Möbel ist heil beantwortet, sondern explizit wie in 2b, da dem System bekannt ist, daß die im referentiellen Netzwerk gespeicherten Angaben nur momentane Attribute von Objekten sind.

In vielen Fällen wird Information darüber gespeichert, warum ein heuristisches Verfahren an einem gewissen Punkt gescheitert ist. Diese "read-time-inferences" (vgl. CHARNIAK 1975) erlauben es dem sonst mit "question-time-inferences" arbeitenden System, vom Benutzer evtl. gewünschte Begründungen der Antwort durch Zugriff auf diese Information unmittelbar zu generieren.

5.4. Als Ergebnis des auf einer geordneten Menge von Antwort-Tripeln (A-Tripel) arbeitenden Generierungsteils können in Abhängigkeit von dem textlinguistischen Modus (vgl. 3.2.) Ellipsen wie Alles, Nichts, Doch, Nein usw. oder transformierte Tripel oder ein Text auftreten, der z.B. auch Begründungen für die Antwort enthält. Bei der Erzeugung wohlgeformter Oberflächenstrukturen werden Verkettungsalgorithmen verwendet. So würden z.B. auf die Frage Wo steh- Du ? die folgenden A-Tripel erzeugt (MM1 RAUM1 IN), (MM1 TISCH1 VOR), (MM1 STUHL1 HINTER), (MM1 STUHL2 HINTER) und daraus die Antwort Ich steh- in d- Raum, vor d- Tisch und hinter d- Stühle generiert.

Mit der Ausgabe der Antwort wird die Verarbeitung einer Frage abgeschlossen.

## 6. Technische Hinweise

Das Programm ist in der Programmiersprache LISP 1.6 II (vgl. WITTIG 1975) geschrieben und auf der mit dem Prozessor KA 10 und dem Betriebssystem TOPS-10 ausgestatteten time-sharing-Anlage PDP-10 des Instituts für Informatik in Hamburg implementiert. Die ungefähr 2200 Programmzeilen sind zu 178 oft rekursiven Prozeduren zusammengefaßt. HAM-RPM belegt mindestens 41KWÖ bei einer Wortlänge von 36 Bits. Die Zeitangaben beziehen sich auf eine nicht-compilierte Version.

Der Quelltext des Programms kann Interessenten zur Verfügung gestellt werden.

### Anmerkungen

1. Da alle Autoren in gleicher Weise an der Entwicklung beteiligt sind, legen wir bei Zitierung Wert auf die Nennung aller vier Namen.
2. Vgl. zu diesem Abschnitt die unter Literatur angegebenen Titel.
3. Flexionsendungen werden bei der Verarbeitung nicht berücksichtigt. Derzeit beabsichtigen wir nicht, die von uns entwickelten Algorithmen HAM1 und HAM2 (vgl. v. HAHN/FISCHER 1975, v.HAHN/HOEPFNER 1975) in HAM-RPM einzusetzen.
4. Die Zeiten schwanken von 0,5 sec. für einfache Entscheidungsfragen bis zu 30 sec. bei Fragen, die größeren Aufwand erfordern.
5. Für weitere Ko-Text-Probleme vgl. WITTIG 1975.
6. Zum Verständnis dieser für LISP typischen Struktur vgl. QUAM 1969.

7. Vgl. die diesbezüglichen Untersuchungen von SCHEFE 1975.
8. Zu einer formalen Betrachtung und Verallgemeinerung semantischer Netzwerke vgl. BOLEY 1975.
9. Die Tatsache, daß es sich hier um "fuzzy sets" handelt, wurde z.Zt. noch nicht genügend berücksichtigt; vgl. LE-FAIVRE 1974. So dürfte z.B. STUHL1 (REF:HEIL) nicht in gleicher Weise wie STUHL2 (REF:KAPUTT) auf STUHL bezogen werden.
10. Zur Konvergenz beider Richtungen vgl. NILSSON 1974.
11. Da in LISP 1.6 keine Möglichkeiten zur Deklaration und Kontrolle konkurrierender Prozesse z.B. in Form von Coroutinen vorhanden sind, und Mechanismen zur "pattern-directed invocation" ebenfalls fehlen (vgl. LEAVENWORTH/SAEMET 1974), konnte dieses Konzept nicht in geeigneter Weise implementiert werden, sondern mußte durch komplexe Kontrollstrukturen und Pattern Matching ersetzt werden.

## **Literatur**

- BOBROW, Daniel G. / COLLINS, Allen M. (eds.) (1975): Representation and understanding: Studies in cognitive science. New York: Academic.
- BOLEY, Harold (1974): Einfache natürlichsprachliche Dialoge mit einem semantischen Netzwerkprogramm. Mitteilung Nr. 10, Institut für Informatik, Hamburg.
- BOLEY, Harold (1975): Representing knowledge with directed recursive laibelnode hypergraphs. Institut für Informatik, Hamburg (erscheint Ende 1975).
- CHARNIAK, Eugene (1975): "Inference and knowledge". Lecture Notes of the Tutorial on Computational Semantics. Castagnola (vervielf.).
- GÖSCHEL, Joachim / VEITH, Werner H. (eds.) (1975): Neuere Forschungen in Linguistik und Philologie. Zeitschrift für Dialektologie und Linguistik. Beiheft Nr. 13. Wiesbaden: Steiner.
- v.HAHN, Walther / FISCHER, Hartmut (1975): "Über die Leistung von Morphologisierungsalgorithmen bei Substantiven". GÖSCHEL/VEITH (eds.): 130-150.
- v.HAHN, Walther / HOEPPNER, Wolfgang (1975): "HAM2 -Ein Algorithmus zur Lemmatisierung deutscher Verben". GÖSCHEL/ VEITH (eds.): 151-171.
- HUNT, Earl B. (1975): Artificial intelligence. New York: Academic Press.
- LEAVENWORTH, B. / SAMMET, Jean (1974): "An overview of nonprocedural languages". Sigplan Notices 9,4.
- LEFAIVRE, Richard A. (1974): Fuzzy problem-solving. Technical Report No. 37, Univ. of Wisconsin, Madison.
- MINSKY, Marvin L. (1968): Semantic information processing. Cambridge, Mass.: M.I.T. Press.
- MYLOPOULOS, John / BORGIDA, Alex / COHEN, Philip / ROUSSOPOULOS, Nicholas / TSOTSOS, John / WONG, Harry (1975): "TORUS - a natural language understanding System for data management". Advance Papers of the fourth Joint Conference on Artificial Intelligence. Tiflis: 414-421.
- NEWELL, Allen / SIMON, Herbert A. (1972): Human problem solving. Englewood Cliffs: Prentice-Hall.
- NILSSON, Nils J. (1974): "Artificial intelligence". Information Processing 74. Amsterdam: North Holland: 778-801 (preprint).
- NORMAN, Donald A. / RUMELHART, David E. (1975): Explorations in cognition. San Francisco: Freeman.
- QUAM, Lynn (1969): LISP 1.6. Stanford-AI-Lab. Operating Note 28.3, Decus No. 6/10-38A.

- RUSTIN, Randall (ed.) (1973): Natural language processing. Englewood Cliffs: Prentice-Hall.
- SCHANK, Roger C. (ed.) (1975): Conceptual information processing. Amsterdam: North Holland.
- SCHANK, Roger C / COLBY, Kenneth M. (eds.) (1973): Computer models of thought and language. San Francisco: Freeman.
- SCHEFE, Peter (1975): Beschreibung eines Programms für das Konstruieren und Testen von formalen Grammatiken im Dialog mit einer Rechenanlage. Mitteilung Nr. 11, Institut für Informatik, Hamburg.
- WAHLSTER, Wolfgang (1975): "Pattern Matching und die Transformation von S-Expressions". WITTIG (ed.).
- WALKER, Donald E. (1973): Automated language processing. Technical Note 77, Stanford Research Inst.
- WILKS, Yorik (1975): "Parsing English". Lecture Notes of the Tutorial on Computational Semantics. Castagnola (vervielf.).
- WINOGRAD, Terry (1972): Understanding natural language. Edinburgh: Univ. Press. (1974): Five lectures on artificial intelligence. Stanford-Cs-74-459.