In: Bolc, L. (ed.): Natural Language Based Computer Systems
München: Hanser / Macmillan, p. 119-253

The Anatomy of the Natural Language Dialogue System HAM-RPM

Walther v. Hahn, Wolfgang Hoeppner, Anthony Jameson, Wolfgang Wahlster

ABSTRACT: HAM-RPM is a dialogue system which converses with a human partner in colloquial German about limited, but interchangeable scenes. The objective of this report is to give a detailed, complete and self-contained description of the system in its present state of implementation. After a discussion of the goals and methodological principles which guide our research and a short introduction to the implementation language an overview of the system's architecture, of its knowledge base and of the domains of discourse is given. Then each processing phase from the analysis of natural-language input to the generation of a natural-language utterance is described in detail. The examples used during these descriptions are supplemented by transcripts of complete dialogue sessions. Finally HAM-RPM's programming environment is described.

1 THEORETICAL AND TECHNICAL BACKGROUND OF HAM-RPM

1.1 UNDERLYING ISSUES

HAM-RPM is a dialogue system which converses with a human partner in colloquial German about limited, but interchangeable scenes.

Recently, much AI research even in non-linguistic areas has been seeking contact with natural-language simulation systems. This interest has been motivated by many arguments, including the following:

- Natural language is a highly efficient means of interacting with non-linguistic AI systems.
- The other cognitive abilities, which are objects of interest of AI systems in general, are often directly or indirectly connected with linguistic abilities.
- Natural language systems are better suited to the expectations of naive users and can be managed by them better.

Most natural-language artificial-intelligence systems have been developed as interfaces to some sort of task-oriented system. Typical of this function are question-answering systems involved with information storage and retrieval or other documentation tasks (WALTZ 1977).

In the research group 'Simulation of Language Understanding' at the University of Hamburg, we are investigating the feasibility of computer simulation of natural-language structures and their communicative use in a systematic way. We are thus studying natural language as an object of interest in itself.

The overall goal of HAM-RPM can be characterized as the simulation of a natural-language dialogue partner on a computer. The three main components of this definition will now be described in more detail.

1.1.1 Simulation

The kind of simulation aimed at is simulation of function rather than that of performance; the degree to which this is achieved varies greatly from one component of the program to another.

The first direction of research on HAM-RPM involved an attempt to extend the breadth of the system so as to obtain a representative variety of linguistic features of performance. With respect to the function of language, this meant that we built up a sequence of communicative mechanisms extending as completely as possible from the analysis of an input sentence (more generally: a text) to the utterance of a response. We are convinced that a system is, in respect of methodology, the more interesting the more it generates or analyzes a really large number of performance features. First, such systems are realistic enough (reduction in the number of unsimulated features of the original, cf. Report No. 4). Much more importantly, these systems permit the investigation of the interactions among a large number of dynamic components. Incidentally, the performance of such systems is generally very difficult for the naive user to predict.

While we have been working in this first direction of research – the one which we have emphasized since the beginning of the project – some components have been represented only by provisional processes to mark the interfaces and to return acceptable values in standard situations.

It seems to us to be very important for sophisticated simulations that right from the beginning each new or modified process be testable in an existing and running program representing the communicative framework, so that each extension of the system takes place under comparable conditions and endless problems involving 'local correctness' of program components are avoided. With the aim of obtaining a realistic simulation of a dialogue partner, we have attempted to develop the different linguistic, communicative and cognitive abilities of the system as evenly as possible. Accordingly, we have developed individual parts of the system only to the extent which seemed sensible in view of the state of development of the system as a whole. We hoped thus to avoid both the dominance of one component or another (an often-criticized feature of earlier systems) and disadvantageous restrictions placed by one weak component on other components of the system.

1.1.2 Natural language

Our intention is that the linguistic behaviour of the system should exhibit the typical characteristics of natural-language dialogue behaviour. The features of special interest are:

- vague statements and inferences
- 'incomplete' syntactic structures
- implicit and redundant semantic structures.

While question-answering systems even today often proceed from the assumption that the best effects can be obtained using a restricted technical language like a subset of natural language, we try to adopt stylistic and pragmatic standards which are influenced as little as possible by the structures of technical language.

Furthermore, we intend that the internal structures used during the processing of a sentence remain as long as possible closely related to the structures of colloquial language, as we believe that colloquial language best exhibits the characteristics of natural language as a means of communication.

The efficiency of communication in colloquial language has undoubtedly been underrated by most application-oriented AI research. The work of HENDRIX 1977 and WALTZ 1978, among others, has shown, however, that the usefulness of such systems would be increased if they adopted some of the basic communicative and linguistic structures of this kind of language.

From a linguistic point of view, the combination - aimed at by HAM-RPM - of realistic natural-language utterances, an acting system, and a scene to which it can refer, is most interesting: Modern linguistics has been trying to establish a revised explication of language based on the notion of speech-action complexes and reference semantics. Traditional ways of proposing hypotheses or theories are in connection with this type of analysis less powerful with respect to:

- the assessment of the practicability of the underlying linguistic model
- the range of explanation
- the problems raised by interactions among the components of a complex system.

1.1.3 Communication

By speaking of a 'dialogue' partner we wish to emphasize that the spectrum of verbal actions performed by the system should be much wider than in question-answering systems. The sequences of verbal behaviour possible within dialogues with HAM-RPM are shown in Fig. 19. A prerequisite for such a breadth of simulation is the possibility of a uniform interpretation of all the system utterances as a simulation of natural behaviour. It is a part of our strategy in building the system, that no system messages, presentation of the system data, or the like are admitted when the system is in dialogue mode.

It is a further fundamental feature of more sophisticated artificial-intelligence systems to allow metacommunicative dialogues. This permits both partners to deal with problems involving the process of communication itself, giving them the opportunity to clear up missunderstandings, supply missing information, and so on. Here are two (independent) examples:

(1)	AUF WAS BEZIEHT SICH 'ES'? What does 'it' refer to?	(system)
(2)	WARUM WILLST DU DAS WISSEN? Why do you want to know that?	(partner)

1.1.4 Goals

By explaining how the system works and discussing the philosophy underlying the algorithms and their implementation, we hope to make a contribution to the following topics of research:

- the value of AI systems as a powerful scientific method of formulating hypothesis or theories, in particular as a way of operationalizing linguistic speech-act theories
- the extension of the scope of the simulation of natural language
- considerations of the communicative relevance of colloquial language in AI systems
- natural-language communication with computers.

Furthermore, HAM-RPM can be used as a testbed for the exploration of detailed theories about certain aspects of intelligent language behaviour.

Unlike many other workers in the field of AI, we try to take into account the results obtained by language researchers using other paradigms.

1.1.5 The aims of this report

Although the system HAM-RPM is described in this document in considerable detail, this document is not intended for use as a manual for the system; this function is rather fulfilled by HAM-RPM Memo No. 6.

On the other hand, relatively little is said here about the results of this research or the theoretical considerations underlying its design. Discussions of these issues can be found in the HAM-RPM Reports.

Although a great deal of work on the system is still planned, it is here described exactly as it is presently implemented, with the aim of presenting a complete picture of a large system, including both its strong and weak points. The examples of the system's behaviour presented are all excerpts from actual dialogues conducted with it. The English translations are as literal as possible within the bounds of acceptable English style.

1.2 THE PROGRAMMING LANGUAGE FUZZY

1.2.1 General characterization

HAM-RPM is implemented in FUZZY (LEFAIVRE 1977), a very-high-level artificial-intelligence language containing the most important mechanisms found in other languages in the tradition of PLANNER. Its distinguishing features are a set of mechanisms for dealing with fuzzy knowledge.

The description of FUZZY which follows is only detailed enough to permit an understanding of the rest of this document. All of the mechanisms mentioned are in fact more general and more powerful than the simplified descriptions given here suggest. Some features which appear in only one or two examples are not explained here, as it is assumed that their use will be understandable in context. FUZZY is embedded in Rutgers-UCI-LISP (LEFAIVRE 1978). Brief introductions to LISP can be found in SIKLOSSY 1976 and WINSTON 1977.

1.2.2 Pattern matching

FUZZY's pattern matcher uses 'FUZZY variables' which are independent of LISP variables. The prefix '?' indicates that a variable is to receive a value, whereas the prefix '!' causes the variable to be evaluated. Thus after the successful match

(1) (MATCH (A FORM OF ?STEM) (A FORM OF CHILD))

the FUZZY variable '?STEM' is bound to 'CHILD'. The expression
(2) (L STEM CHILDREN !STEM)
is then said to 'instantiate' to
(3) (L STEM CHILDREN (CHILD)) .

If the given pattern doesn't match the given structure, the match is said to 'fail', i.e. no FUZZY variables are bound and the special atom FAIL is returned. Otherwise the match is said to 'succeed'. The words 'fail' and 'succeed' are also used in this way in connection with data-base search (see 1.2.3) and the invocation of procedural knowledge (see 1.2.4).

A '?' standing alone matches anything. Segment variables are indicated by the double prefixes '??' and '!!'.

1.2.3 The associative net

FUZZY provides an assertional data base whose entries have the form

(4) ((D APPLE/ JUICE SWEET) . 0.7).

The left-hand side is an arbitrary list structure. In (4), it is interpreted by HAM-RPM as ascribing sweetness to apple juice.

The number on the right is the 'z-value' which characterizes the assertions in FUZZY's data base. The z-value is often interpreted as a fuzzy truth value, but it is used in HAM-RPM in several other ways as well. In (4) the interpretation is in terms of frequency: apple juice is usually sweet. Note that the interpretation of the z-value within a particular context is specified only implicitly, by the procedures which make use of the assertion in question.

The FUZZY primitive for adding assertions to the associative net is ADD:

(5) (ADD (D APPLE/ JUICE SWEET) 0.7)

Assertions are retrieved using the primitive FETCH, whose first argument is a pattern. The command

(6) (FETCH (D ?OBJECT SWEET) 0.5)

will return an assertion which matches the pattern and which has a z-value equal to or greater than 0.5, such as the assertion in (4). Since the assertions are ordered according to z-value in the net, the appropriate assertion with the highest z-value is returned. If no such assertion is present in the net, the FETCH fails.

The associative net can be divided into any number of separate parts, called CONTEXTs, each of which has a name. The procedure CONTEXT is used to switch the FUZZY interpreter's attention from one CONTEXT to another. Thus the command

(7) (CONTEXT 'LEXIKON)

means that all following ADDS and FETCHes will refer only to the part of the net called LEXIKON.

The interpretation of the various CONTEXTs is up to the programmer. In HAM-RPM, the CONTEXT mechanism is often used simply as a way of avoiding having data which have nothing to do with each other shuffled together in a single part of the data base.

1.2.4 DEDUCE-procedures

Knowledge can be represented procedurally in FUZZY using DEDUCE-procedures, as exemplified by the following definition:

```
(8)     (PROC NAME: DRINK-TASTY
              (D (*R ?OBJECT (EQUAL (*TYPE !OBJECT 'DRINK))) TASTY)
                    (FETCH (D !OBJECT SWEET))
                    (FETCH (D !OBJECT NON-ALCOHOLIC))
              (SUCCEED))
```

This procedure could be invoked directly, for instance with the command

(9) (DRINK-TASTY (D APPLE/ JUICE TASTY))

to check whether apple juice is to be considered tasty. If the interpreter could find assertions in the net to the effect that apple juice was sweet and non-alcoholic, the procedure would succeed. Usually however, DEDUCE-procedures are invoked not by name, but via their 'calling pattern'. In (8), this is the expression in the second line, which specifies that this procedure can be used to show that a particular object which is a drink is tasty.
The command

(10) (GOAL (D APPLE/ JUICE TASTY))

causes the interpreter first to perform a FETCH on the given pattern to see if such an assertion is stored in the associative net. If the FETCH fails, all DEDUCE-procedures whose calling patterns match the pattern passed as an argument to GOAL are invoked in turn, until one of them succeeds.

The primitive GOAL can thus be used to specify a condition without knowledge of how the fulfillment of the condition might be proved. For example, the definition (8) could be made more powerful by replacing the two FETCHes with GOALs. (The version of this procedure actually used in HAM-RPM is shown in section 2.12.3).

1.2.5 Procedure demons

Each DEDUCE-procedure has a procedure demon assigned to it by the programmer, which supervises the execution of the procedure: after the evaluation of each expression in the procedure, the value of the expression is passed to the responsible demon, which can then take appropriate action, such as aborting the evaluation of the procedure if the prospects of success appear too poor.

A typical activity of procedure demons is the updating of a special variable ZACCUM, which is usually a number. For example, FUZZY's default demon continually resets ZACCUM to the minimum of all the z-values which have been returned during the evaluation of the procedure. Thus if, during the evaluation of the procedure in (8) the first FETCH succeeded with a z-value of 0.7 and the second with 0.5, the value of ZACCUM at the end of the execution would be 0.5. This value could be used to estimate the reliability of the conclusion that the drink in question was tasty.

1.2.6 The language AIMDS

A powerful extension of FUZZY, called AIMDS, has been developed at Rutgers University (SRIDHARAN 1978). An important feature of this language is its implementation of the 'template' data structure. Templates are comparable to the data structures often referred to as 'frames' or 'schemata' (BOBROW/WINOGRAD 1977).

None of the mechanisms of AIMDS are used in the currently implemented version of HAM-RPM, but experiments are being performed in the following areas:

- the representation of semantic deep structure (see 2.11.1)
- the declarative representation of prototypical dialogue sequences
- the representation of complex actions.

2 PROCESSING STEPS IN HAM-RPM

2.1 OVERVIEW

2.1.1 Domains of discourse

The domains of discourse which will in most cases underly examples in this report are a living room (WORLD3) and a traffic scene (WORLD2).

To give an impression of the kind of objects which are contained in these domains two graphic representations illustrate the scenes. These drawings are bird's-eye views, in which the locations of objects are represented by circles. Not all of the properties of the objects are represented in these figures.

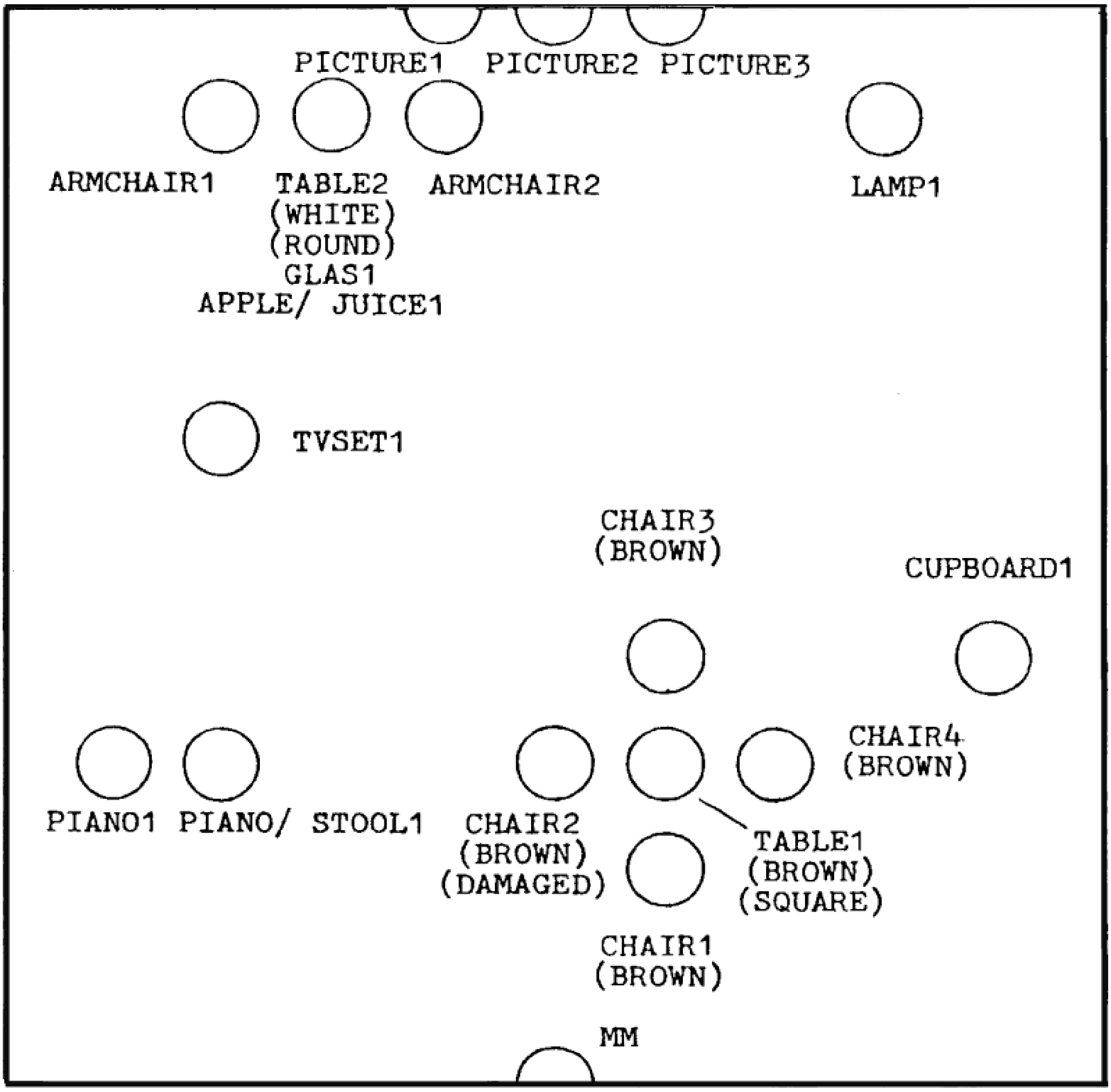


Fig. 1

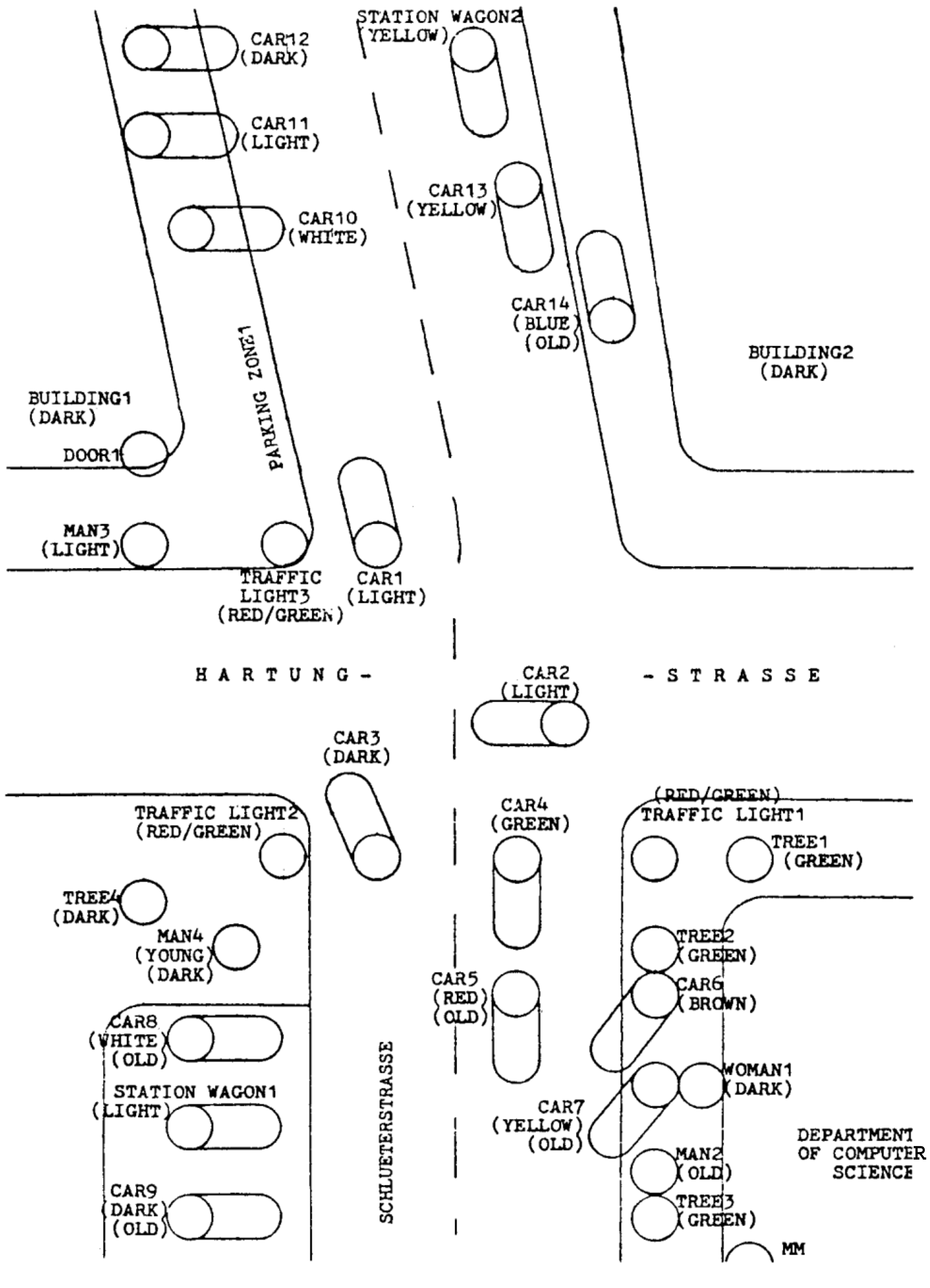


Fig. 2

2.1.2 Processing phases

This section introduces the interaction between the components of HAM-RPM and the connections of processing modules and the knowledge base.

Fig. 3 represents the components in a linear order their functions being indicated by a few key words. The first two pages may be regarded as the environment of the system. On the following four pages components for the processing of single sentences in the context of a dialogue are illustrated. The names of components on the left-hand side of each page represent the call to components which deviate from the linear ordering. The numbers on the right refer to the sections of this report, in which the components are described in detail.

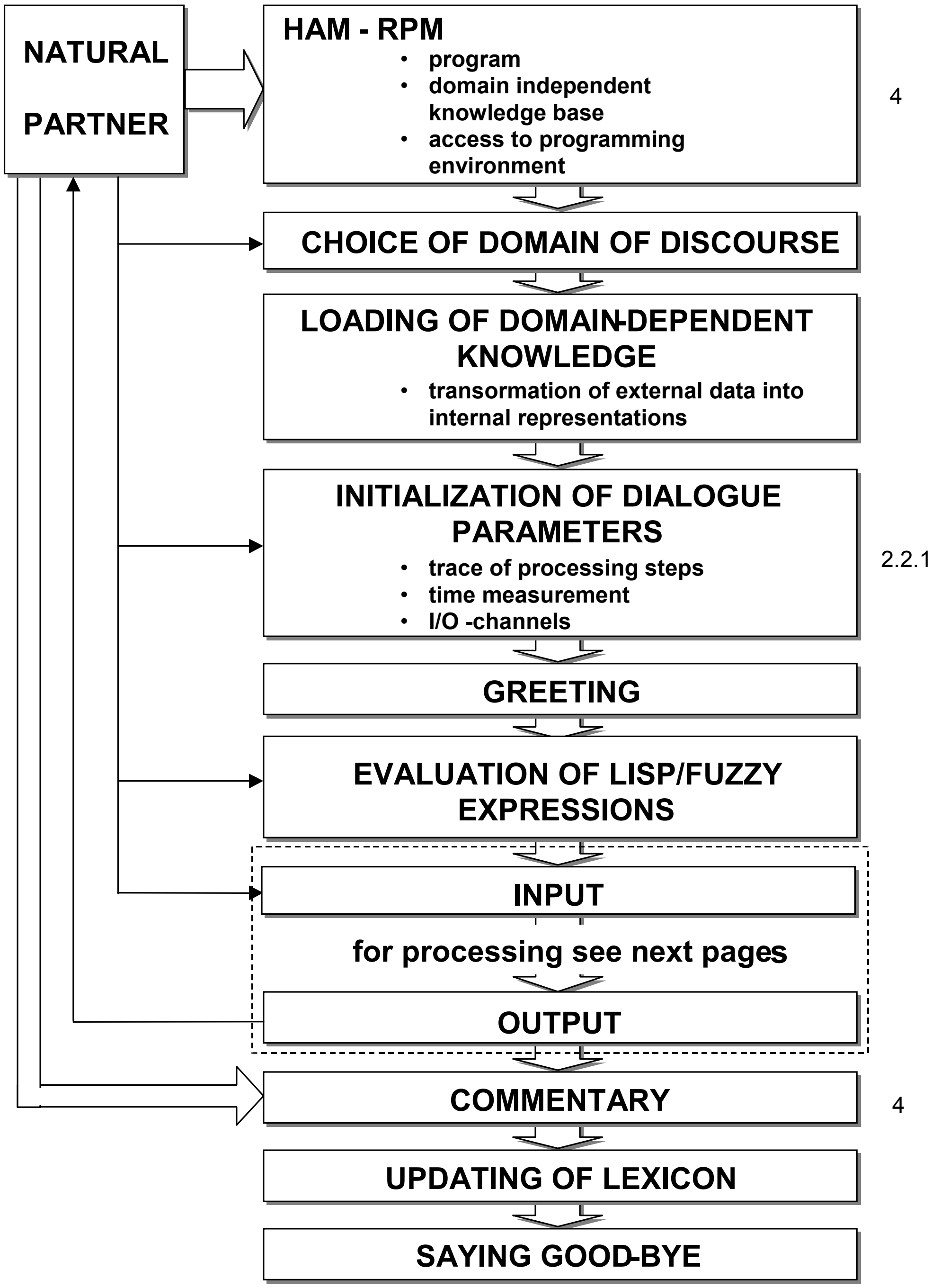


Fig. 3a + b

Fig. 4 lists the parts of the knowledge base whose external representation is introduced in section 2.1.3. Arrows pointing to names of processing components indicate the accessing of the corresponding part of the knowledge base by the component. Arrows the other way around represent the construction or modification of parts of the knowledge base by processing components.

In this report we do not distinguish the names of files containing domain-dependent knowledge with numbers, such as REF2.FUZ which is the actual name of the file on which the referential semantic network for WORLD2 is stored.

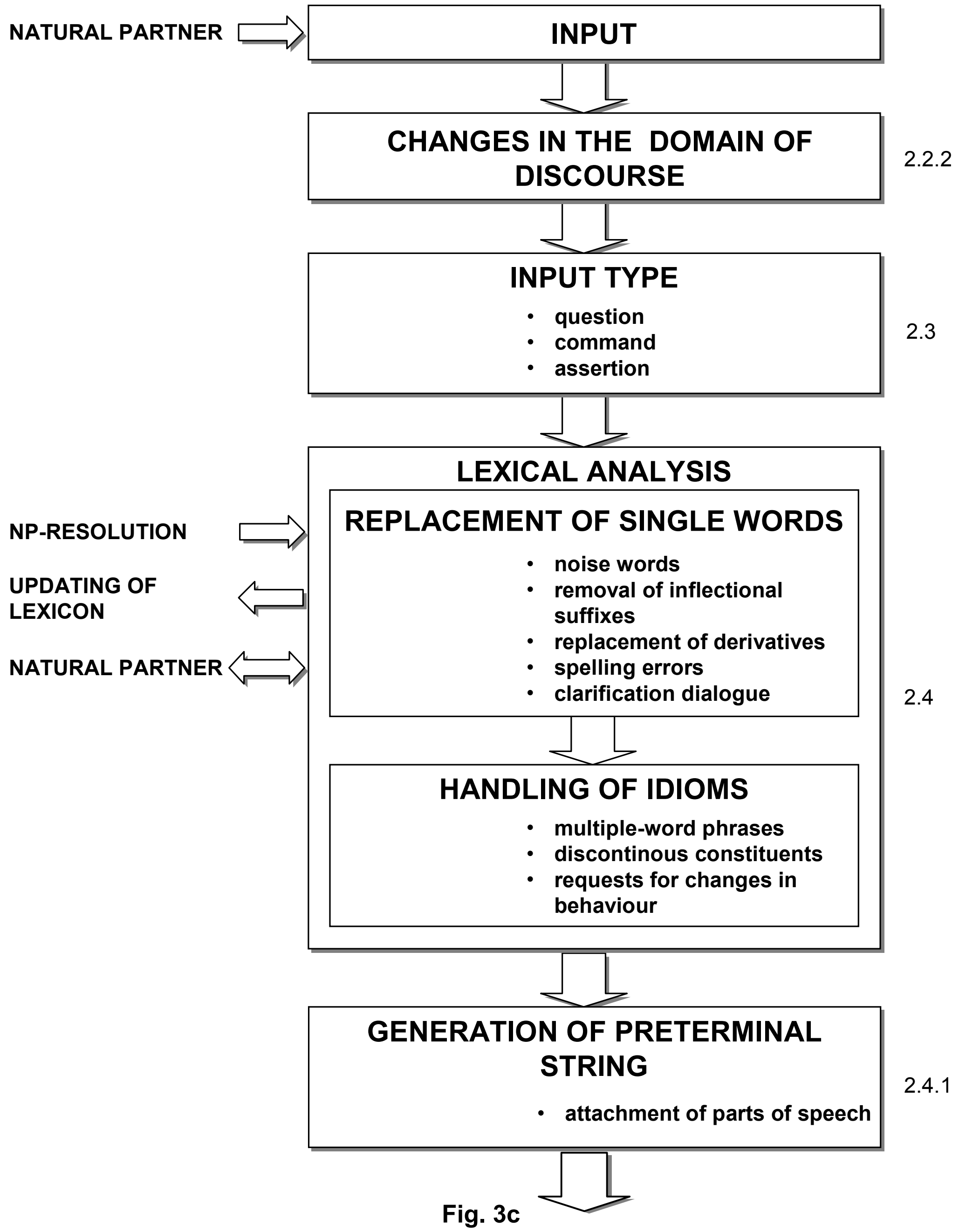


Fig. 3c

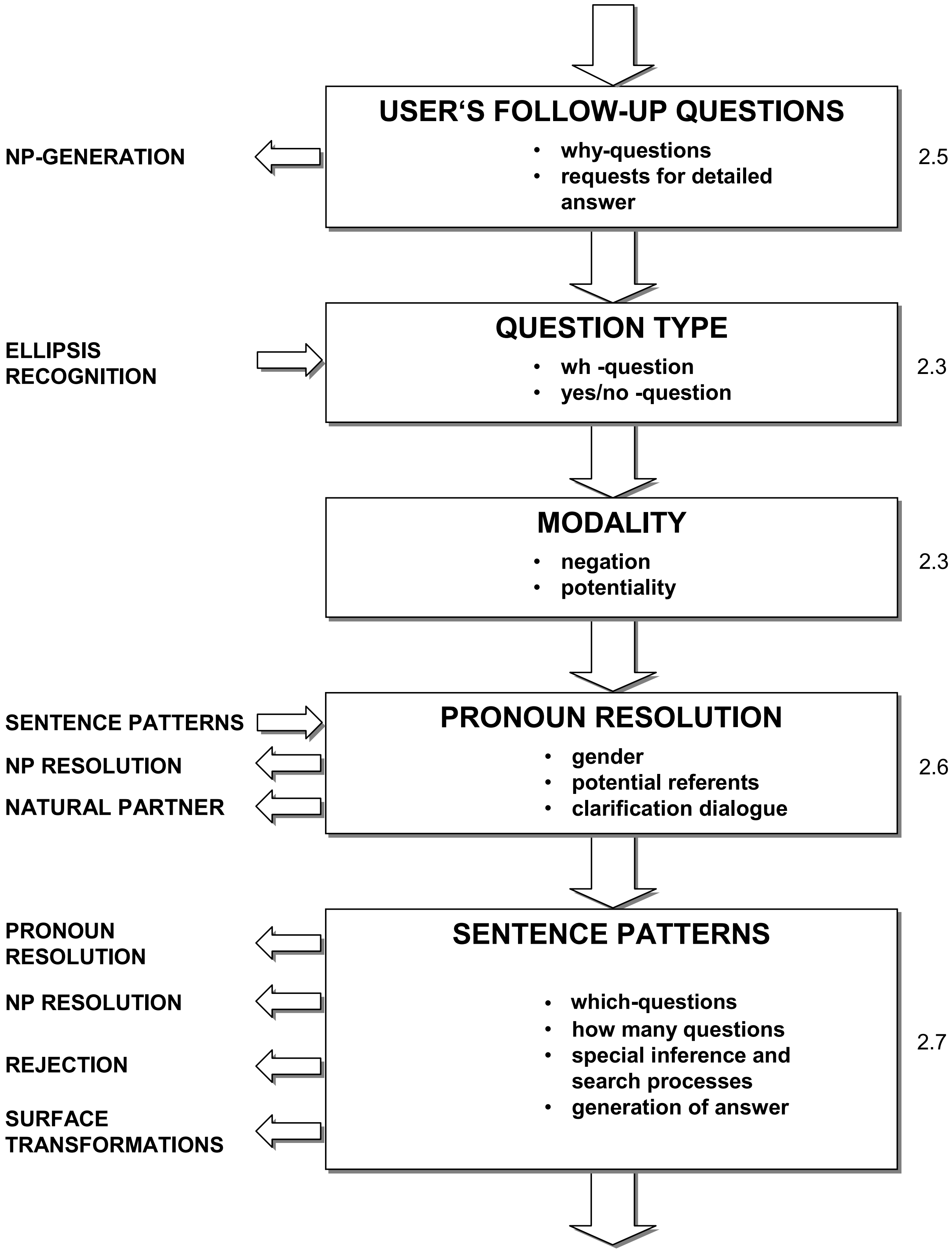

USER'S FOLLOW-UP QUESTIONS
• why-questions
• requests for detailed answer
NP-GENERATION
2.5
QUESTION TYPE
• wh -question
• yes/no -question
ELLIPSIS RECOGNITION
2.3
MODALITY
• negation
• potentiality
2.3
PRONOUN RESOLUTION
• gender
• potential referents
• clarification dialogue
SENTENCE PATTERNS
NP RESOLUTION
NATURAL PARTNER
2.6
SENTENCE PATTERNS
• which-questions
• how many questions
• special inference and search processes
• generation of answer
PRONOUN RESOLUTION
NP RESOLUTION
REJECTION
SURFACE TRANSFORMATIONS
2.7

Fig. 3d

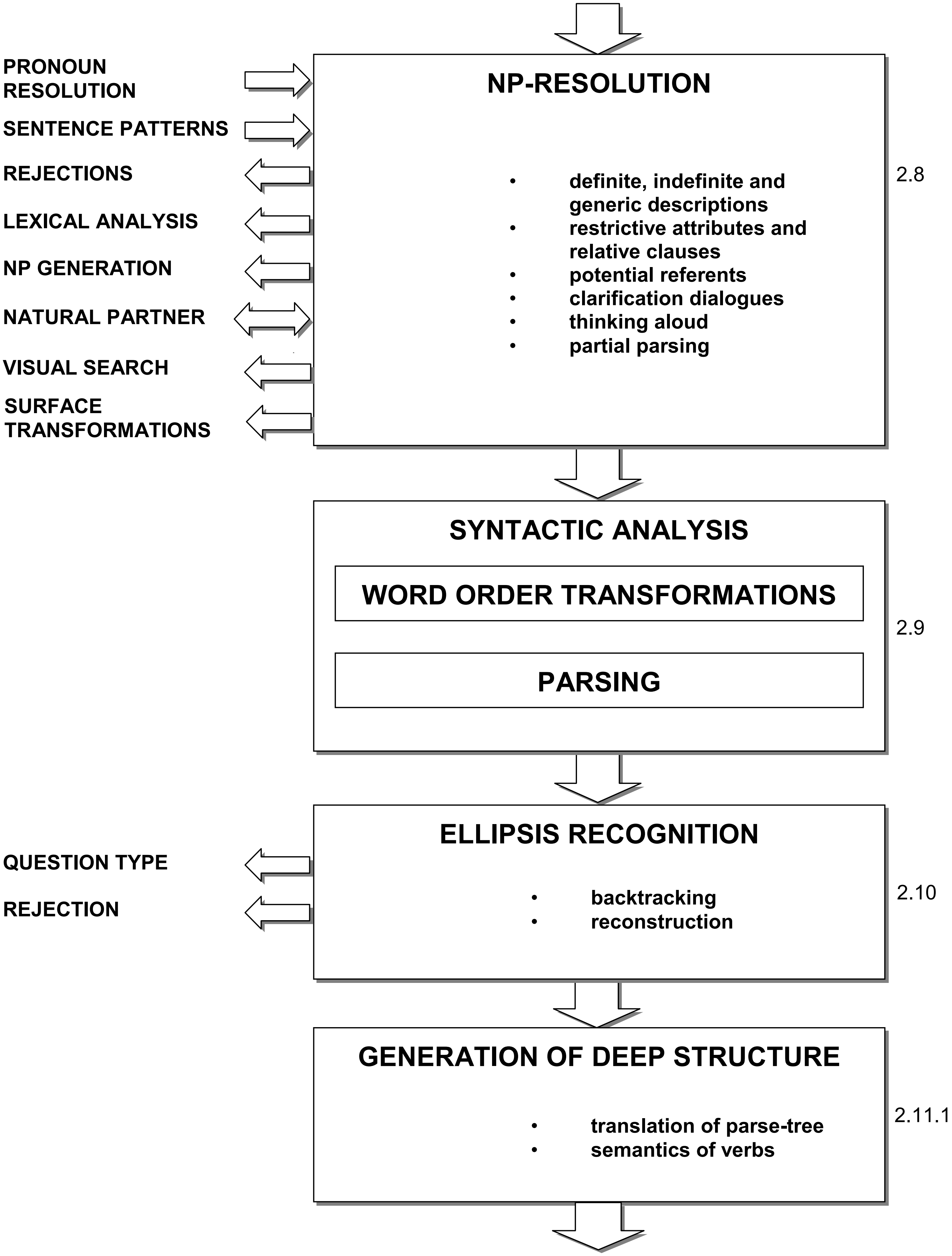

PRONOUN RESOLUTION
SENTENCE PATTERNS
REJECTIONS
LEXICAL ANALYSIS
NP GENERATION
NATURAL PARTNER
VISUAL SEARCH
SURFACE TRANSFORMATIONS
NP-RESOLUTION
• definite, indefinite and generic descriptions
• restrictive attributes and relative clauses
• potential referents
• clarification dialogues
• thinking aloud
• partial parsing
2.8
SYNTACTIC ANALYSIS
WORD ORDER TRANSFORMATIONS
PARSING
2.9
QUESTION TYPE
REJECTION
ELLIPSIS RECOGNITION
• backtracking
• reconstruction
2.10
GENERATION OF DEEP STRUCTURE
• translation of parse-tree
• semantics of verbs
2.11.1

Fig. 3e

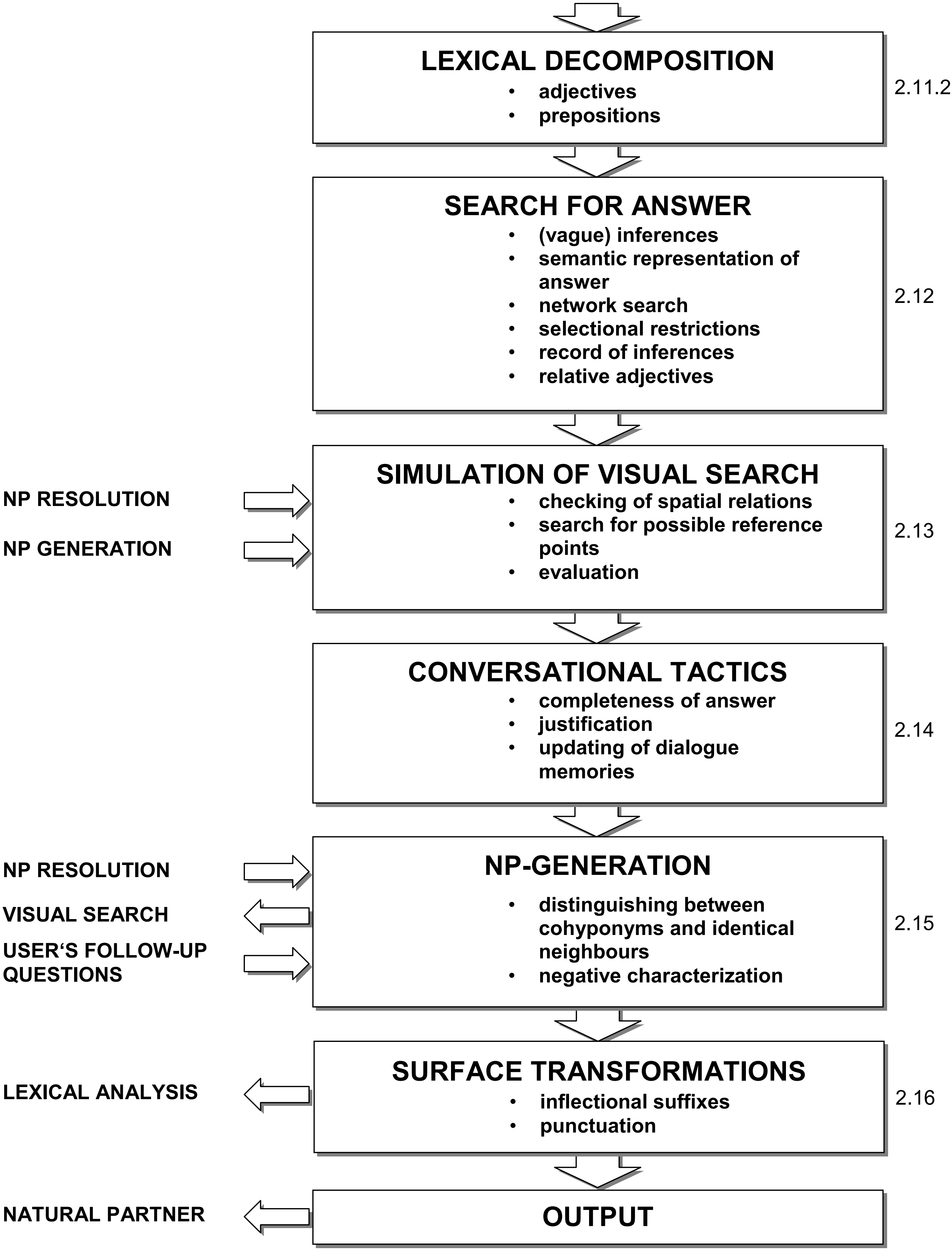

LEXICAL DECOMPOSITION
adjectives
prepositions
2.11.2
SEARCH FOR ANSWER
(vague) inferences
semantic representation of answer
network search
selectional restrictions
record of inferences
relative adjectives
2.12
NP RESOLUTION
NP GENERATION
SIMULATION OF VISUAL SEARCH
checking of spatial relations
search for possible reference points
evaluation
2.13
CONVERSATIONAL TACTICS
completeness of answer
justification
updating of dialogue memories
2.14
NP RESOLUTION
VISUAL SEARCH
USER‘S FOLLOW-UP QUESTIONS
NP-GENERATION
distinguishing between cohyponyms and identical neighbours
negative characterization
2.15
LEXICAL ANALYSIS
SURFACE TRANSFORMATIONS
inflectional suffixes
punctuation
2.16
NATURAL PARTNER
OUTPUT

Fig. 3f

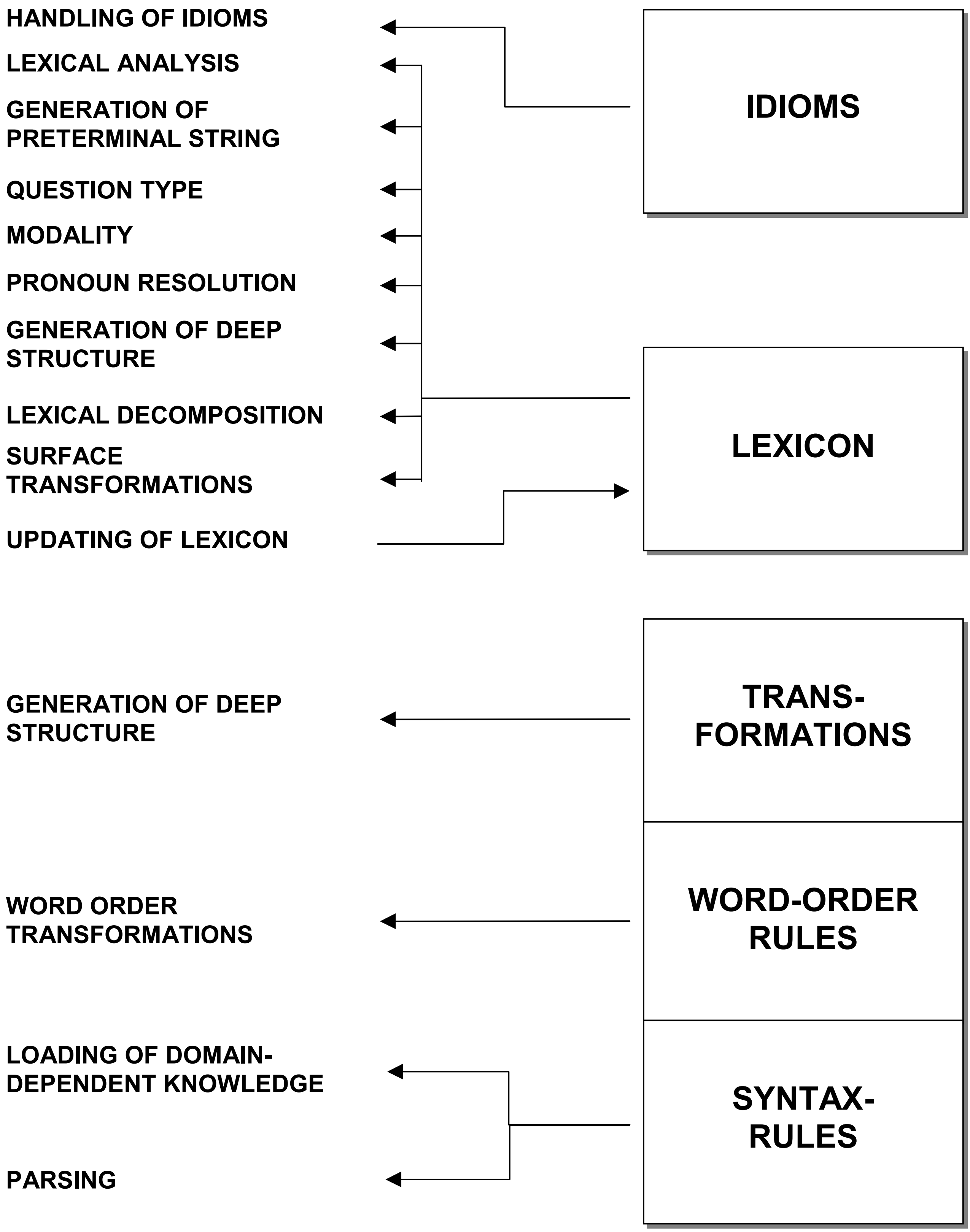

HANDLING OF IDIOMS
LEXICAL ANALYSIS
GENERATION OF PRETERMINAL STRING
QUESTION TYPE
MODALITY
PRONOUN RESOLUTION
GENERATION OF DEEP STRUCTURE
LEXICAL DECOMPOSITION
SURFACE TRANSFORMATIONS
UPDATING OF LEXICON
IDIOMS
LEXICON
GENERATION OF DEEP STRUCTURE
WORD ORDER TRANSFORMATIONS
LOADING OF DOMAIN-DEPENDENT KNOWLEDGE
PARSING
TRANS-FORMATIONS
WORD-ORDER RULES
SYNTAX-RULES

Fig. 4a

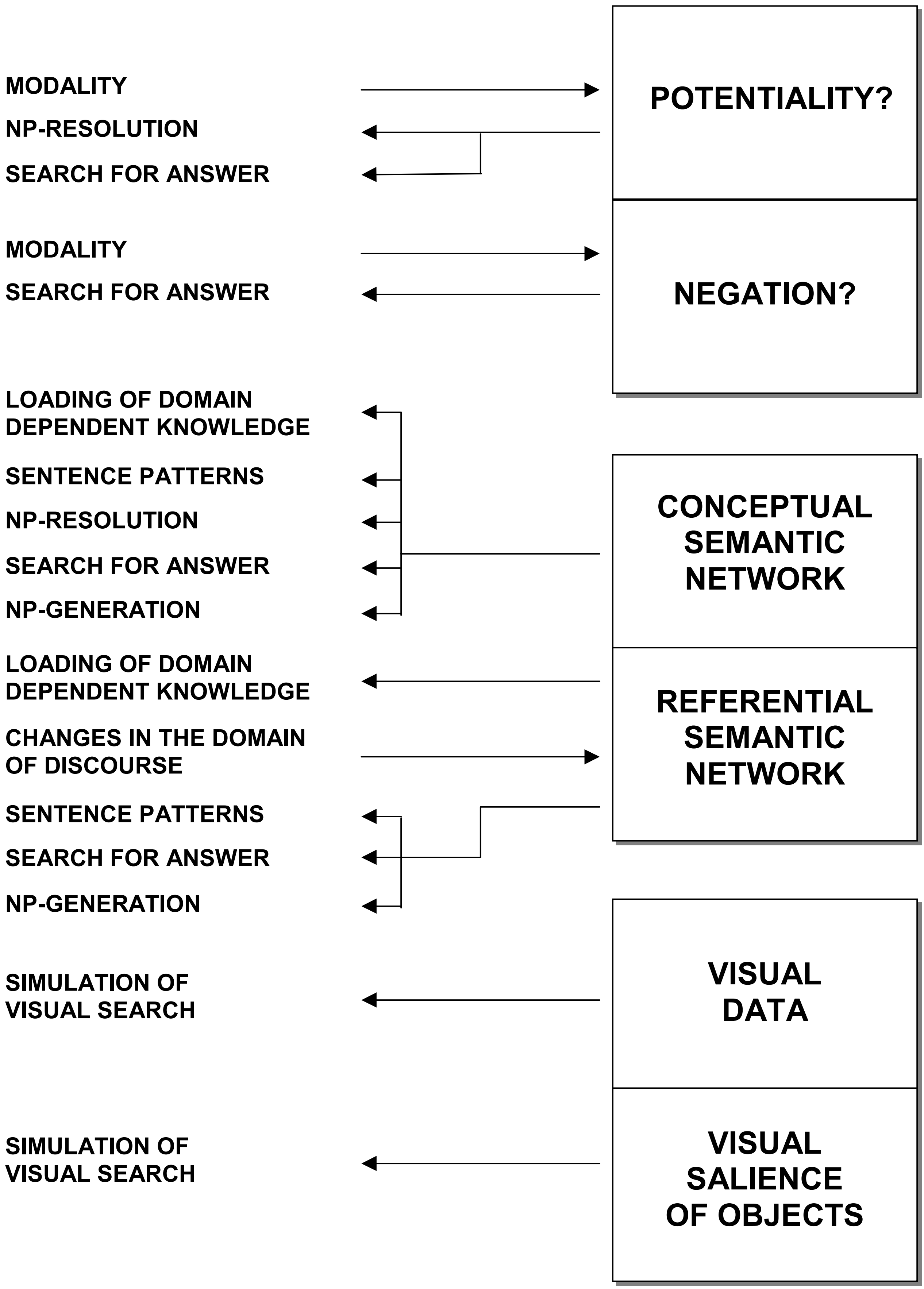

MODALITY
NP-RESOLUTION
SEARCH FOR ANSWER
POTENTIALITY?
MODALITY
SEARCH FOR ANSWER
NEGATION?
LOADING OF DOMAIN DEPENDENT KNOWLEDGE
SENTENCE PATTERNS
NP-RESOLUTION
SEARCH FOR ANSWER
NP-GENERATION
CONCEPTUAL SEMANTIC NETWORK
LOADING OF DOMAIN DEPENDENT KNOWLEDGE
CHANGES IN THE DOMAIN OF DISCOURSE
SENTENCE PATTERNS
SEARCH FOR ANSWER
NP-GENERATION
REFERENTIAL SEMANTIC NETWORK
SIMULATION OF VISUAL SEARCH
VISUAL DATA
SIMULATION OF VISUAL SEARCH
VISUAL SALIENCE OF OBJECTS

Fig. 4b

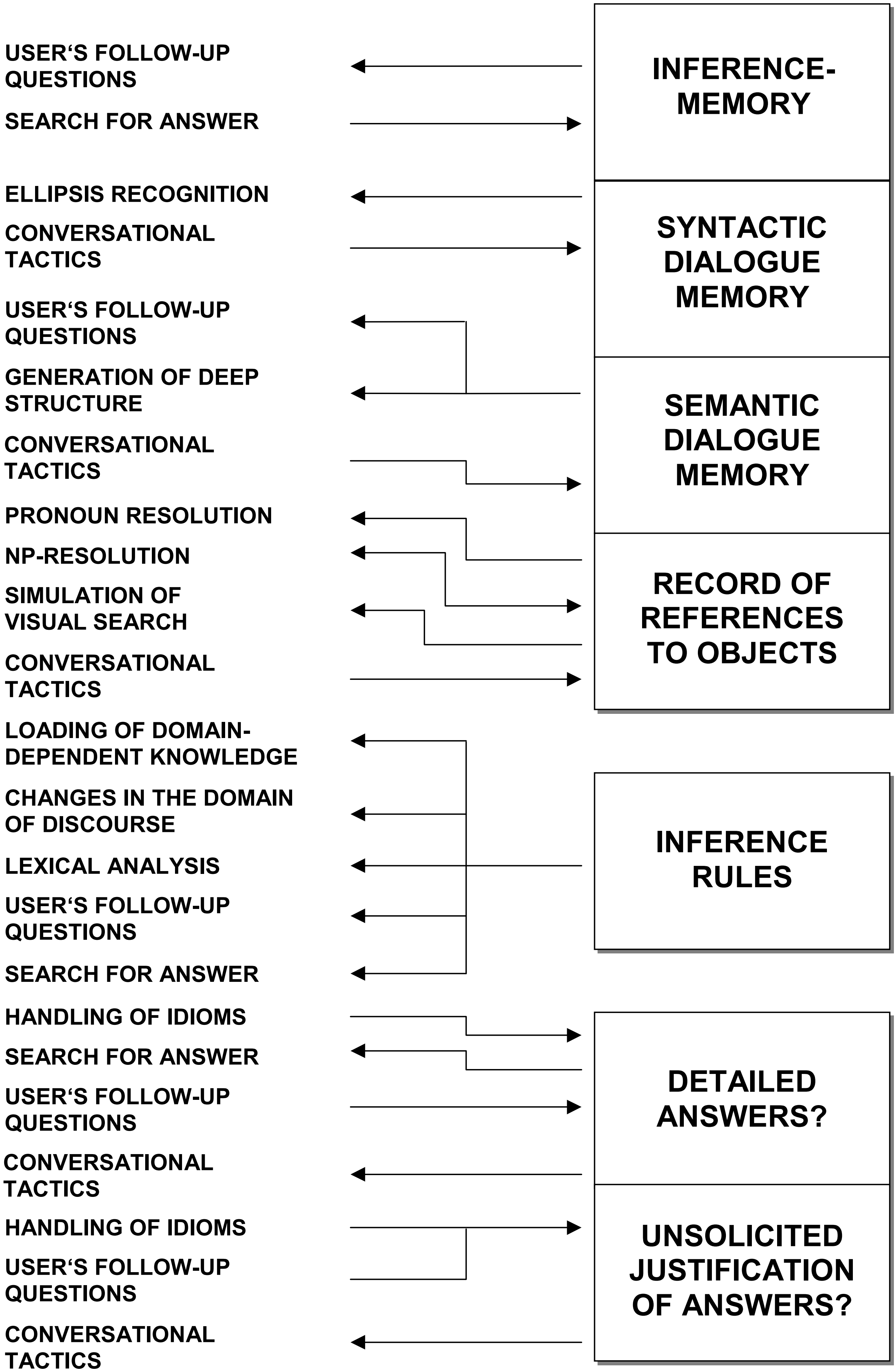

USER'S FOLLOW-UP QUESTIONS
SEARCH FOR ANSWER
ELLIPSIS RECOGNITION
CONVERSATIONAL TACTICS
USER'S FOLLOW-UP QUESTIONS
GENERATION OF DEEP STRUCTURE
CONVERSATIONAL TACTICS
PRONOUN RESOLUTION
NP-RESOLUTION
SIMULATION OF VISUAL SEARCH
CONVERSATIONAL TACTICS
LOADING OF DOMAIN-DEPENDENT KNOWLEDGE
CHANGES IN THE DOMAIN OF DISCOURSE
LEXICAL ANALYSIS
USER'S FOLLOW-UP QUESTIONS
SEARCH FOR ANSWER
HANDLING OF IDIOMS
SEARCH FOR ANSWER
USER'S FOLLOW-UP QUESTIONS
CONVERSATIONAL TACTICS
HANDLING OF IDIOMS
USER'S FOLLOW-UP QUESTIONS
CONVERSATIONAL TACTICS
INFERENCE-MEMORY
SYNTACTIC DIALOGUE MEMORY
SEMANTIC DIALOGUE MEMORY
RECORD OF REFERENCES TO OBJECTS
INFERENCE RULES
DETAILED ANSWERS?
UNSOLICITED JUSTIFICATION OF ANSWERS?

Fig. 4c

2.1.3 Knowledge base

The external representation of the system's knowledge apart from those memory structures built up in the course of a dialogue is distributed on nine files. The purpose of this section is to introduce the syntactic structure of the expressions on the files together with an outline of their semantics, which actually is defined by the procedures which interpret the internal representations.

The overall goal in designing the syntax of the files was to achieve a simple notation which permits the files to be understood and modified even by nonexperts. Therefore many external representations are transformed into internal structures better suited for manipulations by FUZZY and LISP. The internal representation of knowledge will be described together with the components of HAM-RPM which make use of it.

2.1.3.1 Lexicon

The lexicon consists of lexical entries stored on the file LEX.DTA. Each lexical entry has the following form:

(1) (<word> ({<relation> <value>}$^{1-n}$))

Here 'word' stands for the actual graphemic string of a word, 'relation' is one of six possible identifiers and 'value' indicates the actual property possessed by the word to be defined. (The curved brackets in (1) are not part of the lexical entry but stand for the fact that more than one relation-value pair is allowed for a single word.)

The six relations are WORTART (part of speech), STAMM (stem), SYNONYM, SEMANTIK (semantics), DEKOMPOSITION and ATT-DEKOMPOSITION (decomposition of attributively used words), each of which may be followed by certain values.

WORTART defines the part of speech of a word and thus is the basic property for generating the preterminal string of an utterance. The following abbreviations for parts of speech, which in terms of (1) were called values, are interpreted by HAM-RPM at present.

(2)

$(NOM\begin{Bmatrix} MAS \\ FEM \\ NTR \end{Bmatrix})$	Noun with gender (masc., fem., neuter)
$(PRN\begin{Bmatrix} MAS \\ FEM \\ NTR \end{Bmatrix}^{*})$	Personal pronoun with gender (potentially ambiguous; see 2.6)
(ADJ)	Adjective
(ADV)	Adverb
(VRB)	Verb
(MOD)	Modal verb
$(IPRN)$	Interrogative pronoun, standing for a noun phrase, e.g. who.
$(IPRP)$	Interrogative pronoun, standing for a prepositional phrase, e.g. where.
(DET)	Article
(NEG)	Negation particle
(KON)	Conjunction
(ITJ)	Interjection

An example of a lexical entry with only a WORTART specification is:

(3) (BODEN (WORTART (NOM MAS))) for the masculine noun 'floor'.

The relation STAMM relates inflected words to their canonical form; thus the value for this relation is again a word, e.g.:

(4) (FAEHRT (STAMM (FAHR))) for the verb form 'drives'.

The canonical form of inflected nouns it is nominative singular, of adjectives it is the uninflected form and of verbs the infinitive stem. Articles, which differ in German according to gender, number and case, have the canonical form D- (definite article) and E- (indefinite article).

SYNONYM is used for replacing one word by one or more words, which already should exist as lexical entries. In the following example the contraction IM is to be expanded to in IN DEM (in the):

(5) (IM (SYNONYM (IN DEM))).

Note that only the word(s) substituted is (are) used for further processing as for instance determining the parts of speech. To map words onto other words means to take over everything that is defined for these words.

It is also possible to delete words from the input string by means of the SYNONYM relation. For these purposes the value is NIL:

(6) (MMH (WORTART (ITJ) SYNONYM NIL))

The SYNONYM relation is only inspected in lexical analysis (see 2.4), not in surface transformations (see 2. 16), so the WORTART value is not completely superfluous in example (6).

The remaining three relations (SEMANTIK, DEKOMPOSITION and ATT-DEKOMPOSITION) have a value which is the name of a procedure. This procedure is evaluated during the processing of the natural partner's input. In example (7) the procedure LEER-DEK is employed for generating an answer if the word LEER (empty) is used predicatively; in attributive use of this word the procedure LEER-ATTDEK would be evaluated in the NP-resolution component.

(7) (LEER (WORTART (ADJ) DEKOMPOSITION (LEER-DEK)
ATT-DEKOMPOSITION (LEER-ATTDEK)))

About 500 words are contained in the lexicon at present.

2.1.3.2 Idioms

The term 'idiom' is used very broadly within HAM-RPM to denote any pattern comprising several words which are treated by the system as a unit. The term thus covers the following sub-categories:

- idiomatic expressions

(8) WOHER WEISST DU DAS (replaced by WARUM)
What makes you think so why

- rigid multiple-word phrases

(9) FACHBEREICH FUER INFORMATIK (replaced by FBI)
Department of Computer Science

- conventional expressions

(10) ICH MOECHTE GERNE WISSEN... (deleted)
I'd very much like to know…

- clauses to which the system responds in a specialized manner

(11) WIE SPAET IST ES?
What's the time?

An idiom is defined in terms of one of the special procedures ERSETZEN (replace), KOMPRIMIEREN (compress) and ERWIDERN (reply), whose operation is described in detail in section 2. 4. 4. Each of these procedures takes as its first argument a pattern suitable for use by FUZZY's pattern matcher and specifies a certain type of action to be taken when this pattern is recognized within the input string.

Each idiom is stored on the property list of a 'key word' chosen at the time of its definition from among the words which are always present in an occurrance of the idiom (some words in an idiom may be optional, variable, or unspecified). Specifically, under the property IDIOME (idioms) of a word there is a list containing any idiom definitions for which that word has been chosen as a key word.

The idiom definitions – at present about 100 – are stored on a separate file called IDIOM.DTA. An entry on this file has the form which is exemplified in (12).

```
(12)   (DEFPROP WISS
           (ERSETZEN (WOHER WISS DU D- ) (WARUM))
           (ERSETZEN (ICH MOEG (*OPT GERNE) WISS) NIL))
        IDIOME)
```

2.1.3.3 Word-order rules

Word-order rules are contained in the file TRANS.RUL and are used to transform shallow structures of questions or commands into the structure of an affirmative proposition
(see 2. 9.1). The syntax of each word-order rule is:

(13) (<structural changed> <structural description>).

The structural description is a list of the parts of speech of the whole structure to be transformed. The structural change is a list of integers specifying a rearrangement of the shallow structure.

(14) ((2 1 3 4) VRB NOM PRP NOM)

for instance transforms the shallow structure ((VRB (STEH)) (NOM (STUHL1)) (PRP (VOR)) (NOM (TISCH1))) according to the structural change into:

(15) ((NOM (STUHL1)) (VRB (STEH)) (PRP (VOR)) (NOM (TISCH1))) .

50 word-order rules are defined at present.

2.1.3.4 Syntax rules

The file BASIS.RUL contains rules which are interpreted by the parsing process (see 2.9.2). The notation for rules is similar to the one commonly used for context-free grammars. A category is expanded into dominated categories, the expansion being indicated by an arrow. Neither the number of the dominated categories nor their name are restricted in these rules. An example is:

(16) (S → NP VP)

If the parts of speech are regarded as terminal symbols there are at present six preterminal categories (S, NP, VP, PP, VPA, AP). At present BASIS.RUL consists of 24 rules.

2.1.3.5 Transformation rules

Transformation rules for generating the semantic representation of an input are represented on the file SESY.PAT. This external knowledge base also contains patterns used by other components of HAM-RPM, such as the NP-pattern (see 2.8.2) and the patterns for 'which'- and 'how many'-questions (see 2.7). The syntax of transformation rules is described in section 2.11.1.

2.1.3.6 Inference rules

DEDUCE-procedures (see 1.2.4) and ASSERT-procedures (which correspond to the 'antecedent theorems' of PLANNER) of the programming language FUZZY are contained in the file INFER.PRC. The majority of these procedures are used in connection with changes in the domain of discourse (see 2.2.2), lexical analysis (2.4) and the search for an answer (2.12).

2.1.3.7 Conceptual semantic network

The conceptual semantic network is represented in the file BEGRI.FUZ. Each entry consists of an arc connecting two concept nodes. The general syntactic structure is shown in (17).

(17) ((< arc-name> <concept1> <concept2>) . z-value)

The six symbols which are defined as primitive arc-names are introduced in this section. The semantics of arcs is implicitly defined by the processes interpreting the network structure. The superset relation is represented by the U-arc:

(18) ((U <concept1> <concept2>) . z-value).

Concept1 is the superset and concept2 the subset; the superset relation between 'thing' and 'object' would thus be represented by (19):

(19) ((U THING OBJECT) . 1.0)

Note that FUZZY's associative net (see 1.2.3) makes the separate representation of inverse relations – e.g. in this case, the subset relation – unnecessary.

Names of dimensions used to describe particular types of objects are introduced by the E-arc:

(20) ((E <type> <dimension>) . z-value)

Dimension is a linguistic variable (ZADEH 1974) which is also associated with particular properties via the V-arc:

(21) ((V <dimension> <property>) . z-value)

The assertions below represent the fact that objects have a colour and some properties on the dimension 'colour'.

(22) ((E OBJECT COLOUR) . 1.0)
((V COLOUR BLACK) . 1.0)
((V COLOUR RED) . 1.0)
⋮
((V COLOUR BROWN) . 1.0)

Properties inherent to concepts or classes of concepts are attached to them by the D-arc:

(23) ((D <concept> <property>) . z-value)

Examples are:

(24) ((D OBJECT SOLID) . 1.0)
((D PIANO BLACK) . 0.7)

The properties of concepts are mostly linked to a linguistic variable via the V-arc. The D-arc in the conceptual semantic network corresponds to the REF-arc of the referential semantic network (see next section).

Finally two arcs represent the relation between concepts and parts. The T-arc links parts to wholes, the z-value representing the degree of necessity of the assertion.

(25) ((T <concept> <part>) . z-value)

In HAM-RPM the z-value of the T-arc corresponds to a linguistic hedge (see Report No. 5).

The ANZAHL-arc (number-arc) is used to represent the number of parts of a concept, the degree of necessity again being expressed by the z-value.

(26) ((ANZAHL <T-arc> <value>) . z-value)

The value in (26) may be either a number or a natural-language quantifier. Using the knowledge of the examples (27) it can be inferred that a tree (BAUM) usually has branches (AST) and that, more often than not, it has many (VIELE) branches.

(27) ((T BAUM AST) . 0.8)
((ANZAHL (T BAUM AST) VIELE) . 0.6)

For a formal definition of the semantics of the primitive arcs the reader is referred to Report No. 5. For further examples and a discussion of part relations see section 2.7.2.

At the end of the next section Fig. 5 shows a section of the conceptual and the referential semantic networks of WORLD3.

2.1.3.8 Referential semantic network

The general structure of referential relations stated in the file REF.FUZ is the same as that of conceptual relations (see (17) in the previous section). Relations of the referential semantic network describe features of objects in the current state of the domain of discourse. Only relations in this network are subject to change in the course of a dialogue (see 2. 2. 2).

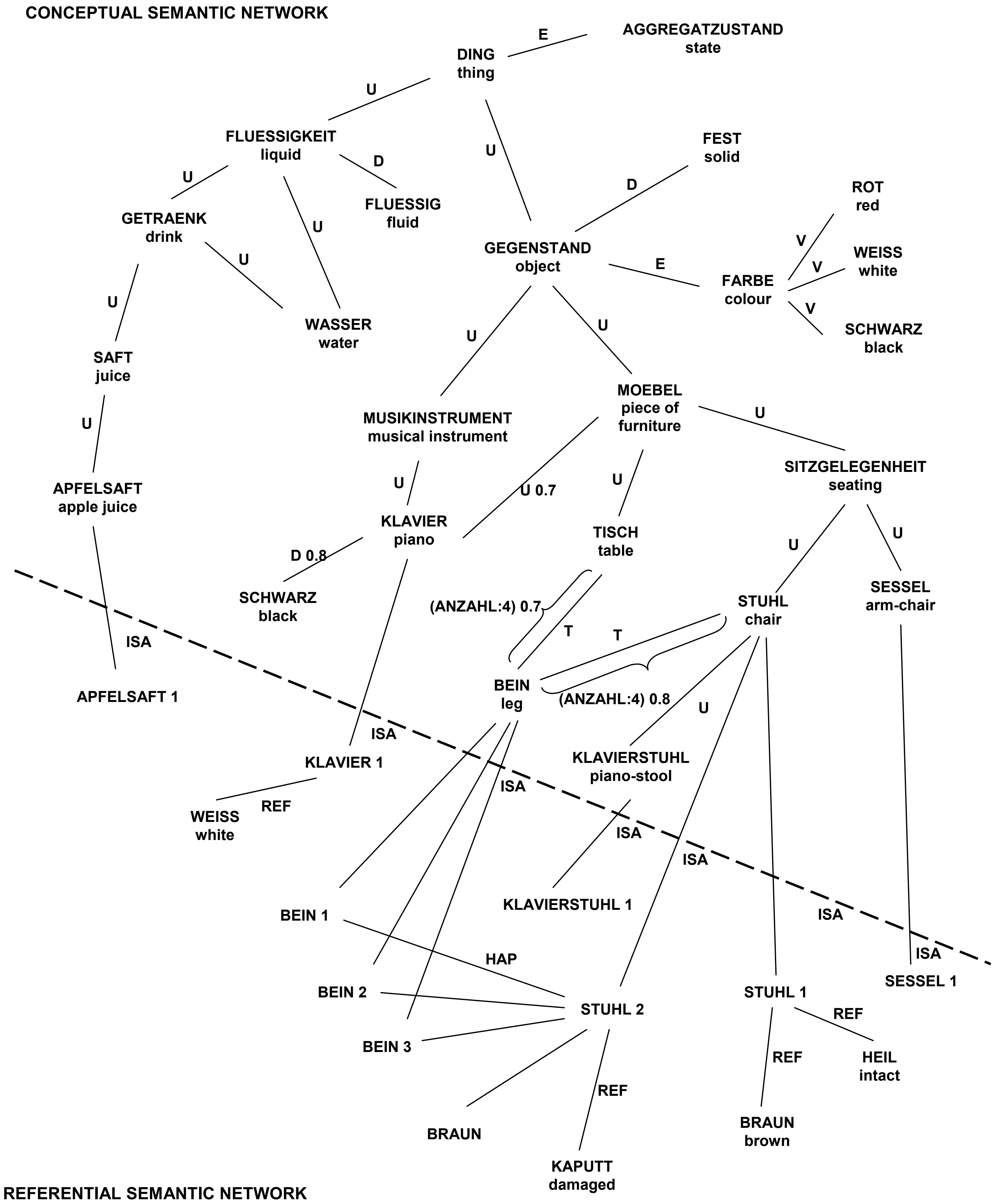

CONCEPTUAL SEMANTIC NETWORK
DING
thing
E
AGGREGATZUSTAND
state
U
U
FLUESSIGKEIT
liquid
D
FLUESSIG
fluid
FEST
solid
D
U
GETRAENK
drink
U
U
ROT
red
V
V
WEISS
white
GEGENSTAND
object
E
FARBE
colour
V
SCHWARZ
black
U
SAFT
juice
WASSER
water
U
U
U
MOEBEL
piece of
furniture
MUSIKINSTRUMENT
musical instrument
U
U
SITZGELEGENHEIT
seating
APFELSAFT
apple juice
U
U 0.7
U
KLAVIER
piano
TISCH
table
U
U
D 0.8
SCHWARZ
black
(ANZAHL:4) 0.7
SESSEL
arm-chair
STUHL
chair
T
T
ISA
APFELSAFT 1
BEIN
leg
(ANZAHL:4) 0.8
U
ISA
KLAVIER 1
KLAVIERSTUHL
piano-stool
ISA
REF
WEISS
white
ISA
ISA
KLAVIERSTUHL 1
ISA
BEIN 1
ISA
HAP
SESSEL 1
BEIN 2
STUHL 2
STUHL 1
REF
BEIN 3
REF
REF
HEIL
intact
BRAUN
BRAUN
brown
KAPUTT
damaged
REFERENTIAL SEMANTIC NETWORK

Fig. 5

The ISA-arc links each instance ('token') to its concept node ('type') in the conceptual semantic network.

(28) ((ISA <token> <type>) . z-value)

The REF-arc introduces referential properties of objects, such as a certain colour.

(29) ((REF <token> <property>) . z-value)

The properties are usually values of a linguistic variable in the conceptual network (see (21) in the preceeding section). A slightly different type of the REF-arc contains a numeric value which refers to a dimension introduced in the conceptual semantic network. An example of this arc type together with appropriate inference rules is given in section 2.12. 3.

The third primitive symbol is the HAP-arc (has as parts) which links actual parts to actual objects.

(30) ((HAP <token> <part-token>) . z-value)

A part of the referential semantic network as specified in the file for WORLD3 is shown below:

(31) ((ISA STUHL2 STUHL) . 1.0)
((REF STUHL2 BRAUN) . 0.8)
((ISA BEIN1 BEIN) . 1.0)
((HAP STUHL2 BEIN1) . 1.0)

Fig. 5 shows a graphic representation of sections of the conceptual and referential semantic networks taken from WORLD3.

2.1.3.9 Spatial data and visual salience

The entries on the file GEO.NET describe the position and the visual salience of all tokens in the current domain of discourse. The internal representations into which these data are transformed are described in sections 2.13.1 and 2.13.3.

An entry beginning with ORT (location) represents the position of an object in the horizontal plane:

(32) ((ORT <token> <coordinates>) . 1.0).

Entries beginning with the word SENKRECHTE (vertical) describe vertical relations between tokens using natural-language relation words such as AUF (on):

(33) ((SENKRECHTE <relation> <token1> <token2>) . z-value) .

The visual salience of each token of the domain is represented by the AUFFAELLIG-arc (salient-arc):

(34) ((AUFFAELLIG <token>) . z-value)

Here the z-value is interpreted as the degree of salience.

Example (35) is an excerpt from the external representation of such data for WORLD3.

(35) ((ORT GLAS1 (3. 9.)) . 1.0)
((ORT TISCH2 (3. 9.)) . 1.0)
((SENKRECHTE AUF GLAS1 TISCH2) . 1.0)
((AUFFAELLIG GLAS1) . 0. 2)

2.2 CONDUCTING A DIALOGUE WITH HAM-RPM

2.2.1 Starting the system

Human engineering has played a key role in the designing of the interactive software environment in which the natural-language dialogue processing of HAM-RPM is embedded. HAM-RPM has been designed to be a highly flexible on-line system. Some of the features provided include elaborate tracing and help facilities, error protection and error protocol mechanisms, batch-mode processing and time measurement.

The purpose of this section is to provide a short overview of the interaction with the user and the processing done by HAM-RM before the actual natural-language conversation begins. The HAM-RPM user manual (Memo No. 6) provides detailed information on the operation of the system.

At present the user can converse with HAM-RPM about three different domains: two different interiors of living rooms (WORLD1 and WORLD3) and a natural traffic scene (WORLD2) which have been introduced in section 2.1.1. The user chooses a domain of discourse by typing on monitor level DO WORLD n (where n ranges from 1 to 3).

First the system types a list of the names of all files which are automatically included in all current versions of the system as domain-independent knowledge sources (e.g. idioms, transformations). Then the loading of files which contain domain-specific knowledge is protocolled. These protocolls enable the user to access the largely self-explanatory external representations of knowledge underlying the behaviour of HAM-RPM (see 2.1.3).

It should be emphasized that not all knowledge sources are actually loaded into core memory; instead, some are only indexed to permit random disc access during the conversation (see 2.4.3 and 2.13.3). This approach allows us to interface HAM-RPM to large domains of knowledge.

During the loading of the knowledge a good deal of processing is done, such as the mapping of the external data onto various internal representation structures and the extraction of specifc information (e.g. a list of all physical objects in the domain, the maximal length of the right-hand side of a syntax rule) needed for efficient processing of the subsequent natural-language dialogue.

Having brought in the knowledge base the system interacts with the user in a menu-selection mode. In this dialogue the user can select certain options which specify the processing mode for the subsequent conversation. For example, the user can request that informative intermediate results be displayed to let him know how the processing of his input is proceeding. This is called the 'trace mode'. The user can choose to operate the system not in the usual interactive mode but in batch mode: a file prepared by the user containing a large number of test sentences is processed and the resulting dialogue is protocolled on a file. This is extremely useful for debugging and for verifying that particular changes that have been made in the program do not produce unexpected and undesirable side-effects.

All options may be changed during the conversation after the command (IO-OPTIONS) has been typed in.

When the user has specified his options, the system initializes all dialogue parameters and dialogue memories (see 2.14). Note, incidentally, that the user can re-initialize the system to the initial dialogue state by typing (HALLO!), if he wishes to undo certain effects of the conversation dynamics implemented in HAM-RPM. On the other hand, when the conversation has been interrupted, a command of the form (JETZT... (Now...)) (e.g. (JETZT GEHTS WEITER! (Now let's go on!))) allows the user to continue his conversation starting from the latest dialogue state.

HAM-RPM signals that it is ready to begin the conversation by greeting the user with an expression like GUTEN ABEND! (good evening!) depending on the time of day. When HAM-RPM outputs its prompt character '?', the user may type in natural-language expressions or LISP, FUZZY and AIMDS code. This means that a user who knows these programming languages and wishes to mix interactions with HAM-RPM and interactions with the programming system, can do so easily. This allows the system's designers to operate in an efficient extend-and-test-mode.

2.2.2 Changes in the domain of discourse

As MCDERMOTT 1978 has pointed out, many issues involving the organization of an episodic memory and the processing of temporal references are still open. Although the research reported in this document has up to now been focussed on conversations about static scenes, in the traffic scene (WORLD2) we have introduced dynamic elements, whose properties change with time independently of the actions of the participants in the dialogue.

Changing traffic lights and moving cars and persons can provide a framework for the development of a systematic representation of time. In HAM-RPM the time between the completion of the processing of a dialogue step and the complete input of the user's next utterance is used to evaluate the procedure STATE-TRANSITIONS-IN-DOMAIN-OF-DISCOURSE.

In the current version of HAM-RPM this procedure causes the changing of the traffic lights every two minutes in real-time. Hence, in the course of the dialogue the same question involving a traffic light may be answered differently by the system. Information about the dependencies between the traffic lights and their usual behaviour is stored in HAM-RPM as FUZZY antecedent procedures, so that the state of one single traffic light can imply the state of others in the scene.

2.3 CATEGORIES OF INPUT BY THE HUMAN PARTNER

Three types of initial utterances of the natural partner (as opposed to follow-up questions and answers to the system's questions) must be discriminated to guide the system's process of analysis: questions, commands and assertions. At present the understanding capability of HAM-RPM is largely restricted to initial input of type 'question'.

These types of utterances can be distinguished by simple means as e.g. punctuation marks or by more sophisticated and linguistically justified criteria as syntactic structure or illocutionary verbs.

Apart from some patterns used for recognizing speech acts which directly influence the behaviour of the system (see 2.4.4) HAM-RPM interprets every initial utterance by the natural partner as a question, not caring about punctuation marks. Thus an input syntactically structured like an assertion is regarded as a question - which in spoken utterances would be marked by prosodic features. For instance the processing of IST DER STUHL BLAU? (Is the chair blue?) and DER STUHL IST BLAU? (The chair is blue?) will be identical, apart from word-order transformations (see 2.9.1).

Two minor components of HAM-RPM, which are not invoked until after the preterminal string has been generated (see 2. 4), will be mentioned in this section because they categorize the complete utterance of the natural partner.

Two question types are established: wh-questions (including How … ?) and yes/no-questions. This distinction predicts in several ways the process of finding an answer and the structure of the answer. In artificial-intelligence systems these two question types are used widely.

Furthermore input sentences are divided into a proposition and a modal component. 'Negation' and 'Potentiality' are analyzed in HAM-RPM as modal components ranging over the whole input sentence. Modality is detected on the basis of the parts of speech NEG and MOD.

Fig. 6 illustrates this for the sentence KANNST DU DAS WASSER NICHT TRINKEN? (Can't you drink the water?).

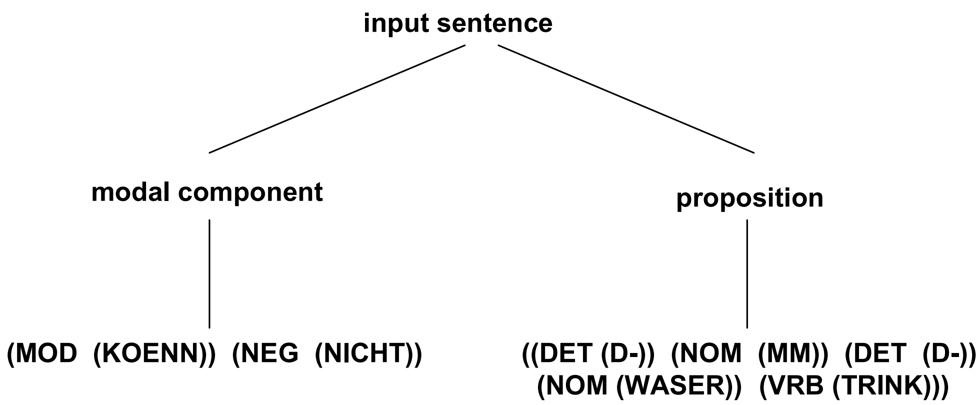


Fig. 6

Having recognized a modal word the input sentence is reduced by it and the kind of modality is bound to variables in the knowledge base of the system, thus allowing the use of modality as a processing parameter for the actual sentence.

At present modal expressions ranging over parts of the input sentence are not analyzed correctly. However, the system itself makes use of the restricted scope of modality (see 2.15).

2.4 LEXICAL ANALYSIS

2.4.1 The role of lexical analysis in HAM-RPM

The task of the component of HAM-RPM called 'lexical analysis' is twofold:

- to transform the string typed in by the user into a string of identical meaning which contains only canonical forms, i. e. words or other symbols about which the system has the information required by the process of syntactic and semantic analysis which are to follow.
- to construct the preterminal string in which these canonical forms are paired with their parts of speech.

The second of these two tasks is nearly trivial - the part of speech of a canonical form is easily determined - but HAM-RPM's handling of the first one will be described in some detail.

2.4.2 Types of lexical transformations of the sentence

Finding the canonical forms for an input sentence requires, among other things, the replacement of words which are not themselves canonical forms by other words. At present three categories of 'replacement' are defined in HAM-RPM. The categories STAMM (stem) and SYNONYM were described in section 2.1.3.1. Regarding the former, it should be noted that in the present version of HAM-RPM the information contained in an inflectional suffix is simply dropped. This is because at present the program’s processes of syntactic and semantic analysis make no use of such information anyway. When these processes have been improved, a marker representing the information contained in the removed suffix will have to be associated with the stem of the word in question.

The third category, VERBESSERUNG (correction), involves the replacement of incorrectly spelled or typed words with their correct forms.

In addition to 'replacements', 'idioms' - as defined in section 2.1.3 - must be handled before the preterminal string can be created.

The next two sections describe in more detail than was possible in section 2.1.3 how these two types of transformations are realized.

2.4.3 The representation of 'replacements'

Information about the replacements admissible for specific words, e.g. irregular forms and synonyms, is stored declaratively on the file LEX.DTA, as is all other lexical information on particular words. Because this file is very large, it isn't loaded into working memory at the beginning of the dialogue. Rather, it is 'indexed', i.e. the byte address of each entry in the file describing a word is stored in working memory on the property list of that word so that the entry can be found quickly using the random-access facility of Rutgers-UCI-LISP.

On the other hand, the system often acquires new lexical information during the course of the dialogue, either deriving it itself by applying general rules, as described below, or getting it directly from the dialogue partner (see 2.4.5).

This new information is stored in a special CONTEXT of FUZZY's associative net called LEXICON. (It would be impractical to update the file LEX.DTA with each new piece of information acquired, and in any case it is useful to be able to use the more sophisticated mechanisms provided by FUZZY in dealing with this new information; this is easiest when the information is stored in the associative net.)

The form of these assertions is shown by the following examples:

(1) (L VERBESSERUNG AGGREGATSZUSTAND (AGGREGATZUSTAND))
(L STAMM BILDER (BILD))
(L SYNONYM GREIS (ALTER MANN))
old man

A list is maintained of all new words learnt from the dialogue partner during the dialogue. After this interaction in natural language is over, these words are presented to the user individually for inspection, and he is asked to decide whether a word is worth saving or not. A fresh version of LEX.DTA is then made which includes the new words.

Almost all of the extensions to HAM-RPM's lexicon come about in this way - even the system's designers use this method rather than editing LEX.DTA directly.

Since a piece of specific lexical information may be stored in either one of the two places, a modified form of GOAL (FUZZY) (see 1.2.4), called LEX-GOAL, is used for determining the properties of words. LEX-GOAL first performs a FETCH in the CONTEXT LEXIKON of the associative net. If this FETCH fails, it LEX-GOAL checks whether the word has an address for the file LEX.DTA on its property list: if so, it reads the information on the word from the file. Only if the desired information is neither in the net nor on the file does LEX-GOAL try using DEDUCE-procedures. Incidentally, this checking in two places isn't as time-consuming as it might at first appear, since each of the checks fails very quickly if there is no information on the word in the place checked.

FUZZY DEDUCE-procedures (see 1.2.4) stored on the file INFER.PRC (see 2.1.3.6) are used to represent general knowledge about spelling correction and morphological analysis.

Their calling patterns have the same general form as the entries in the CONTEXT LEXIKON. For example, a procedure for removing certain adjective suffixes has the calling pattern

(2) (L STAMM ?WORT ?STAMM)

since what it can do is find a stem for a given word. A successful application of this procedure returns an instantiation of its calling pattern, e.g.

(3) (L STAMM BRAUNEM (BRAUN))

Similarly, there is a procedure for analysing diminutive nouns such as BAEUMCHEN (little tree). Its successful application returns an expression like

(4) (L SYNONYM BAEUMCHEN (KLEIN BAUM))

So far little effort has been devoted to the extension of HAM-RPM's repertoire of lexical rules. As can be seen, however, this method of representing such knowledge permits it to be extended in a modular manner.

2.4.4 The implementation of 'idioms'

As was mentioned in section 2.1.3.2, an idiom is defined in terms of a call to one of three special procedures.

Calls to ERSETZEN (replace) have the form

(5) (ERSETZEN <old-pattern> <new-pattern> <LISP- expression>)

If 'old-pattern' can be matched against the sentence, the corresponding part of the sentence is replaced with 'new-pattern' (cf. the example in section 2.1.3).

If any 'LISP-expressions' are present, they are evaluated one after the other. This makes it possible for the use of certain expressions to have side-effects on the system's behaviour:

(6) (ERSETZEN (?? AUSFUEHRLICH ANTWORT) NIL (SETQ AUSFUEHRLICHKEIT 3))

The first part of the question

(7) BITTE AUSFUEHRLICHER ANTWORTEN: WAS STEHT HINTER DEM BAUM?
Please answer in more detail: what's behind the tree?

is deleted by the system, but it results in a resetting of the global variable AUSFUEHRLICHKEIT, which represents the partner's presumed interest in detailed answers (see 2.14.2). The system's answer to this and the succeeding questions would therefore be at its highest level of detail.

The procedure KOMPRIMIEREN (compress) is a simpler version of ERSETZEN for certain rigid, multiple-word phrases.

(8) (KOMPRIMIEREN <word>)

The concepts corresponding to such phrases are represented conveniently by single words consisting of the several words of the phrase compressed together with blanks between theme. Thus the idiom

(9) (KOMPRIMIEREN RECHTS NEBEN)

results in the two words RECHTS NEBEN (to the right of) being replaced with the single word RECHTS/ NEBEN.

Finally, the procedure ERWIDERN (reply) permits the system to give quick responses to particular utterances which are conceptually simple, but difficult to recognize with the usual processes of analysis. Its second argument is a LISP-expression which evaluates to an answer which is passed directly to the surface-transformation component (see 2.16), bypassing all of the usual intervening processes.

(10) (ERWIDERN (WIE SPAET SEIN ES) (LIST (TIME-OF-DAY)))
What's the time

2.4.5 Clarification dialogues

Whenever a word in the sentence is not recognised as a canonical form and cannot be eliminated by any lexical operation, the system initiates a clarification dialogue with the human partner, typing a question of the form

(11) WAS HEISST '<word>'?
What's the meaning of '<word>'?

Since the partner's answers to such questions are invariably elliptical, they aren't subjected to the usual syntactic analysis described in section 2.9. Rather, the system attempts to match them against an (easily extensible) set of patterns, each of which has one of a limited number of consequences. The following three interchanges show how the three assertions listed in (1) above might be ADDed to the CONTEXT LEXIKON as a result of clarification dialogues:

(12)	WAS HEISST 'AGGREGATSZUSTAND'?	ICH MEINTE AGGREGATZUSTAND. I meant Aggregatzustand.
(13)	WAS HEISST 'BILDER'?	EINE FORM VON 'BILD'. A form of 'Bild'.
(14)	WAS HEISST 'GREIS'?	DASSELBE WIE 'ALTER MANN'. The same thing as 'old man'.

If the user types something like NICHTS (nothing), NIL is recorded as a 'synonym' of the word in question, so that it will in the future be deleted from any sentence in which it occurs (unless it is recognized as part of an idiom: see 2.4.9). This an important expedient for words like EIGENTLICH (actually) and DENN (then), which occur frequently in colloquial German but which could not at present have any effect on the way HAM-RPM answers a question.

If the partner's answer is something like

(15) LASSEN WIR DAS!
Forget it!

the system abandons the processing of the current question and asks for the next one.

Note that the replacements supplied by the human partner are not subjected to any analysis before they are ADDed to the lexical data base. For example, the inflected form ALTER of the adjective ALT (old) in the definition of GREIS supplied in (14) is retained. The consequences of this are discussed at the end of section 2.4.9.

2.4.6 The control structure for 'replacements'

By 'control structure' is meant here a scheme which specifies the order in which the applicability of the various lexical operations is to be determined and the order in which they are to be carried out.

A different control structure is used for replacements than for idioms, because of an important difference between them: whereas the former only changes the sentence at one point, the latter may alter the form of the entire sentence.

A pass through the sentence making replacements involves the repeated execution of essentially the following loop until the end of the sentence has been reached:

(16) 1. Determine whether the next word in the sentence is:

A. a canonical form

B. a word for which a replacement has been found

C. neither of these, i. e. an 'unrecognized word'

2. If A, then pass over the word and go to 1.

If B, then substitute the replacement (which may consist of several words) for it, so that the next word is now the first word of the replacement, and go to 1.

If C, then ask the dialogue partner for help (see 2.4.7) and, after receiving it, go to 1, checking the same word again. (This time, the information just supplied will be available.)

Note that the words constituting a replacement are themselves subjected to analysis after they have been inserted. The necessity for this is clearest when the replacement has just been supplied by the dialogue partner, as in this sequence:

(17) IST DER STRASSENKREUZER ALT?
Is the 'street cruiser' old?

(18) WAS HEISST 'STRASSENKREUZER'?
What's the meaning of 'street cruiser'?

(19) DAS IST EIN GROSSES AMERIKANISCHES AUTO.
That's a big American car.

(20) WAS HEISST 'AMERIKANISCHES'?
What's the meaning of 'American'?

2.4.7 The control structure for 'idioms'

The control structure for idioms specifies a reprocessing of the sentence beginning with the first word as soon as an idiom has successfully been applied. It likewise involves the repeated application of a loop:

(21) 1. Check to see if the next word has any idioms on its property list (see 2.1.3.2).

2. If it doesn't, pass it over and go to 1.

3. Try each of the idioms to see if it can be applied successfully to the sentence; if one of them can, go to 1 and proceed from the beginning of the (now transformed) sentence.
4. (if none of the idioms listed for the word has succeeded:) Pass over the word and go to 1.

Note that if an idiom applied successfully in step 3 is of the type ERWIDERN (reply) (see 2.4.4), the process of lexical analysis is exited prematurely with a large jump to the surface-transformation component.

2.4.8 The automatic storage of derived results

As will become clear in the next section, it often happens that a piece of information derived using a general lexical rule is needed at some later point in the processing. Since the application of general rules is relatively time-consuming, it is important to store derived results so that they needn't be derived a second time.

FUZZY's procedure demons provide a simple means of accomplishing this: a special demon called MEMORY-DEMON is assigned to all lexical DEDUCE-procedures. This demon differs from FUZZY's default procedure demon (see 1.2.5) only in that, when the DEDUCE-procedure whose execution it is supervising succeeds, MEMORY-DEMON ADDs the result returned by the procedure to the CONTEXT LEXIKON of the associative net where it can be retrieved quickly by a future call to LEX-GOAL (see 2.4.3).

2.4.9 The overall control structure for lexical operations

In the course of the lexical analysis of an input sentence, three passes are made through the sentence in which individual words are replaced, and two in which idioms are handled. In these five passes, the control structures described in sections 2.4.6 and 2.4.7 are applied, with minor variations as described in the following sketch:

(i) a replacement pass

Recognized spelling errors and inflected words are replaced as a prerequisite to the handling of idioms, which can only be defined economically in terms of word stems. On the other hand, any synonyms which are recognized are not replaced at this point: this can be done only after all idioms have been handled, since an idiom is defined in terms of particular words, not in terms of particular concepts. (The same is true in English: thus 'I'm ill and tired of this' is not recognizable as a use of an idiom even if the idiom 'to be sick and tired of something' is known. 'Sick' can therefore be replaced by 'ill' only after idioms have been handled.)

If a word is not recognized at all, an assertion to this effect is added temporarily to the lexical data base, so that the system will not try again in vain to analyse it, but the dialogue partner is not yet asked for a clarification of the word. This is because the word may yet be eliminated in the course of the handling of idioms which is to follow.

(ii) an idiom pass

Some of the words marked as 'unrecognized' during the first pass may prove to be words which are only recognizable as components of idioms; these will often be eliminated from the sentence if the corresponding pattern is in fact found. Other unrecognized words may be misspellings or unknown inflections of known words. If such a word forms an essential component of an idiom occurring in the sentence, this idiom will not be recognized during this pass (see, however, the fourth pass). Otherwise, all known idioms which actually occur in the sentence will be handled correctly within this pass.

(iii) a replacement pass

This pass differs from the first one only in that unrecognized words now lead to a clarification dialogue of the sort described in section 2.4.5. Aside from any such clarification dialogues, there is little to be done in this pass, as all words except any which may have been introduced during the second pass have already been analysed during the first pass.

(iv) an idiom pass

This pass is only necessary to pick any idiom whose recognition may have been made possible only by a clarification dialogue concerning one of the words which it contains. It could be skipped if there were no unrecognized words left after the third pass.

(v) a replacement pass

This pass differs from the third one only in that any recognized synonyms are finally actually replaced in the sentence.

A complete exposition of the rationale for this relatively complicated control structure would exceed the scope of this report, but two of its distinguishing features should be mentioned.

First, a good deal of its complexity is made necessary only by the goal of robustness in dealing with words which can only be dealt with after a clarification dialogue. When no such words are present in the sentence, in effect only three passes are made.

Second, it is nowhere assumed that the words used in the definitions of replacements and idioms in the lexical data base are canonical forms, or even known stems: all words inserted into the sentence as the result of a lexical operation are analysed in just the same way as the words in the original input string (see 2.4.5, example (14)). This aspect of the control structure makes it possible to evade certain knotty problems that would be involved in any attempt to insure that the definitions in the knowledge base contained only the right sort of forms, the difficulties being aggravated by the fact that much of the updating of the date base is done in natural language by naive users. On the other hand, this relative lack of constraints on the lexical data base leads to inefficiency of processing, as the system must at some points be ready to deal with a much larger number of possibilities than it would otherwise have to take into account. It seems to us to be also psychologically less plausible. Work is in progress on an improvement of the lexical-analysis component which is more satisfactory in this respect.

2.5 ANALYSING FOLLOW-UP QUESTIONS

A typical feature of question-answer sequences in human dialogues is that some questions refer to questions just posed or to answers just given by the conversational partner. The handling of such follow-up questions is a major factor introducing dialogue coherence into conversations with HAM-RPM. In most cases such follow-up questions are elliptical and have a clear-cut illocutionary force, e.g. What else? or Why not?.

Some types of elliptical follow-up questions can easily be recognized via pattern matching without being processed by the general ellipsis-recognition process described in section 2.10. These follow-up questions are analysed in an early processing phase of HAM-RPM immediately after lexical analysis. The only two types of follow-up questions which have been implemented so far are 'why'-questions and requests for details. These are described in the following two sections.

2.5.1 'Why'-questions

Since expressions like AUS WELCHEM GRUND (for what reason) and WIE KOMMST DU DARAUF? (What makes you think that?) are transformed into the canonical form WARUM (why) by the lexical-analysis component (see 2.4.4) requests for a justification of a previously given response are recognized by simply looking for the word WARUM.

Two types of 'why'-questions are processed by the component described in this section. First, when a 'why'-question follows an answer to a user's question which resulted from failed selectional restrictions (see 2.12.3) the system explains which restrictions were violated. Second, when a 'why'-question follows an answer of the system which was derived by inferences, the system explains the reasoning underlying the answer. This is of great importance since in HAM-RPM most inferences are based on fuzzy or partial knowledge (see 2.12.3) and since the results of such reasoning are of limited use to the human dialogue partner unless the system is able to justify the underlying inference process.

A third type of 'why'-question, which occurs in clarification dialogues during NP-resolution, is described in section 2.8.7.

When selectional restrictions on predicates fail, the violated conditions are stored as the value of the global variable BEGRUENDUNG (explanation) for future use. This means that when BEGRUENDUNG is non-NIL the answer to a 'why'-question is constructed straightforwardly by passing BEGRUENDUNG to the language generation routines (for an example see 2.12.2 below). Finally BEGRUENDUNG is reset to NIL and the input sentence is marked as being incomplete, so that any elliptical inputs which may follow immediately are matched against the sentence preceding the follow-up questions. When an answer cannot simply be retrieved from the knowledge base but must be derived using inference rules, a sort of trace of the inference process is stored in a separate data base called INFERENCE MEMORY while the reasoning is being performed (see Reports Nos. 5 and 7).

If BEGRUENDUNG is NIL when an explanation of a conclusion has been requested, the inference memory is traversed to find those assumptions used in the inference chain which are on a communicatively appropriate level of detail. This is accomplished by using meta-knowledge. A thorough discussion of this topic is contained in Report No. 9.

An interesting capability of the current version of HAM-RPM is that it is able to answer iterated 'why'-questions appropriately. For example consider the following dialogue fragment:

(1) MAGST DU DIESEN APFELSAFT GERNE?
Do you like this apple juice?

(2) ICH GLAUBE JA.
I think so.

(3) WARUM MAGST DU IHN GERNE?
Why do you like it?

(4) EIN APFELSAFTGETRAENK IST OFT SUESS UND EIN APFELSAFTGE-TRAENK IST ALKOHOLFREI.
Apple juice is often sweet and apple juice is non-alcoholic.

(5) WIE KOMMST DU DENN DARAUF?
What makes you think so?

(6) EIN APFELSAFTGETRAENK IST EIN SAFT UND EIN SAFT IST ALKOHOLFREI.
Apple juice is a juice and juice is non-alcoholic.

The 'why'-question (5) is interpreted by HAM-RPM as a request to explain further the explanans given for (2) in (4). Since the first conjunct of the explanans is not derived but retrieved from the conceptual semantic net, the second conjunct, which results from a property-inheritance inference (see 2.12.2) is explained. As shown in sentence (4) HAM-RPM hedges its explanations according to the status of the knowledge used. In the current version of HAM-RPM explanations are modified by the linguistic hedges MEIST (mostly), OFT (often) and MANCHMAL (sometimes) depending on the z-value of the premises stored in the inference-memory.

When the dialogue partner has repeatedly requested explanations and the system has already mentioned all of the steps in the inference chain, the system finally outputs the message IST DOCH KLAR (It's obvious). The same response is generated when BEGRUENDUNG is NIL and the answer did not result from an inference process at all.

2.5.2 Requests for details

A variety of expressions like UND SONST (and what else) and WEITER NICHTS (nothing else) whose pragmatic meaning is a request for further information are recognized via pattern matching and are processed according to the following rather simple scheme.

When the semantic representation of an answer consists of a set of alternative items which is ordered according to relevance, usually only the item with the greatest relevance is verbalized, since the system is designed to act in accordance with the conversational maxims 'Don't make your contribution more informative than is required' and 'be relevant' formulated by GRICE 1975. This initial behaviour of the system is, however, subject to change during the dialogue (section 2.14.2).

All items not verbalized are stored under the variable RESTANTWORT (rest of the answer), so that follow-up questions which express a request for details are answered by passing RESTANTWORT to the language generation routines. As an example consider the following dialogue fragment:

(7) WAS STEHT IN DER NAEHE DES FERNSEHGERAETES ?
What's near the TV set?

(8) DER COUCHTISCH.
The couch table.

(9) UND SONST?
And what else?

(10) DAS KLAVIER UND DER KAPUTTE STUHL.
The piano and the damaged chair.

After the answer (8) has been given, the value of the variable RESTANTWORT is the list (PIANO1 CHAIR2).

When the system has already verbalised all of the items contained in the semantic representation of the answer (see 2.15), it answers NICHTS (Nothing).

Finally, as with requests for justification, the follow-up question is marked as an incomplete input.

2.6 PRONOUN RESOLUTION

There are various types of pronouns which can be analysed or generated by HAM-RPM during the various processing phases.

Interrogative pronouns such as WER (who), WAS (what), and WO (where) are recognized early when the type of the question is determined (see 2.3). Later on, the semantics of these pronouns, which is encoded in translation rules (see 2.11.1), is used to construct a semantic representation of the question.

Relative pronouns such as WELCHER (who) and WELCHES (which) are processed by the NP-resolution component (see 2.8).

In this section we present our approach to the resolution and generation of non-reflexive personal pronouns. In the current version of the system, incidentally, demonstrative pronouns occurring alone as complete noun phrases – for example DIESER (this) in WELCHE GROESSE HAT DIESER? (How big is this one?) – are treated exactly like the corresponding personal pronouns.

HAM-RPM has no trouble with the analysis of forms of the personal pronoun DU (you) or with the generation of forms of the pronoun ICH (I), as these forms in these cases always refer to the simulated dialogue partner. The generation of forms of DU is likewise straightforward, as these generated forms always refer to the human dialogue partner.

To find the antecedent of a personal pronoun, HAM-RRM uses a simple and efficient, but somewhat naive algorithm which makes use of the record of references to objects described in section 2.14.1. For every pronoun which occurs in the shallow structure of the sentence, the system performs a FETCH within the CONTEXT VORERWAEHNT which, if successful, returns an entry containing the name of the most recently mentioned object which satisfies the constraints on coreferentiality with the pronoun. This search is restricted to those objects which have been referred to during the previous six interchanges (clarification dialogues are not counted as separate interchanges). The name of the object found is substituted immediately for the pronoun in the shallow structure of the sentence; the NP-resolution process for the definite description which was the antecedent of the pronoun thus need not be repeated.

It is well known that in contrast to English in German the gender often can be used to constrain the set of potential referents of a pronoun drastically. As an example, consider the following dialogue sequence:

(1) IST DER BAUM, WELCHER RECHTS NEBEN DER AMPEL STEHT, GROSS?
Is the tree which is to the right of the traffic light big?

(2) JA.
Yes.

(3) WELCHE FARBE HAT SIE IM MOMENT?
What colour is it at the moment?

In German we know that SIE (it) which has the gender feminine, must refer to the traffic light, which has the same gender, because the gender of tree is masculine. In English the correct reference can only be determined using world knowledge. For example, the phrase 'at the moment' gives a hint that the colour of the object often changes. This would rule out the tree as a potential referent.

Since in a few cases the gender of some forms of a pronoun can be ambigous – e.g. IHM has the gender masculine or neuter – the algorithm only checks whether the intersection of the gender of the potential referent and the gender of the pronoun is nonempty.

Since objects referred to by answers of the system are also recorded, the pronoun DIESER (this one) in (6) below can be correctly replaced by the internal name of 'the brown chair':

(4) KANNST DU MIR SAGEN, WOVOR DER TISCH STEHT?
Could you please tell me what the table is located?

(5) DER BRAUNE STUHL.
The brown chair.

(6) UND WOVOR STEHT DIESER?
And what's that located in front of ?

Expressions like DARAUF (thereupon) and DANEBEN (next to it) are transformed by the process of lexical analysis (see 2.4.2) into the expressions AUF IT and NEBEN IT, where IT is treated a an artificial pronoun which can refer to objects of all genders. As an example, consider:

(7) DER KLAVIERHOCKER STEHT WO?
The piano stool is where?

(8) VOR DEM KLAVIER.
In front of the piano.

(9) WAS STEHT DANEBEN?
What's next to that?

DANEBEN in (9) is interpreted as referring to the piano.

If the system cannot find a referent for the pronoun, it will behave cooperatively, asking the user with a metacommunicative question to provide the necessary information. As an example, consider the following dialogue sequence:

(10) STEHT SIE VOR DER AMPEL?
Is she standing in front of the traffic light?

(11) AUF WAS BEZIEHT SICH 'SIE'?
What does 'she' refer to?

(12) AUF DIE FRAU, WELCHE IN DER NAEHE DES GELBEN AUTOS STEHT.
To the woman who is standing near the yellow car.

The user's answer (12) to the question (11) is analysed by the NP-resolution component (see 2.8), which eventually returns a unique name which is then substituted for the pronoun.

The record of references to objects is updated as soon as an object has either been returned by the process of NP-resolution or included in the semantic representation of the answer. In particular, in the second case, some objects will be on record as having been referred to even though they were not included in the surface structure of the answer. This way of updating the record has two advantages:

First, the algorithm can find referents for pronouns which don't have explicit antecedents in the surface structure of any previous sentence. This is demonstrated by the following dialogue sequence:

(13) GIBT ES DENN UEBERHAUPT EINEN AUFGEBROCHENEN SCHRANK?
You mean, there's no damaged cupboard here at all?

(14) DOCH.
Oh, yes, there is.

(15) WELCHE FARBE HAT DIESER DENN?
Well, what colour is it?

The positive answer (14) implies that at least one damaged cupboard has been found. The name of this implied entity, which appeared in the semantic representation of the answer and was thus stored in the record of references to objects, is substituted for the pronoun DIESER in question (15).

Second, the system can cope with pronouns which have antecedents in the very sentence in which they themselves occur. For example, in the question

(16) STEHT DAS FERNSEHGERAET EIGENTLICH LINKS NEBEN DER LAMPE, WELCHE SICH IN DER NAEHE VON IHM BEFINDET?
Is the TV set in fact to the left of the lamp which is located near it?

IHM (it) is replaced by the result of the NP-resolution process applied to DAS FERNSEHGERAET (the TV set).

A problem in using the constraints of gender for the resolution of pronouns is that in German the gender which is associated with an object in a particular case is not determined once and for all by the gender of the lexical item that was used to refer to it, but may change during the course of the dialogue. For example, consider the following dialogue:

(17) SIEHST DU DAS FERNSEHGERAET (gender: neuter)?
Do you see the TV set?

(18) JA.
Yes.

(19) WELCHE FARBE HAT DIESER GEGENSTAND (gender: masculine)?
What colour is this object?

(20) WEISS.
White.

(21) WO STEHT ER (gender: masculine)?
Where is it?

ER in (21) refers to the object first described as DAS FERNSEHGERAET in (I7). This sort of phenomenon is what makes it necessary to store the gender of the lexical item which was used to refer to an object together with the name of the object, as is described in section 2.14.1.

At present HAM-RPM has no general procedures for the processing of reflexive pronouns, possessive pronouns, or pronouns which refer to events or whole sentences instead of to objects.

The experience of a large number of dialogue sessions with naive users has shown that the simple algorithm presented here works remarkably well. This is reminiscent of HOBBS' (1976) observation that in English about 90% of all pronoun references can be resolved using purely syntactic constraints. On the other hand it is clear, as CHARNIAK (1973) has demonstrated convincingly, that for difficult cases of pronoun resolution the full range of cognitive processes involved in understanding must be available including complex common-sense inferences.

We have found that an algorithm similar to the one presented here for pronouns can also handle anaphoric noun phrases (see 2.8.6). This is indicative of the functional similarity of these two means of referring (cf. GROSZ 1976b).

The pronoun-resolution component is a major contributor to dialogue coherence in HAM-RPM.

2.7 UNDERSTANDING BY MEANS OF SENTENCE PATTERNS

Some frequent types of questions are not processed in the usual way described in the sections to follow, but rather are recognized, immediately after the resolution of pronouns, as instances of specific patterns on the basis of certain crucial syntactic categories. No complete syntactic analysis is attempted; rather, special search and inference processes are invoked to find an answer to these questions without generating a deep structure.

This short-cut processing of particular types of questions corresponds to the ability of human speakers to recognize and process frequently heard questions without going through a deep linguistic analysis each time.

In the current version of HAM-RPM certain types of 'which'- and 'how-many'-questions are handled in this manner. Several patterns specifying the position of certain characteristic syntactic entities – in other words, a partially instantiated shallow structure – are part of the knowledge base (see 2.1.3.5); these patterns match some simple 'which'- and 'how-many'-questions directly. There are, however, many more complex questions of these types which are not matched by these patterns, but which must be subjected to processing by the complete linguistic-analysis and inference components of HAM-RPM.

During the processing of those sentences which are matched by patterns, NP-resolution is carried out in the same way as for exhaustive linguistic analysis (see 2.8), though at present there are some differences. A characteristic feature of the short-cut method is the distinction between the generic and the definite or indefinite interpretation of noun phrases, which for experimental purposes is presently implemented only in this processing phase. For details on such differences, see section 2.8.9.

After each successful match with a sentence pattern, the shallow structure which has been generated is stored in the syntactic dialogue memory to permit the recognition of any elliptical sentences which might immediately follow (see 2.10).

2.7.1 'Which'-questions

Two patterns describe the shallow structures of some simple 'which'-questions:

(1) WHICH-PATTERN1:
((IPRN (WELCH)) ?? (NOM?) (VRB?) ?? (NOM ?) ??)

WELCH is the stem corresponding to the English 'which'. Here are some examples of sentences whose shallow structures would be matched by WHICH-PATTERN1:

(2) WELCHE GEGENSTAENDE SIND MOEBEL?
Which objects are pieces of furniture?

(3) WELCHE FARBE HAT DER STUHL, WELCHER RECHTS NEBEN DEM KLAVIER STEHT?
What colour is the chair that's standing to the right of the piano?

(4) WELCHE BEQUEMEN SITZGELEGENHEITEN STEHEN IN DER NAEHE VOM TISCH?
What comfortable seats are near the table?

(5) WHICH-PATTERN2:
((IPRN (WELCH)) (NOM?) (VRB ?) (ADJ ?EINGABEADJ))

(The FUZZY variable EINGABEADJ (input adjective) is referenced by the NP-generation routines described in section 2.15). An example which would be recognized via this pattern is:

(6) WELCHE MOEBEL SIND KAPUTT?
Which pieces of furniture are damaged?

The special search and inference mechanisms which were developed for analysing and answering the 'which'-questions which are matched by the above patterns will now be introduced.

Several types of question-rejections provide the human dialogue partner with information as to why an answer could not be found.

(7) WELCHE TISCHE SIND MOEBEL?
Which tables are pieces of furniture?

(8) UNSINNIGE FRAGE!
Meaningless question!

The question presupposes a superset relation between tables and pieces of furniture, whereas the actual relation is the other way around.

(9) WELCHE HAARFARBE HAT EIN TIER?
What is the colour of hair of an animal?

(10) WEISS ICH NICHT!
Don't know!

There is no conceptual property 'colour of hair' attached to animal or to a concept in the superset hierarchy of animal. Therefore the system cannot possibly answer the question on the basis of its present knowledge.

(11) WELCHE FARBE HAT EIN MENSCH?
What colour is a man?

(12) UNENTSCHEIDBAR!
Undecidable!

It is not possible to answer this question for one specific instance of man, but for certain subclasses various values (V-arcs) along the dimension 'colour' are stored. Thus for the superset 'man' no unique property of 'colour' can be determined. This rejection and the one in number (10) differ in that here some superset-concept of 'man' ('thing' in our case) is known to have the property 'colour' but there is no unique value of this property relatable to 'man'.

(13) WELCHES BILD IST GRUEN?
Which picture is green?

(14) KEIN BILD IST GRUEN.
No picture is green.

The question in (13) matches with WHICH-PATTERN2. In the current domain of discourse not even one possible referent can be discovered, nor is 'green' a conceptual property of the concept 'picture'.

Now let's turn to more positive and informative answers to 'which'-questions:

(15)) WELCHE MOEBEL SIND SITZGELEGENHEITEN?
Which pieces of furniture are seats?

(16) STUHL UND SESSEL.
Chair and arm-chair.

A superset hierarchy can be established between 'pieces of furniture' and 'seat'. All subset nodes of the latter are taken to be the answer. This dialogue example contrasts with number (7). Note that in this case the question is interpreted as concerning only conceptual knowledge.

(17) WELCHER STUHL STEHT DIREKT LINKS NEBEN DEM TISCH?
Which chair is standing directly to the left of the table?

(18) DER KAPUTTE STUHL.
The damaged chair.

Detecting a preposition in the shallow-structure of a 'which'-question which matches WHICH-PATTERN2, the system assumes that a local relation between certain objects is asked for. The processes for finding the objects described are part of the NP-resolution component (see 2.8) and are likewise employed for 'how many'-questions (see 2.7.2). Roughly speaking the task is solved by finding all referents of the definite noun phrase 'the chair which is standing directly to the left of the table'.

(19) WELCHE FARBE HAT DER RUNDE TISCH?
What colour is the round table?

(20) WEISS.
White.

The system cannot establish a superset hierarchy between 'table' and 'colour' as in example (17) and therefore tries to interpret 'colour' as a conceptual dimension. An answer is found by looking for a inferential property of that specific table which is a possible value of the concept 'colour'.

If this inference strategy had not been successful or the object had been interpreted generically the system would have tried to answer the question solely on the basis of its conceptual knowledge as in examples (22/23) and (24/25).

(22) WELCHEN AGGREGATZUSTAND HAT EIN TISCH?
What's the state of a table?
(23) FEST.
solid.

A superset concept of 'table' (for instance 'object') is known to be solid. Making use of inheritance hierarchy this is also true for the subsets.

(24) WELCHE FARBE HAT EIN KLAVIER?
What colour is a piano?
(25) NORMALERWEISE SCHWARZ.
Usually black.

If properties inferred from the conceptual semantic network are not true with absolute certainty, the extent of uncertainty is expressed by a linguistic hedge as in example (25). For a more detailed description of the use of linguistic hedges in generating answers see section 2.7.2 and Report No.7.

Modifying the answers to 'which'-questions according to the source of knowledge employed is analogously realized with shallow-structures matched by WHICH-PATTERN2:

(26) WELCHES MUSIKINSTRUMENT IST TEUER?
Which musical instrument is expensive?
(27) JEDES MUSIKINSTRUMENT IST TEUER.
Every musical instrument is expensive.

The concept of 'musical instrument' is known to have the conceptual property (D-arc) 'expensive'; thus the answer can be inferred from conceptual knowledge and is expressed as a general statement.

On the other hand properties attached to particular objects in the domain of discourse are presented in the inferential semantic network; in the answer to a 'which'-question this is indicated by the complete enumeration of the objects in question:

(28) WELCHE MOEBEL SIND WEISS?
Which pieces of furniture are white?
(29) DAS KLAVIER UND DER KLEINE TISCH.
The piano and the small table.

2.7.2 'How many'-questions

As in the case with 'which'-questions two patterns are provided to identify simple types of 'how many'-questions:

(30) HOW-MANY-PATTERN1:
((IPRN (WIEVIEL)) ?? (NOM ?) (VRB ?) ?? (NOM ?) ??)
how many

Examples:

(31) WIEVIELE KAPUTTE MOEBEL SIEHST DU?
How many damaged pieces of furniture do you see?

(32) WIEVIELE BRAUNE SITZGELEGENHEITEN STEHEN IN DER NAEHE VOM TISCH?
How many brown seats are standing near the table?

(33) WIEVIELE BEINE HAT EIN MENSCH?
How many legs has a man got?

(34) HOW-MANY-PATTERN2:
((IPRN (WIEVIEL)) ?? (NOM ?) (VRB ?) ?? (ADJ ?))

Examples:

(35) WIEVIELE GEGENSTAENDE SIND HEIL UND WEISS?
How many objects are intact and white?

The analysis algorithm for simple 'how-many'-questions distinguishes between two semantic groups: One asking for the number of objects with certain properties or relations to other objects, the other one asking for the number of parts of objects.

All question types of the first group are handled with processes which are part of the NP-resolution component (see 2.8). Roughly speaking these question types are transformed into definite noun phrases whose scope of extension, i.e. the number of possible referents, is the answer. As with 'which'-questions noun phrases occurring in the shallow structure of 'how many'-questions are resolved beforehand. The entire analysis capacity of the NP-resolution component is therefore employed twice during the processing of these question types.

Examples of the transformation of 'how many',-questions into definite noun phrases:

(36) WIEVIELE HEILE GEGENSTAENDE HIER SIND BRAUN?
How many intact objects here are brown?
→ scope of extension of DER HEILE BRAUNE GEGENSTAND
The intact, brown object

(37) WIEVIELE STEUHLE STEHEN DIREKT NEBEN DEM TISCH?
How many chairs are standing directly next to the table?
→ scope of extension of DER STUHL, WELCHER DIREKT NEBEN DEM TISCH STEHT.
The chair which is standing directly next to the table.

Special search and inference processes are at present being developed for the second group of 'how many'-questions, questions about parts of objects. Here the representation of parts in the knowledge base of HAM-RPM (see 2.1.3) will be described along with algorithms implemented so far. Then problems arising with the interpretation of the parts representation will be discussed.

Three types of arcs represent the relation between concepts or objects and their parts in the semantic networks of HAM-RPM.

The conceptual semantic network contains two of these arcs

(38) ((T <concept> <part of concept>) z-value)
(39) Example: ((T MAN LEG) . 1.0)
(40) ((ANZAHL (T <concept> <part of concept>) <quantity>) z-value)
(41) Example: ((ANZAHL (T MAN LEG) 2) . 0.9)

Using these representation constructs, part relations are established on the conceptual level which allow inferences about compound objects and the number of their parts, each associated with a z-value (for further inferences on the T-arc see Report No. 5).

The third type of arcs belongs to the inferential semantic network. It links special instances of parts to objects in the domain of discourse:

(42) ((HAP <object name> <part name>) z-value)
(43) Example: ((HAP MAN2 LEG1) . 1.0)

A HAP-arc will only be stated in the inferential network where it may be necessary to address oneself to specific parts of specific objects, or if the number or the properties of specific parts differ from the default values in the conceptual knowledge. For example one of the chairs in the living-room scene is broken. It has only three intact legs, although four legs are said to be parts of the concept 'chair'. The discrepancy is represented with four HAP-arcs linking the object name to four instances of 'leg', three of these with the renferential property 'intact' and to one with the property 'broken'.

Answering 'how many'-questions which concern parts of objects is achieved by determining the component of the knowledge base presumably containing relevant information. If the NP-resolution component decided to interprete the object noun phrase (i.e. the last noun phrase) as generic (see 2.8.9) the conceptual network is searched for suitable T-arcs. T-arcs are involved in superset inheritance-hierarchy, so supersets of the concept are inspected if the direct attempt fails. If an appropriate T-arc is found the number value of the ANZAHL-arc, which may be either a number or a quantifying word such as VIELE (many), answers the question. As it was found in the conceptual semantic network the z-value of the ANZAHL-arc is interpreted as a degree of certainty. This degree is expressed by linguistic hedges (see above example (25)), which are assigned to possible subranges of the z-value. For instance NORMALERWEISE (usually) represents the interval [0.8, 0.9] of z-values; MANCHMAL (sometimes) represents the interval [0.4, 0.5].

Here is an example of a 'how many'-question answered by conceptual knowledge:

(44) WIEVIELE BEINE HAT EIN STUHL?
How many legs has a chair?
(45) NORMALERWEISE VIER.
Usually four.

When the NP-resolution has assumed that the last noun phrase refers to a specific object, the HAP-arcs of the name of this object are examined. The number of HAP-arcs – if there are any – satisfying the attribute restrictions is the answer.

Here is an example of a 'how many'-question answered by referential knowledge:

(46) WIEVIELE HEILE BEINE HAT DER STUHL, WELCHER LINKS NEBEN DEM TISCH STEHT?
How many intact legs has the chair which is standing left to the table got?

(47) DREI.
three.

It may be the case that there are no corresponding HAP-arcs associated with the object name. Parts of objects may be concealed by other objects in the domain of discourse. The system then backs up and tries to find an answer on the basis of its conceptual knowledge as described above, using instead of the object names the concept, which is linked to the object name by an ISA-arc.

One major problem with inferences about parts involves the part-inheritance hierarchy: How should parts of parts of objects be represented and what is to be inferred using a given representation scheme?

A second question is: is the transitivity of parts hierarchies limited to a certain degree? For example, all subparts of objects relatable to an object via T-arcs might not represent a single quantifying assertion. Third, how can parts hierarchies, which are embedded in superset-inheritance hierarchies, be verbalized if the system is required to reason about its inferences?

Finally, it seems that the use of specific linguistic hedges expressing the degree of certainty of an assertion has to be dependent on the concepts involved. For example, to say 'a passenger car nearly always has four wheels' sounds perfectly normal, whereas 'a man nearly always has two legs' is at best somewhat awkward. Partitioning the semantic networks for special purposes – for example, into 'object concepts' and 'living beings' – and interpreting degrees of certainty according to this partitioning may be helpful, not only where part relations are concerned.

Future work within HAM-RPM on parts representation and inferences will include the topics just sketched and might produce a more elaborate structuring of the semantic networks.

2.8 NP-RESOLUTION

2.8.1 Objectives

In HAM-RPM semantic, pragmatic and syntactic processes cooperate closely in analysing the natural partner's input. One major component which attempts to interpret parts of the input chiefly by means of semantic and pragmatic knowledge is the one responsible for NP-resolution.

The first task of this component is to determine whether a given noun phrase is used as a generic, a definite or an indefinite description (cf. OOMEN 1977, NORMAN/RUMELHART 1975, ch. 3). After having made sure of the type of determination or – if that is not possible – after a hypothesis about it has been put forth, all of the knowledge available to the system at the moment is employed to determine the reference of the noun phrase.

The reason for analysing noun phrases before the complete input sentence is parsed is that a natural partner does not necessarily wait for a whole utterance which might be syntactically analysable before trying to process the parts of the utterance already perceived (cf. GERSHMAN 1977, in contrast to WINOGRAD 1972, RITCHIE 1977). This is often observed in natural dialogues (cf. GROSZ 1976a), when either an utterance is not completed at all or the speaker hesitates while looking for an appropriate continuation. The listener in these cases is able to detect semantically or pragmatically ill-formed or ambiguous noun phrases and interrupt the discourse with rejections, reassurances or clarification questions.

We do not claim that noun phrases necessarily and always have to be or even can be resolved before the whole sentence has been parsed; but rather that the inferential analysis of partial sentences is one characteristic feature of natural dialogue behaviour. This ability has to be taken into account in the simulation of a natural dialogue partner.

2.8.2 Definite descriptions

Now let's take a closer look at the analysis of noun phrases in HAM-RPM. In this section we shall restrict ourselves for the sake of simplicity to the assumption that the noun phrase to be analysed is a definite description (for indefinite and generic descriptions see 2.8.8 and 2.8.9).

Input to the NP-resolution component is the shallow structure generated so far in the course of the analysis. In a first step noun phrases are looked for using the following pattern:

(1) (?? DET ?? NOM ??)

This pattern describes any segment between an article (DET) and a noun (NOM). If it matches a segment of the input sentence, a relative clause directly following the norm is looked for. Thus the structure of a noun phrase identified by the NP-resolution is represented in (2). With definite descriptions the article assumed to be definite, i. e. 'D-'.

(2) $$NP ::= DET + (\{ADJ\}) + NOM + (RELCL)$$

The structure of a relative clause is characterized by (3):

(3) $$RELCL ::= (IPRN(\underset{\text{which}}{WELCH})) + \begin{pmatrix} \{ADJ\} \\ (ADV) + PRP + NP \end{pmatrix} + VRB \ .$$

Before going into detail concerning the further processing we shall illustrate the definition of noun phrases with some examples of noun phrases identified by partial parsing:

(4) DAS FERNSEHGERAET
the TV set

(5) DER BRAUNE, KAPUTTE STUHL
the brown, damaged chair

(6) DER RUNDE TISCH, WELCHER WEISS IST
the round table which is white

(7) DER BRAUNE STUHL, WELCHER DIREKT RECHTS NEBEN DEM TISCH STEHT
the brown chair which is standing directly to the right of the table

In German relative clauses are not marked with respect to restrictiveness (RITCHIE uses punctuation marks to discriminate restrictive relative clauses from non-restrictive ones). No syntactic features indicate the type of a relative clause. To interpret it as non-restrictive would entail adding new information to the knowledge base and proving its consistency with the knowledge already present. This is not done in HAP-RPM at present. Relative clauses, like attributive adjectives, are always regarded as being restrictive.

While relative clauses are identified according to the syntactic structure of (3) by a process similar to that used by GERSHMAN 1977 they are transformed into an internal representation associated with the noun phrase they refer to. Subsequently the relative clause is eliminated from the shallow structure.

If the relative clause introduces attributive adjectives, as in example (6), these adjectives are added to the attributes of the dominating noun phrase. Under certain conditions the system comments on this transformation by 'thinking aloud'. (see 2.8.5).

The second syntactic type of a relative clause (example (7)) essentially consists of a prepositional phrase which may be modified by an adverb. This prepositional phrase is bound to a name of the dominating noun phrase (a quoted description, which is a compressed version of its words) and is analysed later on in connection with that noun phrase. At this stage a relative clause which was part of the first noun phrase has disappeared from the shallow structure.

After this syntactically guided identification process the noun phrase is now subjected to semantic and pragmatic analysis.

In German a proper name may be preceeded by an article. Thus if the noun of the NP is already a proper name such as 'Marilyn Monroe' (who, incidentally, is represented in the pictures of WORLD3) the noun phrase in the shallow structure is replaced by it and in the remaining shallow structure further NPs are looked for. The same applies to noun phrases of the structure (DET NOM) having only one possible referent in the domain of discourse. Thus example (4) would be replaced by FERNSEHGERAET1 (TV set1).

The system does not attempt to establish sets of potential referents for nouns representing property dimensions in the conceptual semantic network (linked to concepts via the E-arc), such as 'D- FARBE' (the colour).

2.8.3 Attributive adjectives

Attributive adjectives, as in example (5), are analysed using the knowledge stored in the semantic networks. There are three ways in which properties of concepts or of objects are represented: denotative properties belong to the conceptual semantic network (D-arc, see 2.1.3.7), decomposed properties are defined by specific procedural knowledge (see 2.12.2) and referential properties are stated in the referential semantic network (REF-arc, see 2.1.3.8).

In the analysis of noun phrases each attributive adjective is examined to see whether it is a denotative property of the concept, i.e. the noun of the NP, or one of its supersets. All of HAM-RPM's deductive mechanisms (see 2.12) are employed in this process. If this is the case the denotative attribute is eliminated from the set of adjectives preceding the noun and the next adjective is inspected. Thus the system itself will never mention denotative attributes, which actually are inherent properties of concept classes, as for instance 'fluid water', in asking for additional information (see 2.8.7).

If, however, a denotative property is not applicable to the concept the whole utterance containing the actual noun phrase is rejected, as illustrated in the following example:

(8) IST DER FLUESSIGE STUHL ROT?
Is the fluid chair red?
:
EIN STUHL IST GRUNDSAETZLICH NICHT FLUESSIG!
A chair is by definition not fluid!

After the system has looked for properties of concepts and analysed them, the set of relevant tokens is created. All tokens linked to the noun of the NP by ISA-arcs – whether directly or indirectly via inheritance – are contained in the set of tokens. For example, the set of tokens of the noun MOEBEL (pieces of furniture) will be:

(9) (SESSEL1 SESSEL2 TISCH1 TISCH2 STUHL1 STUHL2 STUHL3 STUHL4 KLAVIER1 KLAVIERSTUHL1)
(arm-chair1-2, table1-2, chair1-4, piano1, piano stool1)

This set of tokens is the basis for determining whether one of its elements is the referent of the given definite description.
As in the lexical-decomposition component (see 2.11.2) attributive adjectives may be reduced to semantic primitives by decomposition procedures. The lexicon entry of these adjectives contains the keyword ATT-DEKOMPOSITION linked to the name of a procedure. There is, for instance, a procedure for the adjective LEER (empty), if it is used attributively, which evokes processes similar to the one described in section 2.12.3 for predicative use. The decomposition procedure yields a set of tokens which will consist of fewer elements than initially if the decomposed attribute only applies to a subset-of the initial set of tokens.

All the remaining adjectives of the NP under investigation are assumed to be referential properties. Going through the set of tokens, each single token whose referential properties do not fit the adjectives of the NP is removed from the set. With definite descriptions an exhaustive search has to be performed, as GROSZ 1976a,b has already stressed. During the checking of referential properties all of the deductive mechanisms of the inference component, such as the handling of relative adjectives (see 2.12.3; cf. RITCHIE *1977)* are employed.

The analysis of attributive adjectives finally yields a set of tokens. This set may be empty, which means that there is no appropriate object in the domain of discourse:

(10) DAS KAPUTTE KLAVIER
The damaged piano
⋮
ES GIBT HIER KEIN KAPUTTES KLAVIER.
There is no damaged piano here.

Here, as in example (8), the analysis of the whole utterance is rejected.

Having successfully analysed the attributes of an NP the initial set of tokens (9) has been reduced to a single element:

(11) NP: DAS BRAUNE, KAPUTTE MOEBEL
The brown, damaged piece of furniture
set of tokens: (STUHL2) .

2.8.4 Relative clauses

If the set of tokens consists of one or more elements, a previously transformed relative clause might help to confirm the analysis if a unique token has already been found or it might disambiguate the set if more than one token has remained so far. The type of relative clause whose analysis has not yet been described consists of a – possibly modified – prepositional phrase (see 2.8.2). The noun phrase which is part of this prepositional phrase is analysed in the same way as was described in the previous section. If there are no rejections, for example an existential presupposition is violated, a set of tokens is determined to represent the extension of the NP of the relative clause. The analysis so far has yielded two sets of tokens. The preposition of the relative clause specifies a relation between elements of these two sets.

An example will illustrate this:

(12) DER BRAUNE STUHL, WELCHER VOR DEM TISCH STEHT
The brown chair which is standing in front of the table

The set of tokens of DER BRAUNE STUHL is determined to be (STUHL1 STUHL2 STUHL3 STUHL4); the set of tokens of DER TISCH turns out to be (TISCH1 TISCH2). Some element of the STUHL-tokens is said to be in the relation VOR (in front of) to an element of the TISCH-tokens. At present prepositions in HAM-RPM have only locative meaning. Therefore the procedure used for checking spatial relations (see 2.13.1) is called to yield those pairs of tokens which stand in the proper relation to one another.

An adverb possibly modifying the scope of the relation determines the range of the domain of discourse which has to be inspected. At the moment four ranges are discriminated, each of which may be referred to by adverbs interpreted as linguistic hedges (cf. LAKOFF *1975).*

(13)

linguistic hedge (adverb)		range
DIREKT, GENAU, DICHT, UNMITTELBAR directly, precisely, closely, immediately	→	very narrow
BEINAHE, MEHR ODER WENIGER almost, more or less	→	rather narrow
IN ETWA, UNGEFAEHR about, roughly speaking	→	very broad
no adverb present	→	medium

Note that some of these adverbs are used similarly for the generation of definite descriptions of objects (see 2.15).

2.8.5 Thinking aloud

As the checking of spatial relations usually takes some time the system tells the partner what it is doing at the moment and – more important – that it is doing something. This strategy of 'thinking aloud' is often observed in natural dialogues. Longer breaks, which are not understandable for the partner, violate a communicative convention. In a system which is supposed to behave cooperatively, such pauses should be avoided.

Hendrix' LIFER system (HENDRIX *1977)* holds the attention of the user by printing a protocol – not in natural language – of the processing of his query. In HAM-RPM, messages not in natural language are permitted only in the trace mode (see 2.2.1). Its thinking aloud, like all other utterances intended for the dialogue partner, is in natural language.

A number of different syntactic frames are provided which are filled with actual words characterizing the current processes. The degree of detail of the utterance chosen is to represent the degree of ambiguity in the sets of tokens and thus gives an impression of the time presumably needed for the checking of spatial relations. Example (14) is an utterance when only two tokens are to be inspected, example (15) an utterance for highly ambiguous sets of tokens.

(14) DIE LAMPE, WELCHE SICH UNGEFAEHR HINTER DER SCHRANKWAND BEFINDET
The lamp which is situated roughly behind the cupboard
. . . UNGEFAEHR, HINTER DER SCHRANKWAND . . .EINE LAMPE . . .
. . . roughly behind the cupboard . . . a lamp . . .

(15) DER STUHL, WELCHER HINTER DEM BRAUNEN MOEBEL IST
The chair which is behind the brown piece of furniture
. . . ES SOLL HIER HINTER EINEM BRAUNEN MOEBEL EINEN STUHL GEBEN . . . DA MUSS ICH MAL GENAUER HINSEHN . . .
. . . behind a brown piece of furniture there's supposed to be a chair . . . I've got to take a closer look . . .

A different step in the analysis which produces thinking-aloud utterances is the transformation of relative clauses containing only adjectives (see 2.8.2). Out of a set of several syntactic frames one or perhaps none are selected at random as shown in example (16).

(16) DAS FERNSEHGERAET, WELCHES WEISS IST
The TV set which is white
. . . HMM . . . EIN WEISSES FERNSEHGERAET . . .
. . . hmm . . a white TV set . . .

2.8.6 Potential referents

Let's now return to the process of disambiguating sets of tokens by means of relations between them. The checking of spatial relations has yielded either nothing or a set of pairs of tokens. In the first case the relation specified could not be found between any pair of tokens in the domain of discourse. The whole utterance is therefore rejected:

(17) DAS BILD, WELCHES HINTER DEM BRAUNEN TISCH HAENGT
The picture which is hanging behind the brown table
HINTER DEM BRAUNEN TISCH SEHE ICH KEIN BILD.
Behind the brown table I do not see a picture.

If only one pair has been found, the complex definite description under inspection is successfully disambiguated and is replaced by the appropriate token. Thus a unique pair has been found in the domain of discourse, although each of the NPs, the one from the relative clause and the one from the dominating proposition may have been highly ambiguous. In the example above, there may have been several pictures and several brown tables, but only one picture behind a brown table.

If there are more than one pair of tokens then even the relation has not been precise enough to identify a single object. In these cases the record of references to objects (see 2.14.1) is inspected to see whether one of the tokens in question has recently been in the focus of attention (cf. GROSZ 1976b). If a suitable token is found, this is taken to be the one meant by the natural partner. If this still does not single out a unique pair, a clarification dialogue is initiated (see 2.8.7) to get more details about the object described by the NP of the relative clause.

The same processes as described for disambiguating the definite NP of the relative clause are carried out for the NP of the main proposition. The record of references to objects is inspected and clarification dialogues are initiated in case of ambiguity.

Finally, if no rejections have terminated the further processing, the definite NP is replaced by the token it referred to. A corresponding entry is added immediately to the record of references to objects; this may be required for pronoun resolution in a later part of the sentence (see 2.6).

The rest of the utterance is then searched for further noun phrases, which will be subjected to the same process of reference analysis. The output of the NP-resolution is the shallow structure with each NP replaced by a token. As NP-resolution always analyses the whole shallow structure and yields a modified one, it may be employed in different steps of the analysis; this actually is the case in HAM-RPM (see 2.1.2). An input to HAM-RPM consisting only of one noun phrase - as do the examples in this section - is also analysed until, if it is not interpreted as an elliptical utterance, the parsing component rejects it as an illformed sentence (see 2.9.2). Further commented examples of NP-resolution are contained in Memo No. 5.

2.8.7 Clarification dialogues

Whenever the reference of a noun phrase cannot be disambiguated the system does not give up and do nothing (WINOGRAD 1972) but rather asks the natural partner for further discriminating features. Such clarification dialogues are frequently used in natural dialogue situations, when the referential assumptions of the two partners are not identical (cf. DEUTSCH/CLAUSING 1979).

Two types of clarification dialogues are provided in the NP-resolution of HAM-RPM. If the number of potential referents is small, i.e. the set of tokens consists of only two elements, the system takes one of them and asks whether this one was the object meant. As the system on principle uses natural language to interact with the dialogue partner, it generates a noun phrase representing the internal object name (see 2.15). The natural partner may now accept or reject the system's choice. If he rejects it, the remaining token is regarded as the referent of the noun phrase and the partner is informed of this decision.

The second type of clarification dialogue is initiated if there are more than two potential referents. The system asks the partner to supply more distinctive features. The features already analysed are mentioned in the system's question. The response to such a question may be a more specific noun phrase, an elliptical one, such as DEN ROTEN (the red one), or a relative clause. Embedding the new information into the already analysed restrictions for the noun the system attempts to find a unique referent in the way described above. Recursive reference (a brilliant example is given in MORTON 1976) is thus modelled by clarification dialogues.

A complex noun phrase with a relative clause may contain two nouns, which may happen to be identical, as in DER SCHRANK, WELCHER NEBEN DEM SCHRANK STEHT (the cupboard which is standing next to the cupboard). In clarification dialogues the partner is informed which of the nouns could not be resolved uniquely. If it is the noun in the relative clause which is ambiguous, the whole prepositional phrase is asked for.

Clarification questions posed by the system are explained to the natural partner if he requests this by asking a 'why'-question (cf. Report No. 9). The system justifies its speech act by naming the quantity of potential referents established so far, as illustrated in (18) below.

Many features of the clarification dialogues appear in the example (18). The user's question is the first one of a dialogue sequence, so the record of references to objects is empty.

The answer to the original question is not only 'yes'; rather the transformed shallow structure, which was the input of the parsing, is added to the reply. This is done because the extensive clarification dialogue may have made the partner forget the original question. For another example of this kind see section 2.16.

(18)

KANNST DU DAS BRAUNE DING DA ANFASSEN?
Can you touch the brown thing there?

WELCHES BRAUNE DING MEINST DU? Which brown thing do you mean?	WIESO FRAGST DU MICH DAS? Why do you ask?

WEIL ES SIEBEN DAVON GIBT!
Because there are seven of them!

WELCHES DING MEINTEST DU? Which thing did you mean?	DAS, WELCHES NEBEN DEM TISCH STEHT The one standing next to the table

. . . MOMENT, EIN BRAUNES DING ALSO UND ZWAR NEBEN EINEM TISCH . . .
. . . just a moment, a brown thing . . . next to a table . . .

MEINST DU NEBEN DEM COUCHTISCH? Do you mean next to the couch table?	WESHALB FRAGST DU MICH SCHON WIEDER? Why do you ask me again?

WEIL ES ZWEI GIBT!
Because there are two!

MEINTEST DU NEBEN DEM COUCHTISCH? NEE!
Did you mean next to the couch table? No!

NEBEN DEM ESSTISCH ALSO.
Next to the dining table then.

MEINST DU DEN KAPUTTEN STUHL? GENAU.
Do you mean the damaged chair? Exactly.

JA, ICH KANN DEN KAPUTTEN STUHL GREIFEN.
Yes, I can touch the damaged chair.

2.8.8 Indefinite descriptions

A noun phrase is interpreted as an indefinite description if the indefinite article 'E-' ('a') was used and a generic interpretation is not possible (see 2.8.9). The analysis of these noun phrases is similar to the one described for definite NPs. At present there are only two differences.

If the modality of an input is 'potential' (see 2.3), violated decomposed or referential properties of objects do not lead to the rejection of the whole utterance. Properties known to be represented in referential knowledge are also known to be subject to change. In questions of the modality 'potential' the referential attributes of indefinite noun phrases do not have to agree with the current state of the domain of discourse, but are rather interpreted as questions concerning possible domains. In the example below there is no lilac picture to be seen in the domain.

(19) KOENNTEST DU EIN LILA BILD AUF DEN BODEN LEGEN?
Could you place a lilac picture onto the floor?
JA.
yes.

The second difference between the handling of indefinite and definite noun phrases concerns the postulate of uniqueness. Indefinite NPs do not have to refer to one specific referent but rather can refer to any member of the set of tokens. Therefore in HAM-RPM one of the appropriate tokens is selected and replaces the indefinite NP. This solution clearly does not in all cases do justice to the meaning of an indefinite noun phrase, as demonstrated in the following example.

(20) STEHT EIN STUHL NEBEN DEM ESSTISCH?
Is there a chair standing next to the dining table?
NEIN.
No.

The reason for the wrong answer 'no' is that just one token has been picked from the set of chairs and this one does not happen to be next to the table. It would be a somewhat more adequate method to generate the set of tokens of the indefinite NP and proceed to analyse the question. This is planned as an improvement of the NP-resolution component. Problems arising with indefinite noun phrases are discussed in more detail in Memo No. 2.

2.8.9 Generic descriptions

Generic or intensional descriptions (NORMAN/RUMELHART 1975) do not refer to objects but to concepts or classes of concepts. Generic noun phrases are syntactically similar to definite or indefinite ones, i.e. they may have either the definite or indefinite article.

In HAM-RPM criteria are being developed to distinguish generic descriptions from definite and indefinite ones. The presently existing criteria are only applied within a limited context, namely the analysis by sentence patterns (see 2.7), which handles simple types of 'which'- and 'how many'-questions. For experimental purposes just this part of the understanding capabilities has been chosen, because here special search and inference procedures are closely connected with the syntactic structure.

At present five rules indicate generic interpretation of noun phrases. An NP is regarded as being generic, if

- the noun is neither a proper name nor an internal name of an object,
- there are no attributive adjectives,
- there are no restrictive relative clauses attached to the noun,
- the definite article is present and no potential token of the concept represented by the noun is in the focus of attention (see 2.8.6),
- the question does not specify any local relations.

If all these criteria are satisfied by an NP of a 'which'- or 'how many'-question, it is regarded as a generic NP and replaced by the noun, which for further processing represents the name of a concept. In trace mode the user is informed of the generic interpretation of the noun phrase, as illustrated in this example:

(21) WIEVIELE RAEDER HAT DAS AUTO?
How many wheels has the car?
⋮
(DAS AUTO WIRD GENERISCH INTERPRETIERT)
(The car is being interpreted generically)
⋮
MEISTENS VIER.
In most cases four.

(22) WIEVIELE RAEDER HAT DER GELBE KOMBI?
How many wheels has the yellow station wagon?
⋮
VIER.
Four.

Obviously the criteria for identifying generic noun phrases are not adequate in every respect. More powerful methods would presumably include the use of verb frames and tenses, the postponement of the decision until the sentence has been parsed and, as a last resort, asking the natural partner directly about the intended type of reference. Together with the analysis of indefinite noun phrases, the handling of generic ones will be a major topic in the improvement of HAM-RPM's NP-resolution component.

2.8.10 Extensions of the NP-resolution

Apart from topics already mentioned in the preceeding sections future research on NP-resolution in HAM-RPM will mainly be concentrated on five points.

First, the syntactic structure of a noun phrase has to be extended to provide for genitive attributes, transformed relative clauses such as 'the chair standing to the left of the table', modified attributes such as 'the slightly damaged cupboard', and negative characterizations (which are already used in NP-generation; see 2.15).

Second, plural and conjoined noun phrases have to be analysed in respect to their referential function (cf. GERSHMAN 1977).

Third, nonrestrictive relative clauses and attributes have to be identified and processed as new information (cf. RITCHIE 1977).

Fourth, 'foregrounding' (see GROSZ 1976b, NORMAN/RUMELHART 1975) has to be employed in solving reference relations in tightly coherent dialogue sequences.

Fifth, any change of topic has to be noticed and have an impact on the resolution of noun phrases (cf. GROSZ 1976a).

2.9 SYNTACTIC ANALYSIS

The syntactic-analysis component in HAM-RPM dates from the beginning of our research on the simulation of a natural dialogue partner in 1975 and has not undergone major changes since that time. It was originally implemented to yield the phrase structure of simple sentences without spending much effort on solving more sophisticated syntactical problems such as structural ambiguity.

Syntactic sentence analysis consists of two successive processes: word-order transformations create a canonical shallow structure which is the input to the actual parsing of the sentence.

2.9.1 Word-order transformations

The objectives of HAM-RPM do not restrict the natural partner to questions (see 2.3). Therefore the component responsible for syntactic sentence analysis has to handle various types of input.

To limit the number of necessary syntax rules for all sentence types the shallow structure which has been generated and modified so far is transformed into a canonical structure. This canonical shallow structure resembles the structure of affirmative propositions (a similar approach was taken by BORGIDA 1975).

The transformation rules are formulated as is usual in a transformational grammar. A structural description (SD) of a preterminal string is associated with instructions on how to rearrange the string (SC: structural change). An example is given in (1).

(1) Input: WO STEHT DER BRAUNE STUHL?
Where is the brown chair standing?

:

shallow structure: ((IPRP (WO)) (VRB (STEH)) (NOM (STUHL1)))

SD: IPRP VRB NOM

SC: (3 2 1)

transformed shallow structure: ((NOM (STUHL1)) (VRB (STEH)) (IPRP (WO)))

The internal representations of word-order rules in HAM-RPM are stored on the property list of the identifier TRANS. The structural change of a shallow structure is stored as the value of a property which is a compressed version of the structural description.

(2)

identifier	value	property	
TRANS	(3 2 1)	IPRP VRB NOM	(actually: IPRPVRBNOM)
	(2 1 3 4)	VRB NOM PRP NOM	
	:	:	
	(2 1 3)	VRB NOM ADJ	

This internal representation is generated while the word-order rules are being read from the file TRANS.RUL (see 2.1.3.3).

On the one hand this method enables a user to add new rules or modify existing ones without detailed knowledge about the system's internal design. On the other hand the whole set of rules does not have to be searched for a suitable one but rather an applicable rule is accessed directly.

When HAM-RPM is operating in trace mode, a message is printed whenever a word-order rule is applied to the shallow structure showing its effect. For example, suppose the input sentence KANNST DU MIR SAGEN, OB DER TISCH VOR DEM STUHL STEHT? (Could you tell me whether the table is standing in front of the chair?) has been reduced to the shallow structure in (3):

(3) ((NOM (TISCH1)) (PRP (VOR)) (NOM (STUHL2)) (VRB (STEH)))
(PRAETRANSFORMATION:)
((NOM (TISCH1)) (VRB (STEH)) (PRP (VOR)) (NOM (STUHL2)))
⋮

If no word-order rule for a given shallow structure exists a corresponding message is typed: (KEINE PRAETRANSFORMATION MOEGLICH) (No word-order transformation possible). Examples of this type are questions syntactically structured as assertions, which in speech would be marked by intonation:

(4) DER BRAUNE STUHL IST BEQUEM?
The brown chair is comfortable?
⋮
((NOM (STUHL1)) (VRB (SEIN)) ((ADJ (BEQUEM)))
(KEINE PRAETRANSFORMATION MOEGLICH)
⋮

When they leave this component, the shallow structures of questions and commands have been unified to the structure of affirmative propositions, which will be parsed in the next processing step.

2.9.2 Sentence parsing

Syntax rules are stored internally in a way similar to that just described for word-order rules. A constant identifier called SYNTAX has one property for each syntax rule. Each property is a compressed version of the dominated categories; its value is the dominating category together with the dominated categories put in brackets.

(5)	<u>identifier</u>	<u>value</u>	<u>property</u>
	SYNTAX	(S (NP VP))	NP VP (actually: NPVP)
		:	:
		(PP (PRP DET NOM))	PRP DET NOM

The internal representation is constructed while the syntax rules are being read from the file BASIS.RUL (see 2.1.3.4). The same advantages as stated for the internal representation of word-order rules (see 2.9.1) apply to syntax rules.

As shown in the examples above, the rules are not restricted to a normal form, i.e. the quantity of dominated categories is not limited to a fixed number. Thus no difficulties arise with uninterpretable auxiliary categories, which become necessary if e.g. Chomsky normal-form is used. (The most extreme rule utilizing the nonstandard form would be the domination of a whole preterminal string by the category 'S', which obviously is inadequate.)

The maximum number of dominated categories appearing in a set of syntax rules is determined when the internal representation has been set up. This numerical value is used by the parsing process.

Having described the syntax rules let's now turn to the process which makes use of them.

The syntactic analysis of a sentence proceeds from left to right in the preterminal string (i.e. parts of speech in the beginning). The concatenation of as many preterminal symbols as the maximum of dominated categories indicates is constructed to see whether there exists a suitable syntax rule. If there is none, the length of the preterminal string just inspected is reduced by one element at a time until either the preterminal string is empty or a syntax rule can be applied.

If no applicable rule is found in the first attempt, the first element of the preterminal string is ignored temporarily and the remaining part is handled by the same procedure as described above. For the sake of simplicity we are calling the string to be analysed 'preterminal string', although this is not quite accurate. In the course of the analysis the relevant string consists not only of parts of speech, but also of dominating categories such as 'NP'.

If, however, a rule can be applied, the dominated categories in the preterminal string are replaced by the value of the syntax rule. Thus the categories replaced do not disappear, but are pushed down one level. The analysis which follows does not take notice of them, for it always regards the first level of the preterminal string as the object to be analysed.

Having reached the end of the preterminal string the analysis starts again at the left margin of the whole string which has been modified by the application of syntax rules.

The analysis terminates if the first level of the complete preterminal string consists only of the symbol 'S'. In this case the shallow structure of the input sentence has been syntactically accepted. In trace mode the system types the message (SATZ AKZEPTIERT:) (Sentence accepted:) and prints the structure which has been built up in the course of the parsing process. The parse tree is also printed – in a slightly distorted manner (see Fig . 7 below) – if the user requested at the beginning of the dialogue that this be done (see 2.2.1).

A shallow structure is syntactically not well formed if no further rule can be applied to a preterminal string which does not consist of the single symbol 'S'. In this case ellipsis recognition is tried (see 2.10) and if that also fails, the sentence is rejected by the system, which types BITTE FRAGE NOCHMAL ANDERS FORMULIEREN! (Please express the question differently!). In trace mode additional information is typed: (SATZ NICHT AKZEPTIERT:) (Sentence not accepted:) followed by the preterminal string constructed so far.

To illustrate the algorithm described above, the steps of the analysis are shown in Fig. 7 on the next page. Here the maximum number of dominated categories was three.

As the analysis does not keep track of which rules have already been applied it has no way to detect more than one structural description of a given shallow structure. But up to now this limitation has not appeared to be particularly disadvantageous. Nevertheless, together with a new concept of the semantic representation of a sentence (see 2.11), a more ambitious parser which can cope with structural ambiguities will be necessary for analyzing more complex sentences.

Experiments are to be made with the replacement of the syntactic sentence analysis in HAM-RPM by already implemented versions of an ATN parser (see BATES 1978). Parts of this parser might equally be employed in components with integrated partial parsing, such as NP-resolution (see 2.8). If the use of an ATN parser turns out to be too painstaking or too rigid for our purposes (see Report No. 9) the possibility still remains of improving the existing parsing component of HAM-RPM.

Natural partner's input:

WAS STEHT HINTER DEM GROSSEN TISCH?
What is standing behind the big table?

⋮

shallow structure:

((IPRN (WAS)) (VRB (STEH)) (PRP (HINTER)) (NOM (TISCH1))) (KEINE PRAETRANSFORMATION MOEGLICH)

applicable rule	inspected section of preterminal string
-	(IPRN VRB PRP)
-	(IPRN VRB)
NP → IPRN	(IPRN)
-	(VRB PRP NOM)
-	(VRB PRP)
-	(VRB)
PP → PRP + NOM	(PRP NOM)
-	(NP VRB PP)
-	(NP VRB)
-	(NP)
VP → VRB + PP	(VRB PP)
S → NP + VP	(NP VP)
	(S)

```
(SATZ AKZEPTIERT:)
(S (NP (IPRN (WAS))) (VP (VRB (STEH))  (PP (PRP (HINTER))
                                           (NOM (TISCH1)))))
S
        NP
                IPRN
                        WAS
        VP
                VRB
                        STEH
                PP
                        PRP
                                HINTER
                        NOM
                                TISCH1
```

Fig. 7

2.10 ELLIPSIS

HAM-RPM handles some kinds of elliptical expressions; this is of great importance in natural conversations, where utterances are by no means always complete. As for the resolution of anaphoric references (see 2.8.6), the dialogue context of an utterance provides a framework for the interpretation of sentences in which some elements are omitted.

In HAM-RPM ellipsis may occur on three levels: questions posed by the user, the user's responses to questions posed by the system, and the system's answers.

In most cases, the user's elliptical responses can be analysed straightforwardly, because the syntactic form of the system's question implies a certain syntactic form of an elliptical response. For example, the response to a question like

(1) WELCHEN STUHL MEINST DU?
Which chair do you mean?

will often be an isolated noun phrase, which might itself be elliptical. Elliptical responses of this sort are handled by the components of the system which generate the corresponding questions, as, for example, in a clarification dialogue (see 2.8.7).

Since HAM-RPM in general answers the user's question as concisely as possible – unless requested to do otherwise by the dialogue partner (see 2.14.2) – it usually outputs expressions which are not complete sentences, for example isolated noun phrases or prepositional phrases:

(2) WAS STEHT VOR DER AMPEL?
What's in front of the traffic light?

(3) DER ROTE PERSONENWAGEN.
The red car.

Information which was already contained in the question is not mentioned in the answer; indeed, it is not even contained in the semantic representation of the answer.

This section will focus on the processing of elliptical questions posed by the dialogue partner.

If the system's attempts to parse a question as a complete sentence fail, it backs up and starts an ellipsis-recognition process. Incidentally, a simple improvement of this strategy could provide faster recognition of relatively obviously elliptical inputs: If the system has noticed a tendency on the part of the human dialogue partner to use ellipsis, it first checks for ellipsis using certain simple criteria such as the absence of a verb; only if this fails does it attempt to parse the input as a complete sentence.

The ellipsis routines match the syntactic structure of the input, in which all pronouns and noun phrases have been resolved, against the shallow structure of the last complete input, which is stored in the value of the variable SHALLOW-ALT (shallow-old), as described in section 2.14. The following dialogue sequence represents a simple example of this approach:

(4) IST ER KAPUTT?
Is it damaged?

(5) NEIN.
No.

(6) DAS HEISST ALSO, HEIL?
Aha! Intact then?

(7) JA.
Yes.

(8) UND DER STUHL, WELCHER IN DER NAEHE DES FERNSEHGERAETES STEHT?
And the chair which is near the TV set?

From the shallow structure of (6), ((ADJ (INTACT))), the ellipsis routines construct the pattern

(9) (??BEGINNING (ADJ ?) ??END)

which is then matched against the shallow structure of (4),

(10) ((VRB (IS)) (NOM (TABLE1)) (ADJ (DAMAGED)))

Since the match is successful, the variable bindings are used to expand the shallow structure of (6) to

(11) ((VRB (IS)) (NOM (TABLE1)) (ADJ (INTACT))) .

An analogous process underlies the expansion of the shallow structure of (8), namely ((NOM (CHAIR4))), into

(12) ((VRB (IS)) (NOM (CHAIR4)) (ADJ (INTACT))) .

We see from this example that when the sentence preceding an elliptical input is itself elliptical, its expanded shallow structure is used. Incidentally, the conjunction 'and', which often appears at the beginning of elliptical questions, is deleted when the system tries to analyse the input as an ellipsis.

In contrast to the ellipsis facility designed by GROSZ (1976c) for the SRI Speech Understanding System, the routines sketched here can cope with sentences in which more than one constituent is present in the incomplete input. For example, consider the sequence:

(13) STEHT DIE FRAU JETZT VOR DEM FACHBEREICH FÜR INFORMATIK?
Is the woman now standing in front of the Computer Science Department?

(14) JA.
Yes.

(15) UND DER MANN RECHTS NEBEN DER AMPEL?
And the man to the right of the traffic light?

(16) NEIN.
No.

(17) DAHINTER?
Behind it?

When operating in trace mode, (see 2.2.1), HAM-RPM types the message ELLIPSE ERKANNT (ellipsis recognised) and the expansion of the elliptical input. In general, all decisions previously made concerning the type of question and the modal component (see 2.3) of the input are revised as soon as it is recognised as being elliptical.

It seems to be a certain improvement over the ellipsis facilities of LIFER (HENDRIX 1977) and PLANES (WALTZ 1978) that for the construction of the semantic representation of the expanded sentence full parsing and semantic translation are not performed; instead, on the basis of a comparison of the shallow structure of the previous sentence with the expanded ellipsis, particular constituents are projected onto the semantic representation of the previous sentence, which has been stored just as the previous shallow structure has (see 2.14.1). Consider the following dialogue sequence:

(18) PARKT DER WAGEN AUF DER PARKZONE?
Is the car parked in the parking zone?

(19) NEIN.
No.

(20) VOR DEM BAUM, WELCHER IN DER NAEHE DES GROSSEN GEBAEUDES STEHT?
In front of the tree which is near the large building?

To construct the semantic representation of (20), the predicate ON in the semantic representation of (18), namely

(21) ((CAR4) (PARKING/ ZONE1) (ON))

is replaced by IN/ FRONT/ OF, and the argument PARKING/ ZONE1 is replaced by TREE3. This produces

(22) ((CAR4) (TREE3) (IN/ FRONT/ OF)),

which is the semantic representation of the elliptical input (20).

One limitation of the current ellipsis routines stems from the assumption that the shallow structure of the elliptical utterance must be similar to some part of the last successfully analysed input. This is not always the case, as is shown by the following example:

(23) HAT DIE STRASSE ZEBRASTREIFEN?
Has the street got zebra stripes?
(24) JA.
Yes.
(25) WIEVIELE?
How many?

The major limitations of the current version of the algorithm are due to the fact that HAM-RPM's ellipsis routines use only syntactic clues. By contrast, the mechanisms of LIFER (HENDRIX 1977) and PLANES (WALTZ 1978) for handling ellipsis rely heavily on the notion of a 'semantic grammar'. This means that a certain amount of semantic information is encoded in the syntactic categories (see Report No. 11), so that similarity of 'syntactic' structures implies semantic similarity. We could easily introduce semantic categories into our grammar, and this would improve the performance of our ellipsis routines considerably. But we think that from a theoretical point of view such 'semantic grammars' are awkward because of their lack of generality and economy of description.

Although the simple but efficient approach described here works remarkably well, a notion of semantic closeness such as the one used by GROSZ (1976c) to select from among the different candidates for the substitution and semantic checks on the suitability of a projection are under exploration so that HAM-RPM's treatment of ellipsis may be improved.

2.11 GENERATION OF THE SEMANTIC REPRESENTATION

2.11.1 The translation component

The input to the search and inference component of HAM-RPM is a semantic representation of the user's question. The propositional part (see 2.3) of this semantic representation is constructed by a translation component which we will describe in this section. The source language of the translation component consists of syntax trees resulting from the parsing process described in section 2.9.2. The target language consists of semantic representation constructions which in general are ordered sets of triples. The semantics of this representation construction are defined in section 2.12.1. As described in section 2.10 the translation process is skipped when the input is an expanded ellipsis; instead a projection is performed, which uses the semantic representation of the previous sentence.

The general scheme of the translation process is first to match the parse tree against an ordered set of patterns and then to use the variable bindings resulting from these matches to construct the semantic representation.

As we will show below by examples, the whole matching process is controlled by the semantic definitions of the main verb of the analysed sentence. In the lexicon entry for each verb (see 2.1.3.1) a pointer is attached to the keyword SEMANTIK, which allows the translation component to retrieve the semantic definition of the verb. For example, consider the lexicon entry for the verb 'to drink':

(1) (TRINK (WORTART (VRB) SEMANTIK (TRINK-SEM))) .

Here TRINK-SEM is the name of a piece of knowledge which procedurally represents the semantics of 'to drink'.

The pattern matching is organized in the following way: For each position of the triple an ordered set of transformation rules is stored in the knowledge base of the system (see 2.1.3.5). Each transformation rule is an arbitrarily complex FUZZY pattern containing only one open variable ?A. A successful match causes the insertion of the value of the variable ?A into the corresponding slot of the triple. If all matches fail, NIL is inserted into the corresponding slot.

Let us examine this process in detail by discussing three simple examples. Suppose that after a successful parsing of sentence (2) the syntax tree displayed in Fig. 8 is to be transformed by the translation component.

(2) KANNST DU DAS GETRAENK, WELCHES SICH AUF DEM COUCHTISCH BEFINDET, TRINKEN?

Could you drink the drink which is located on the couch table?

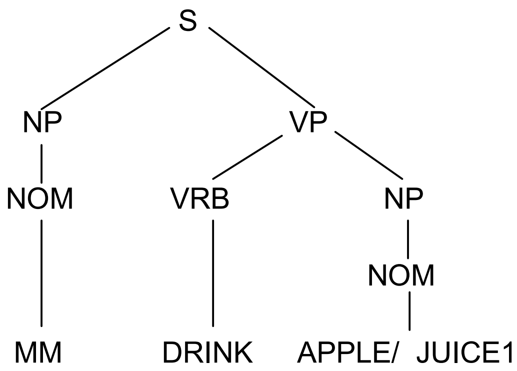


Fig. 8

First the pattern displayed in Fig. 9 is matched against the structure in Fig. 8. As a side effect of this successful match, the variable ?A is bound to the value MM. This value is inserted into the first slot of the first triple. Fig. 9 shows the graphic representation of the pattern, which in linearized form is represented as (3):

(3) (S (NP ?? (NOM ?A)) ??)

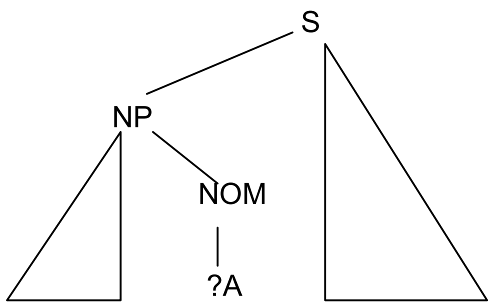


Fig. 9

Note, incidentally, that this is also the form in which the patterns are stored on the file SESY.PAT. This file can easily be updated to include further transformation rules if the scope of the grammar stored on the file BASIS.RUL (see 2.1.3.4) is expanded.

The second slot of the triple is filled with APPLE/ JUICE1. This is accomplished by unifying Fig. 8 with the pattern

(4) (S ?? (VP (?? (NP ?? (NOM ?A)) ??))) .

Having filled the third slot of the triple with DRINK the semantic definition of the verb 'to drink' causes the translation process to stop without producing further triples. In summary, the result obtained by the translation component for this simple case is the triple:

(5) ((MM) (APPLE/ JUICE1) (DRINK))

Let us consider another example, in which a NIL slot is generated. Suppose (7) is the linearized form of the parse tree for (6).

(6) WOVOR STEHT DAS FERNSEHGERAET ?
The TV set is located in front of what?

(7) (S (NP (NOM (TVSET1))) (VP (VRB (LOCAT)) (PP (PRP (IN/ FRONT/ OF)) (IPRN (WHAT)))))

The first slot of the triple is filled in the same way as in the previous example, this time with TVSET1. All patterns defined for the second slot fail to match. Thus the value NIL is inserted into this slot. This signals to the search and inference component of HAM-RPM that a value has to be found which fills the slot. The semantic definition of LOCAT causes a match of (7) with the pattern (8).

(8) (S ?? (VP ?? (PP (PRP ?A) ??) ??))

Since this is a successful match, the third slot is filled with IN/ FRONT/ OF resulting in the triple (9):

(9) ((TVSET1) NIL (IN/ FRONT/ OF)) .

To give an idea of the more general capabilities of the translation component let's finally consider a slightly more complicated example. The parse tree (11), which is derived from the surface structure (10), contains the verb PLACE, which in this case takes three arguments: You, TV set, couch table. Since the predicate and its three arguments obviously cannot be represented by one triple, successively the two triples in (12) are generated.

(10) KANNST DU DAS FERNSEHGERAET AUF DEN COUCHTISCH STELLEN?
Could you place the TV set on the couch table?

(11) (S (NP (NOM (MM))) (VP (VPA (VRB (PLACE)) (NP (NOM (TVSET1))) (PP (PRP (ON)) (NP (NOM (TABLE2))))))

(12) ((MM) (TVSET1) (PLACE))
((TVSET1) (TABLE2) (ON))

The order of the triples reflects a logical dependency: First it must be checked whether MM is able to manipulate the TV set at all and then whether he can place it on the couch table.

When operating in trace mode HAM-RPM types the triples generated by the translation component for inspection by the user.

This notation for the semantic representation of a sentence is inadequate for representing the meaning of very complex sentences. In fact it represents a sort of bottleneck within HAM-RPM as a whole: a number of important types of questions and some interesting inference processes have had their implementation delayed because the corresponding deep structure could not be represented.

Work is in progress on a richer notation for representing deep structures which uses the template construct provided by the language AIMDS (see 1.2.6). The meaning of a particular sentence, e.g. 'Where are you?' is represented by a specific instance of a general template representing a certain type of sentence, e.g. 'request for description of location'. Templates similarly represent other complex structures such as indefinite noun phrases. Each general sentence template contains slots which must be filled in order to represent the meaning of a certain type of sentence, as well as information on how to fill the slots using various kinds of knowledge.

Here are three strong points of this approach:

- Since the slots in a template can be filled with other templates, the meaning representation can be arbitrarily deeply nested.
- The important differences between various types of sentences can be done justice to by defining the corresponding templates accordingly, there being no need to squeeze all meaning representations into a single format.
- The generation of the meaning representation needn't proceed strictly bottom-up: once the sentence type has been determined, the construction of the semantic representation is guided by the corresponding general templates.

2.11.2 Lexical decomposition

The output of the translation component, which was outlined in the previous section, in general consists of an ordered set of triples. This set undergoes a process of lexical decomposition, which we will sketch in this section. The purpose of this component is to map the semantic representations constructed by the translation component onto 'deeper' representations, in which elements of a set of semantic primitives are used. This mapping is essential, since inferences are only defined for semantic primitives, whose formal properties are known to the system. The user can easily extend the set of semantic primitives, for instance by introducing new relations into the semantic networks or adding the appropriate inference rules. It is not our intention here to discuss the strengths and limitations of approaches which make heavy use of semantic primitives, such as the system of SCHANK 1975.

In the current version of HAM-RPM only adjectives are decomposed. Since these linguistic entities always occur in the third slot of a triple, the lexical-decomposition routine checks whether the knowledge base provides a decomposition procedure for the item in the third slot.

This is done by consulting the lexicon (see 2.1.3.1), where for each decomposable word a pointer to a decomposition routine is attached to the keyword DEKOMPOSITION. For example, this is the lexicon entry for LEER (empty):

(13) (LEER (WORTART (ADJ) DEKOMPOSITION (LEER-DEK))) .

The evaluation of LEER-DEK triggers transformations in the modal (see 2.3) and the propositional part of a semantic representation which includes the adjective 'empty'. Suppose that sentece (14) was mapped onto the triple (15).

(14) IST DAS GLAS LEER?
Is the glass empty?

(15) ((GLAS1) NIL (LEER))

When the lexical-decomposition procedure for 'empty' is executed, LEER in (15) is replaced by the semantic primitive CONTAINS and in the modal part of the semantic representation the value of the variable NEGATION is changed from T to NIL or vice versa. This means that actually HAM-RPM tests whether a container is empty by checking whether anything is contained in it. One advantage of lexical decomposition is that other surface predicates can also be decomposed using the predicate CONTAINS. For example, in HAM-RPM the adjective VOLL (filled) is in effect mapped onto 'includes something'.

When operating in trace mode the system types the message DEKOMPOSITION and the transformed triple. For example, after the decomposition of LEER in (14) the system types:

(16) (DEKOMPOSITION ((GLAS1) NIL (ENTHAELT)) NEGATION: T)
contains

In the current version of HAM-RPM little use is being made of the decomposition capabilities, so that most semantic representations constructed by the translation component are passed on to the answer-generation routines without any change.

2.12 THE INFERENCE COMPONENT

The task of the inference component is to build up a semantic representation of an appropriate response to a given question. The problem of how to formulate this response in natural language for presentation to the dialogue partner can then be handled by the components described in sections 2.15 and 2.16.

The inference component accepts as input a semantic representation of the user's question, which has been generated by the translation component (see 2.11.1) and possibly modified by the decomposition routine (see 2.11.2). It is capable of retrieving information explicitly stored in the referential and the conceptual semantic networks and of deriving information not explicitly represented in the knowledge base by invoking inference rules coded as FUZZY DEDUCE-procedures (see 2.1.3.6).

It is clear that even during the language-understanding process which has been described up to this point the system must be able to make inferences concerning the content of the dialogue partner's utterances, for example when determining the referent of a noun phrase (see 2.8). Even the language-generation components must perform inferences. In other words, the derivational and retrieval machinery contained in HAM-RPM is employed in almost every processing phase.

But what we shall describe in this section is only the inference processes involved in the derivation of the semantic content of an appropriate answer to a given question. Although both read-time and question-time inferences, as defined by CHARNIAK (1976), are used, we shall deal in this section only with question-time inferences.

Before describing the various types of inference processes used in HAM-RPM, we shall show how the semantic representations of the input sentences are interpreted by the inference component.

2.12.1 The semantics of the semantic representations

Apart from the modal part of the semantic representation of an input sentence, the triples generated by the translation component are interpreted as logical expressions of the Predicate Calculus. Since a list of triples is treated by the inference component as an abbreviation for an extended conjunction of logical expressions, we shall confine our discussion to the semantics of single triples.

All NIL slots of a triple are interpreted as distinct variables. A NIL in the third slot of a triple is a variable ranging over the set of all primitive predicates known to the system. Variables in the first two slots range over individual constants of the domain of discourse. The third slot is thus the predicate position, the first two slots the argument positions. The second slot is disregarded by the inference component when the third slot contains a one-place predicate, as is the case with the triple ((CHAIR4) NIL (RED)).

When the triple is fully instantiated, as is ((TV/ SET1) (TABLE2) (ON)), the truth of the predication is tested against the domain model encoded in the knowledge base.
The interpretation of the variables contained in the triples depends on whether the triple is interpreted as representing a yes/no-question or a wh-question (see 2.3). For yes/no-questions, all of the variables are interpreted as being existentially quantified with immediate scope. In the case of the triple ((GLASS1) NIL (CONTAINS)), for example (see 2.11.2), the truth value of

(1) $(\exists x)$ CONTAINS (GLASS1,x)

is checked. For wh-questions, on the other hand, all variables are taken to be universally quantified. For example, for the triple ((TV/ SET1) NIL (IN/ FRONT)), the set of all x such that

(2) IN/ FRONT (TV/ SET1,x)

is true represents the complete answer to the question. (In fact, only a particularly relevant subset of this set is found and mentioned in the answer; see 2.13.4.)

Variables can range over predicates, as well as over tokens, as in the example (TV/ SET1) NIL NIL), which is the semantic representation of the question

(3) WO STEHT DAS FERNSEHGERAET?
Where is the TV set?

This sketch of the semantics of triples has been simplified in many respects. For example, when 'potentiality' has been recorded as the modal part of a yes/no-question, the system checks whether the formula denoted by the triple is true in some possible world. An example of such a sentence is

(4) KOENNTEST DU DAS GETRAENK TRINKEN?
Could you drink the drink?

whose semantic representation is ((MM) (APPLE/ JUICE1) (DRINK)).

An important feature of our approach, which distinguishes it from most others, is that it is based on fuzzy set theory and fuzzy logic. This means that the predicates in the third slot are interpreted as fuzzy predicates, that the truth values computed are fuzzy truth values, and that the sets derived by the inference component are fuzzy sets. For the sake of clarity we have disregarded these facts in the informal presentation of the semantics just given. Thorough discussions of this topic are contained in WAHLSTER 1977 and HAM-RPM Report No. 5.

2.12.2 The general structure of the inference process in HAM-RPM

The first task of the inference component is to determine which inputs are requests involving spatial relations and which are requests for network search and deduction. In this section only the latter case is discussed, the treatment of spatial relations being described independently in section 2.13.

Before actually performing any inferences, the system checks for possible selectional restrictions on the predicates contained in the triple. For example, consider the triple ((MM) (CHAIR4) (DRINK)). The semantic definition of TRINKEN (drink) contained in the knowledge base specifies as a selectional restriction that the object of drinking must be a fluid. When a selectional restriction is violated, the input sentence is rejected and the reasons for the rejection are recorded, so that any follow-up-questions posed by the user asking for a justification of the rejection can be answered (see 2.5.1). For the present example, the system generates the justification:

(5) EIN STUHL IST NICHT FLUESSIG.
A chair isn't fluid.

Note that the checking of selectional restrictions may itself require complex inference processes.

Ultimately any question is transformed into one or a series of GOAL-expressions, which trigger search and inference processes. All triples containing a one-place predicate for which no specific semantic definition is stored under the property SEMANTIK are transformed into GOAL-expressions which inspect REP-arcs (see 2.1.3.8). For example the triple

(6a) ((CHAIR3) NIL (BROWN))

is transformed into

(6b) (GOAL (REP CHAIR3 BROWN)).

Suppose the assertion used as an argument for the GOAL-expression is present in the referential network. Then we have the trivial case, since the GOAL succeeds immediately when a matching assertion is found.

Now let's discuss a slightly more interesting example, in which an inference rule must be applied. Suppose in the inference component the expression

(7) (GOAL (REF PIANO1 EXPENSIVE))

is evaluated and the referential network contains only (ISA PIANO1 PIANO) as an assertion referring to PIANO1. Furthermore suppose that the assertions (U MUSICAL/ INSTRUMENT PIANO) and (D MUSICAL/ INSTRUMENT EXPENSIVE) are stored in the conceptual semantic network. Property inheritance, namely the transitivity of D-arcs over U-arcs, is defined via a DEDUCE-procedure with the calling pattern (D ?X ?Y). In the body of this rule, which is one of the general inference rules stored on the file INFER.PRC, the two conjoined expressions (GOAL (U ?Z !X)) and (GOAL (D !Z !Y)) are evaluated.

Since the evaluation of (7) failed HAM-RPM tries a generic interpretation of the piano in (7) evaluating:

(8) (GOAL (D &(*TYPE PIANO1) EXPENSIVE))

*TYPE is one of a large number of general service functions working on the networks and gives back the name of the superset of its argument using the ISA-arc in the referential semantic network. Since no matching assertion for (D PIANO EXPENSIVE) is found in the conceptual semantic network, inference rules are automatically fired by FUZZY's pattern-directed procedure invocation mechanism. In our example the variables ?X, ?Y and ?Z are assigned to PIANO, EXPENSIVE and MUSICAL/ INSTRUMENT respectively. Since the evaluations of (GOAL (U MUSICAL/ INSTRUMENT PIANO)) and (GOAL (D MUSICAL/ INSTRUMENT EXPENSIVE)) succeed, the GOAL in (8) succeeds. Actually, inferences in HAM-RPM are usually far more complicated than this, because in the average inference chain many different inference rules have to be applied and a good deal of backtracking must be done (see Report No. 5 for such examples).

Having sketched the general framework within which the inference component works, in this paper we only discuss in more detail the handling of fuzzy predicates, which is one of the distinguishing features of HAM-RPM's inference component.

2.12.3. Inferences based on fuzzy knowledge

The use of terms with vague meanings and of fuzzy inferences is a typical feature of non-technical communication. As the objective of our research is to reconstruct the communicative and cognitive competence of a human dialogue partner in a simple everyday situation, in the design of HAM-RPM we have considered the human ability to deduce useful information even in case of vague or partial knowledge by approximative and common-sense reasoning.

As a typical example of our approach to vagueness let us first consider the representation of the referential meaning of vague, relative adjectives and its use in the inference component of HAM-RPM. The use of relative adjectives like 'old', 'big', 'narrow' always implies a context-dependent comparison with an expected norm. Furthermore, such predicates in most cases fit only to a certain degree to a given object. Hence we represent the referential meaning of relative adjectives in HAM-RPM as a set of implicitly defined fuzzy sets (see ZADEH 1978) whose members correspond to the possible reference sets. As an example, consider the referential meaning of 'big' for the reference set 'trees' in (9):

```
(9a)   (PROC (REF _> TREE BIG)
(9b)            (FETCH (REF !TREE ?HEIGHTVALUE INST HEIGHT)
(9c)            (SUCCEED (REF !TREE BIG) (SFUNK !HEIGHTVALUE 10 20))
```

Suppose the assertion (10) is stored in the referential net, denoting the fact that the height of

```
(10)   ((REF TREE3 15 INST HEIGHT) . 0.8)
```

TREE3 is about 15 meters. Furthermore suppose in the inference component the GOAL-expression (GOAL (REF TREE3 BIG) 0.2) is evaluated. When no matching expression is found in the referential net, the DEDUCE procedure displayed above is invoked. _>TREE is an implicitly typed variable. This means that it can only be assigned to names of objects of type TREE; this insures that this definition of 'big' is used only for the appropriate reference set. With (9b) the system retrieves the height of TREE3. The degree of 'bigness' of TREE3 is computed in (9c) as a fuzzy set membership value via the standard function SFUNK introduced by Zadeh. In the second and third parameter of SFUNK the expected norm of 'bigness' is encoded. The value of the GOAL-statement above is then ((REF TREE3 BIG) . 0.5). Note that the GOAL-statement has as its second argument a threshold, so that assertions derived by the inference component don't succeed if their z-value is smaller than this threshold. This is important, because in the semantic networks negation is represented by the z-value 0.

Finally, let's consider a derivation in which both fuzzy and highly subjective knowledge are involved. Suppose that the assumptions (11) - (14) are represented in the semantic networks.

(11) ((ISA APPLE/ JUICE1 APPLE/ JUICE) . 1.0)
(12) ((U JUICE APPLE/ JUICE) . 1.0)
(13) ((D JUICE NON-ALCOHOLIC) . 1.0)
(14) ((D APPLE/ JUICE SWEET) . 0.6)

(14) represents the subjective experience that apple juice is often sweet, but sometimes sour. Suppose the question 'Do you like this apple juice?' is finally transformed into the expression (GOAL (REF APPLE/ JUICE1 TASTY) 0.2). The 'subjective meaning' of 'tasty' is defined by the inference rule displayed in fig. 10.

META-KNOWLEDGE:

Apply the control knowledge coded in TRACE-PROCEDURE-DEMON8.
Don't use instantiations of premises with degrees of belief lower than 0.3.
The degree of informativeness of this rule is 0.5.

RULE:

If you want to show that (X is a tasty drink)
show that (X is sweet)
and that (X is non-alcoholic).

Fig. 10

This inference rule is coded in FUZZY as follows:

```
(15)  (PROC DEMON: TRACE-PROCEDURE-DEMON8 ZVAL: '(0.3 0.8)
      (D _>DRINK TASTY)
      (FOR GOAL: (D !DRINK SWEET)
            (FOR GOAL: (D.!DRINK NON-ALCOHOLIC) (SUCCEED?))))
```

Meta-knowledge is associated with each inference rule which is available to the system. This meta-knowledge plays an essential role in the generation of explanations of inferences (see 2.5.1). It is also used by the inference component. For example, when the calling pattern of more than one inference rule matches the pattern in a GOAL-expression, the conflict is resolved by preferring the inference rule with a higher degree of informativeness.

Since in the given dialogue situation the system is unable to taste APPLE/ JUICE1, it derives an answer on the basis of general knowledge, evaluating the expression (GOAL (D APPLE/ JUICE TASTY)). Applying the inference rule (15) and the general rule for property inheritance described above, the system builds up the goal tree shown in Fig. 11.

The z-value 0.6 of the conclusion is verbalised by the output routine of the system by the hedged affirmative ICH GLAUBE JA (I think so). The goal tree and some additional information are recorded in a separate CONTEXT of the assertional data base called INFERENCE-MEMORY by TRACE-PROCEDURE-DEMON8, a procedure demon which differs from the one normally used in that it makes a note of each step in the inference process. The information in this CONTEXT in the basis for HAM-RPM's explanation capability (see 2.5.1).

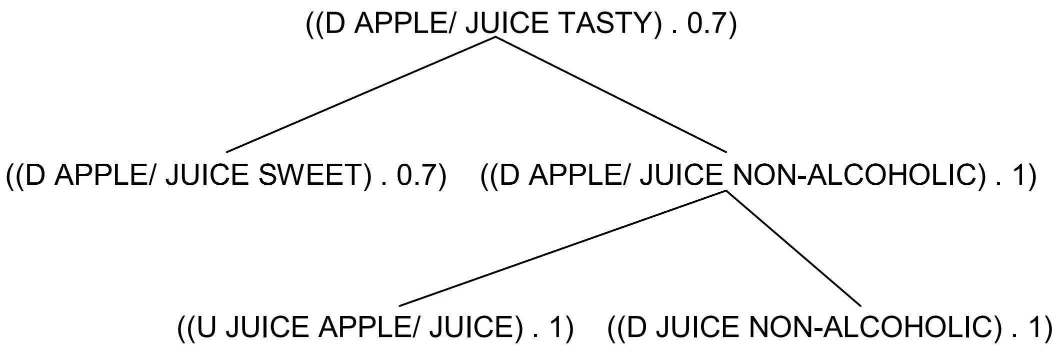


Fig. 11

2.13 SPATIAL RELATIONS AND THE SIMULATION OF VISUAL SEARCH

Spatial relations play an important role in conversations about HAM-RPM's scenes. On the one hand, some questions focus on spatial relations directly, for example

(1) WORAUF STEHT DAS FERNSEHGERAET?
What's the TV set standing on?

On the other hand, both the natural dialogue partner and the system itself often refer to objects in the scene in terms of spatial descriptions, using noun phrases such as

(2) DER SESSEL, WELCHER DIREKT HINTER DEM COUCHTISCH STEHT
the armchair that's directly behind the couch table

(see sections 2.8.4 and 2.15).

Two kinds of tasks which must be performed by the component of HAM-RPM responsible for spatial relations may be distinguished:

- the checking of spatial relations, i.e. the determination of the degree to which a given spatial relation holds between two given objects
- the search for objects which stand in a given spatial relation to a given object.

The former task, which will be discussed first, involves little more than the application of the definitions of the various spatial relations to particular objects.

2.13.1 The definitions of spatial relations

The locations of most of the objects in HAM-RPM's domains of discourse are represented by pairs of integers between 1 and 30 corresponding to points in the horizontal plane (see 2.1.3.9). This information permits an adequate description of the spatial relations between objects for HAM-RPM's purposes, except in the case of very large objects such as the streets in the traffic scene. Such objects could be represented by sets of coordinate pairs, but the definitions of the spatial relations involving such objects are complicated and have not yet been worked out within the context of HAM-RPM. The coordinates of each object, which are loaded from the file GEO.DTA at the beginning of each dialogue, are placed on the property list of the object's name to permit quick retrieval.

The spatial relations in the plane which are presently defined in HAM-RPM are: VOR (in front of), HINTER (behind), NEBEN (next to), RECHTS/ NEBEN (to the right of), LINKS/ NEBEN (to the left of), and IN/ DER/ NAEHE (near).

The procedure PRUEFEN (check) takes as arguments two object names and the name of a spatial relation. It checks to see whether the given relation holds between the two objects by first computing the vector which points from the first object to the second and then comparing this vector with the definition of the spatial relation. These definitions are essentially mappings from vectors onto z-values. Figure 12 illustrates the definition of VOR (in front of) by showing which z-values between 0 and 1 are associated with various vectors.

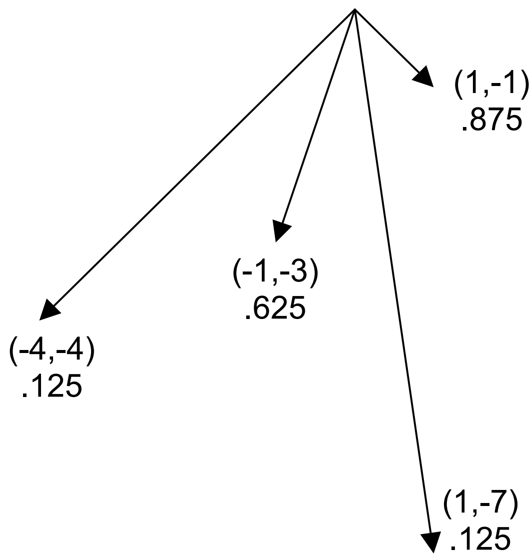


Fig. 12

The definitions presently being used were developed on the basis of intuition and with a view to simplicity; they could certainly be improved upon with respect to their psychological realism.

The procedure PRUEFEN fails if the vector it computes doesn't lie within the area for which z-values are provided by the definition. Otherwise, it returns essentially the z-value found.

The definitions of the individual relations are contained directly in the definition of PRUEFEN. It would have been more consistent with HAM-RPM' s overall design philosophy to store them separately so that they could be easily understood and modified by naive users. This wasn't done because such definitions would be difficult to express in a self-explanatory notation and because the number of relations of this sort is very limited in any case.

As described in section 2.1.3.9, spatial relations involving the vertical dimension are stored directly in the knowledge base. One reason for this is that the small number of vertical relations existing between pairs of objects in HAM-RPM's domains makes this simplification feasible. Another reason is that vertical relations tend to depend more heavily than plane relations on the shape of the two objects and the nature of the contact between then (compare, for example 'on' and 'above'), so that deriving them from coordinates would entail the addition of more detailed information on the position of the parts of the objects.

When the relation name passed to PRUEFEN as its third argument belongs to the vertical dimension, PRUEFEN first checks whether the vector between the two objects is (0 0). If so, it tries to FETCH the corresponding assertion - for example (SENKRECHTE AUF GLAS1 TISCH2) ('(horizontal on glass1 tablet2)') from the associative net which was initially loaded directly from the file GEO.NET. If it finds such an assertion, it returns its z-value; otherwise it fails.

2.13.2 The role of the simulation of visual search

In a number of different cases the system must search for objects standing in a given relation to a given object. One such case is when it is answering a question like (1) above. Another is when it is generating a definite description of an object in terms of the object's spatial relations to other objects (see 2.15).

In contrast to the checking of spatial relations described in the last section, the answer returned by such a search process, which is essentially a list of objects, is not determined solely by the position of the objects in the scene and the definition of the spatial relation in question. This is because there are usually more objects present which fit the given description than it would make sense for the system to mention. The system must therefore select certain objects over others on the basis of several criteria, two of which are:

- proximity to the given object
- salience of the object found.

The objects which rate high in these two respects are, generally speaking, the ones which would tend to be found first by a human being looking in the neighbourhood of the given object. The corresponding search processes in HAM-RPM have therefore been designed to simulate to a certain degree the way in which a human being performs such a search. (A more detailed discussion of the motivation underlying this simulation is contained in Report No. 3.)

2.13.3 The data structures used by the search processes

The key notion involved in this simulation is that of a 'glance': when looking at a certain point in a scene, a human being is aware of certain objects, but not others; some of the objects he can see are more salient than others, this salience depending not only upon the appearance of the object itself, but also on its distance from the point being looked at. Since the processing which actually takes place in a human being during a glance does not lend itself to simulation on a digital computer, the essential points just mentioned are reproduced in HAM-RPM by having the system rather laboriously compute the content of each possible glance in advance, at some time at which it isn't engaged in a dialogue.

The content of each possible 'glance' is stored in a separate FUZZY CONTEXT. There are as many CONTEXTs as there are points with integer coordinates in the scene in question, each CONTEXT bearing a name constructed from the coordinates of the corresponding point, e.g '/ 4/ 16'. An entry in such a CONTEXT has the form

(3) ((AUFFAELLIG <token>) . z-value)

where the z-value represents the 'relative' salience of the object, i.e. its salience with respect to a glance at the point corresponding to the CONTEXT in question, as opposed to its 'intrinsic' salience, stored on the file GEO.NET, which it has only in the CONTEXT corresponding to its own location.

To compute the contents of all of the possible glances, the system determines for each object in the scene the names of the CONTEXTs in which it should be represented and the relative salience it should have in each of these. This is done using a simple, intuitively defined function of the location of the object and its intrinsic salience. For example, most objects of low salience are represented only in the CONTEXT corresponding to their own location, in which they have their intrinsic salience, and in the CONTEXTs corresponding to the four points nearest to them, in which they have an even lower salience.

Once the contents of all of the CONTEXTs have been determined, all of this information is dumped onto the file BLICK.DMP (glance.dmp). This file is used during the dialogue instead of the CONTEXTs in the assertional data base themselves to save space in working memory. At the beginning of the dialogue BLICK.DMP is indexed for random access in roughly the same way as the lexicon file LEX.DTA (see 2.4.3). The FUZZY primitive FETCH can't be used to retrieve data from a file, but an analogous procedure is used. (For the sake of clarity of exposition we shall speak below as if the CONTEXTs were all present in working memory during the dialogue and as if they were accessed using FETCH.)

2.13.4 The search processes

The way in which the data structures just described are used can best be illustrated using the example of a question like

(4) WAS STEHT HINTER DEM JUNGEN MANN?
What's behind the young man?

in which the system is explicitly requested to describe the objects which stand in a certain relation to a given object. Let's suppose that the NP-resolution component (see 2.8) has determined that MANN4 is the referent of the noun phrase 'the young man' and that its location, retrieved from its property list, is (3 15).

There is a FUZZY procedure called WAS (what) which specialises in generating objects which fit such descriptions. Its arguments are essentially the name of an object and the name of a spatial relation. First, WAS calls the procedure BLICKE (glances), which is passed (3 15) and HINTER (behind) as arguments; BLICKE computes a list of 'glances', i.e. CONTEXT names, which correspond to points 'behind' (3 15) in the coordinate system. The *CONTEXT* names returned by BLICKE do not correspond to all of the points which satisfy the definition of HINTER, but rather to a subset of them large enough to ensure that all but the least salient objects will be contained in at least one of the CONTEXTs. At present about every fourth CONTEXT is sampled. (It would be possible to implement BLICKE in such a way that it derived its sample of points directly from the definition of the spatial relation in question, but at present the procedure simply uses a stored list of vectors corresponding to each relation.)

The procedure WAS enters these CONTEXTS in the order in which they were returned by BLICKE, namely the ones closest to (3 15) first. Within each CONTEXT it FETCHes successively any entities of the form ((AUFFAELLIG <object>) . z-value) which it may contain, finding them in decreasing order of z-value, which here - as described in section 2.13.3 above - represents relative salience.

All of the objects found in this way will rate fairly high according to the two criteria mentioned in section 2.13.2, but they must still be regarded as mere 'candidates' which must be evaluated and compared according to a more complete set of criteria.

The following criteria, which include the two mentioned in section 2.13.2, are at present incorporated in the definition of WAS:

(5) i: The degree to which the object satisfies the given spatial description. This is computed by calling PRUEFEN (see 2.13.1 above) and taking the z-value of the result.

ii: The intrinsic salience of the object. This is FETCHed from the associative net originally loaded from GEO. NET.

iii: The prominence of the object in the dialogue so far. This is a function of how often and how recently it has been referred to, and is computed as described in section 2.14.1.

The measures obtained by applying these criteria are combined according to the following formula to yield a measure of the mentionability of each object found:

(6) i · (ii + iii)

Note that even an object which fulfills the spatial description with a z-value of 1.0 may receive a low mentionability rating if it has a low intrinsic salience and hasn't played an important role in the conversation so far. On the level of performance, this is reflected in the fact that small objects in the scene are in general only mentioned spontaneously by the system if they have already played a role in the conversation.

A general procedure called FIND controls the search for relevant objects wherever visual search is simulated. FIND keeps track of the mentionability ratings associated with each object, determines when a sufficient number of mentionable candidates have been found, and selects a certain number of them to be finally mentioned. Applied to our present example, FIND directs the search according to the following instructions:

(7) Continue calling WAS with the arguments MANN4 and HINTER until 4 candidates have been generated with a mentionability rating of at least 0.2, or until no more can be found. Then return the 3 most mentionable of these candidates, in the order of their mentionability.

The list of objects returned by FIND is the semantic representation of the answer. It is transformed by the generation routines (see 2.15 and 2.16) into a natural-language description of the objects.

The three numbers 4, 0.2, and 3 in (7) can be viewed as resource-allocation parameters: the higher they are, the more thorough and selective the search will be, the longer the answer will be – and the more processing time will be required. These parameters are at present fixed for each type of question whose processing involves a call to FIND, but it is planned to make them depend on such factors as the interest the partner has expressed in spatial descriptions and the motivational state of the dialogue partner. This will permit a more sophisticated simulation of the dependence of resource allocation on dialogue context than that which is presently implemented (see 2.14.2).

Clearly, neither the set of criteria (5), the formula (6) for combining them, nor the selection strategy (7) can be claimed to be complete or final. Their use in HAM-RPM has, however, shown that some such rules are essential if a system is to generate easily understandable, natural-sounding spatial descriptions, and that the rules described here serve their purpose quite well. The implementation in HAM-RPM is designed so as to facilitate experimentation with different sets of rules.

Other types of requests involving searches for objects are handled with the help of specialized procedures structurally similar to WAS, and the searches are also controlled by FIND. Thus the answering of questions like

(8) WO STEHT DIE ALTE FRAU?
Where's the old woman standing?

differs from the process just described mainly in that

- objects are found which stand in the relation IN/ DER/ NAEHE (near) to the object named
- not only the names of the objects found are kept track of by FIND, but also the names of the more precise relations in which the named object stands to them.

Thus an element of the semantic representation of an answer to such a question would be

(9) ((HINTER 0.9) AMPEL2)

The number 0.9 associated with HINTER (behind) is the degree to which this relation holds between the object originally given and AMPEL2 (traffic-light2). These numbers are translated into hedges such as DIREKT (directly) and IN/ ETWA (roughly) if they lie within certain ranges (cf. 2.15).

All of the major general procedures used in this component have now been described, and examples have been given of the specialized procedures which are used in connection with particular types of requests involving search. It remains to be explained how the system recognizes which type of request is being made of it. This is accomplished straightforwardly by the procedure STEHT (stands), which is passed as arguments a triple of the sort used to represent the meaning of questions (see 2.11.1). In fact, except when STEHT is being called by the components responsible for the resolution and generation of noun phrases (see 2.8.4 and 2.15), its arguments are just the elements of the semantic representation of the question. For the examples (4) and (8) respectively, the arguments to STEHT would be:

(10) NIL (MANN4) (HINTER)
(11) (FRAU2) NIL NIL

The request's type is determined uniquely by which slots of the triple are NIL and which are not. If none of the slots are NIL the triple represents a yes-no question which can be answered straightforwardly with a call to PRUEFEN. All other cases involve search. In these cases, STEHT calls FIND, specifying the name of a particular procedure such as WAS.

A major extension of the simulation of visual search processes is planned: The information presently contained in the referential semantic network (see 2.1.3.8) could be distributed among the many 'glance'-CONTEXTs just as the information on saliences is. This would mean that in all cases where the system has to look for objects with particular properties – for example, in resolving the noun phrase DER OFFENE SCHRANK (the open cupboard) – it would have to enter various CONTEXTs, instead of simply accessing the referential semantic network, as it does at present (see, for example, 2.8.3).

This would provide some interesting, psychologically interpretable effects. For example, the noun phrase 'the truck' would be resolved much faster than 'the poster': since the property of being a truck is visually much more salient than that of being a poster, the corresponding entries about trucks, for example

(12) ((ISA TRUCK1 TRUCK) . 1.0)

would be present in many more CONTEXTs than the corresponding statements about posters. The system would thus have to enter fewer CONTEXTs when looking for entries about trucks than it would for posters.

In an effort to minimize the number of 'glances' it cast when looking for an object, the system would occasionally overlook an object with an unexpectedly low visual salience, thus reproducing some typically human errors.

2.14 CONVERSATIONAL TACTICS

HAM-RPM's behaviour at any given point in a dialogue is determined in many cases to a large degree by the course of the dialogue up to that point. In particular, the system adapts to a certain extent to the interest expressed by the dialogue partner in

- certain parts of the scene being discussed, and
- certain types of behaviour on the part of the system itself.

The precise nature of these adaptations, as well as the way in which they are realized, is described in the two sections to follow.

2.14.1 The record of references to objects

Each time an object is referred to, either by the human dialogue partner or by the system itself, a corresponding entry is ADDed to a special CONTEXT of the assertional data base called VORERWAEHNT (already mentioned). The entries in this CONTEXT have the following form:

(1) ((<token> <gender> <sum-of-references>) . <last-reference>)

for example,

(2) ((MANN2 MAS 22.5) . 17.5)

'last-reference', the z-value, is the sentence-number of the 1st sentence in which the object was referred to (the system's answers are assigned numbers 0.5 greater than the numbers of the questions which evoked them). 'Gender' is the gender of the noun or pronoun which was used when this latest reference was made.

Both of these pieces of information are used by the pronoun-resolution component, which, when trying to determine the referent of a given pronoun, looks for the most recently mentioned object which was last referred to using a lexical item of a certain gender (see 2.6). The component which resolves noun phrases uses this information in a roughly similar way when trying to resolve ambiguous noun phrases (see 2.8.6).

The number 'sum-of-references' in (1) is the sum of the numbers of all the sentences in which the object has been referred to. As it reflects the number of times the object has been mentioned during the entire conversation, but gives a greater weight to recent references, it can be used to construct a rough measure of the prominence of the object at a given point in the conversation. The system requires such a measure when deciding what refeference points to mention in an answer to a question. (see 2.13.4).

If the prominence of an object is to be determined at a time when the sentence number is N, the assertion beginning with the object's name is FETCHed from the CONTEXT VORERWAEHNT. (Actually, as explained in 2.6, two or three such assertions may be present, differing in the slots 'gender' and 'last-reference'; but the value of 'sum-of-references' will be the same for all of them.) The number 'sum-of-references' is normalized by being divided by N, and the quotient is taken to represent the prominence of the object. If no assertion containing the name of the object is found – i.e. if the object has not yet been referred to – its prominence is taken to be 0.

The ellipsis-recognition component (see 2.10) requires semantic and syntactic dialogue 'memories', but the storage of the necessary information is a simple matter, as only the last complete input is taken into account. Before the system prints its response, it checks to see whether the input was flagged as being complete. If so, it records the shallow structure and the semantic representation of the input as the values of two global variables; otherwise, the values of these variables remain unchanged.

2.14.2 Spontaneous justifications and detailed answers

In general, HAM-RPM justifies its answers to questions only when asked to do so in a particular case (see 2.5.1). It does, however, keep track of the number of times the dialogue partner has requested a justification by incrementing the global variable BEGRUENDUNGSFORDERUNG (request for a justification) after each such request. The value of this variable is checked each time an answer is generated: if it has reached the threshold of 3, any justification which the system has noted is appended to the answer, introduced by the word DENN (because):

(3) MAGST DU DIESEN APFELSAFT GERNE?
Do you like this apple juice?

(4) JA, DENN EIN APFELSAFTGETRAENK IST OFT SUESS UND EIN APFELSAFTGETRAENK IST ALKOHOLFREI.
Yes, because apple juice is often sweet and apple juice is non-alcoholic.

(cf. the dialogue sequence reproduced in section 2.5.1.)

The dialogue partner can also influence this aspect of the system's behaviour directly through a specific request. Such requests are at present recognized through pattern matching during the handling of 'idioms' (see examples (6) and (7) in section 2.4.4).

Another type of question for which the appropriate length of the answer is largely determined by the interests of the questioner includes those in which the system is required to describe some part of the scene (see 2.13.4). The number of objects mentioned in the answer varies from one to three (see 2.5.2) and is determined by factors exactly analogous to those described above in connection with justifications.

This is clearly a simple approach to the problem of cooperatively adaptive dialogue behaviour. A more adequate treatment would require, among other things, the ability to represent explicitly the human dialogue partner's presumed intentions and the system's strategies for recognizing and adapting to them.

2.15 NP-GENERATION

The task of the NP-generation component in HAM-RPM is inverse to that of NP-resolution described in section 2.8. That means that internal object names contained in the semantic representation of the system's answer are to be transformed into definite descriptions. In HAM-RPM we view NP-generation as a cognitive decision process whose result - a definite description - should enable the human dialogue partner to uniquely identify an object intended by the system. HAM-RPM encodes one and the same object in different ways depending on the context of objects in the domain which may be confused with it.

NP-generation is a central problem in natural-language generation since definite NPs are not only contained in complete sentences but very often are the only element of an elliptical answer. In HAM-RPM the NP-generation component is also invoked in early processing phases, e.g. by the routines which answer follow-up questions (see 2.5) and by the NP-resolution component in case of clarification dialogues, in which the system proposes a possible referent to the human dialogue partner (see 2.8.7).

In the current version of HAM-RPM the output of the NP-generation component in general has the following syntactic structure:

$$\text{NP} ::= \begin{array}{l}(\text{D-}) \; + \\ (\text{the})\end{array} \left(\begin{array}{l} \text{ADJ} \\ \{\text{ADJ}\}^{1\text{-}n} + \text{UND} + \text{ADJ} \\ \qquad\quad (\text{and}) \\ \{\text{ADJ}\}^{1\text{-}n} + \text{ABER NICHT} + \text{ADJ} \\ \qquad\quad (\text{but not}) \end{array}\right) + \text{NOM} + (\text{RELCL})$$

RELCL ::= WELCH SICH + PRP + NP + BEFINDET
(which is) (located)

Let us illustrate this general structure by some examples:

(1) TISCH1 D- TISCH
table1 the table

(2) STUHL1 D- ROT STUHL
chair1 the red chair

(3) SCHRANK4 D- ROT KAPUTT UND OFFEN SCHRANK
cupboard4 the red damaged and open cupboard

(4)	SCHRANK6	D- WEISS KAPUTT ABER NICHT OFFEN SCHRANK
	cupboard6	the white damaged but not open cupboard
(5)	SESSEL4	D- HINTER- BRAUN SESSEL WELCH SICH RECHTS/ NEBEN D- MITTLER ROT BILD BEFINDET
	armchair4	the rear brown armchair which is located to the right of the red picture in the middle

The reader may have noticed that the expressions constructed by the NP-generator contain only stems of words. The output of the NP-generator undergoes several surface transformations, described in the next section (2.16), until it becomes morphologically and syntactically well formed.

The NP-generation process is governed by the principle of confusion-avoidance. In (1) the table is supposed to be the only one in this discourse world and hence is identified as 'the table'. When the object to be described is one of several belonging to the same conceptual class, the system looks among the properties of the object stored in the referential semantic net for one which distinguishes it from its cohyponyms as in example (2).

If there are several properties which refer only to the intended object and not to any of its cohyponyms, we have the case multiple codability investigated by HERRMANN/LAUCHT 1976. In this situation the system applies the following algorithm to select one of these distinguishing properties. First it looks among the distinguishing properties for one whose z-value exceeds the z-values of the others markedly. When it finds one it encodes this property in the definite description. For example, suppose RED and OLD are the distinguishing properties of CHAIR1 and the referential net contains the assertions ((REF CHAIR1 RED) . 0.8) and ((REF CHAIR1 OLD) . 0.4). In that case the system selects the property RED and generates the NP 'the red chair'.

Second, when the first step does not succeed, i.e. the z-values of the distinguishing properties are approximately equal, the system encodes that property which it prefers according to a preference order of dimensions stored in the knowledge base. This preference order is interindividually relatively constant and in the present system it is represented simply as a list of dimensions, e.g. (colour, contrast, size, age ...).

For example, suppose that BROWN and BIG are the distinguishing properties of CHAIR2 and the assertions ((REF CHAIR2 BROWN) . 0.75) and ((REF CHAIR2 BIG) . 0.8) are contained in the referential semantic network. Then the system selects BROWN and generates the NP 'the brown chair' because it prefers colour to size. If the system does not find even one distinguishing property, it uses - in the current implementation - all of the properties of the intended object in the referential net, with no further attempt to find a minimal characterizing set. This means that the system uses redundant labels, which saves time both in generation and in the hearer's interpretation of definite descriptions. In that case the ordering of the adjectives in the generated NP reflects the preference order defined by the selection algorithm discussed above. Example (3) arose in this way.

If the characterizing properties of the intended object are a subset of those of a cohyponym, HAM-RPM uses negative characterizations, as illustrated in example (4) above.

The presence of several objects which are indistinguishable on the basis of their attributes alone is the worst case which can occur. The only way to distinguish such objects is by reference to spatial relations. Since objects chosen as spatial reference points in such a description must in turn be identified, the generation algorithm is applied recursively, as illustrated in example (5) above.

Perhaps the most difficult problem in NP-generation is the unique identification of an object when there are objects with exactly the same properties in its immediate neighbourhood. In the present version, the system then first describes the position of the intended object relative to the other objects in its group using the spatial attributes RECHT (right-hand), LINK (left-hand), VORDER (front), HINTER (rear) and MITTLER (middle). These complex relations are constructed from the elementary relations 'in front of', 'behind', 'to the left of' and 'to the right of', which are generated by the spatial-relations component (see 2.13).

Then the system identifies the group of which the intended object is a member within the scene as a whole by spatial relations which are always coded into a relative clause as illustrated in example (5).

Concluding this discussion of HAM-RPM's NP-generator, we would like to stress that our generator in contrast to those described in WOODS et al. 1976 and WALKER 1976 takes into consideration the communicative context of reference. A more complete discussion of this algorithm, which is distinguished from the related ones of WINOGRAD 1972 and WONG 1975 by its adaptation to results of psychological research on reference generation (HERRMANN/DEUTSCH 1976), is contained in Reports Nos. 5 and 7.

2.16 SURFACE TRANSFORMATIONS

The system's final task in answering a question posed by the natural partner is to generate a morphologically and syntactically well-formed output.

Regarding yes/no-questions, in most cases only punctuation marks have to be added to the deep structure of the answer (e.g. JA, NEIN or DOCH). If, however, an extensive clarification dialogue (see 2.8.7) has interrupted a single question-answer sequence, affirmation or negation explicitly refer to the initial question of the natural partner. This is achieved by filling out the one-word answer with a paraphrase of the initial question. The shallow structure as transformed by word-order rules (see 2.9.1) forms the basis of a paraphrase, which has to be retransformed into a natural utterance, as several processing steps had modified the shallow structure, including lexical analysis, modality and NP-resolution.

In the following example the natural language output (4) is obtained by transforming the deep structure of the answer (3) and the shallow structure of the question (2) by means of NP-generation (see 2.15) and surface transformations.

(1) STEHT DER STUHL NEBEN DEM TISCH?
Is the chair standing next to the table?

WELCHEN STUHL MEINST DU? WIESO FRAGST DU MICH DAS?
Which chair do you mean? Why do you ask me so?

WEIL ES FUENF DAVON GIBT!
Because there are five of them!

WELCHEN STUHL MEINTEST DU? DEN SCHWARZEN.
Which chair did you mean? The black one.

(2) ((NOM (STUHL5)) (VRB (STEH)) (PRP (NEBEN)) (NOM (TISCH1)))

(3) NEIN.
(No)

(4) NEIN, DER SCHWARZE STUHL STEHT NICHT NEBEN DEM BRAUNEN TISCH.
No, the black chair is not standing next to the brown table.

The same difficulty arises with answers to wh-questions. The deep structure of the answer does not resemble natural language output. Conversational tactics have already determined some aspects of the answer (see 2.14.2) and internal names of objects have been replaced by noun phrases (see 2.15). But the answer still consists solely of canonical forms, i.e. only stems of words are used and punctuation marks are in general not present.

A different source of utterances of the system (see 3.2) are rejections, clarification dialogues and 'thinking aloud' (see 2.7, 2.8). These utterances, generated by different components of HAM-RPM, are composed of canonical forms with no explicit syntactic coherence whatsoever.

In the following the approach taken in HAM-RPM to transform these internal representations will be described. Any input to the surface transformations will be called the 'underlying utterances'. Their structure was introduced in the sections describing the components which generate them.

In a first step, each occurrence of 'D- MM' (the MM) is replaced by 'ICH' (I) as referential identity of this personal pronoun is unique for the system (see 2.6). The proper surface case for this pronoun is determined when inflectional suffixes are attached to nouns according to surface case. Numbers from 1 to 20 are replaced by the corresponding words at the same time.

The three basic grammatical categories for establishing the correct morphological forms of adjectives and nouns in sentences are gender, case, and number. For verbs the categories person and time replace case and gender. Having ascertained these three features any regularly inflected word can be generated correctly.

A prerequisite for detecting the relevant categories therefore is the construction of the preterminal string of the underlying utterance. The parts of speech of the known words are attached using procedures of the lexical-analysis component (see 2.4.3). The part of speech of an unknown word - which, of course represents an inconsistency in the knowledge base, since the system should have at least grammatical knowledge about any words it uses - is represented by a filler category called FUELLW (filler word) (cf. the class 'bogus' used by GERSHMAN 1977) in the preterminal string of the underlying utterance.

Together with parts of speech the gender of nouns is supplied by lexical analysis. Punctuation marks – if there are any, such as dots and blanks in the 'thinking aloud' utterances – are assigned to the category SATZZ (punctuation marks). An example of the lexical analysis employed in surface transformations (6) of an underlying utterance (5) is shown below.

(5) (. . . VOR D- TISCH . . . E- BRAUN STUHL . . .)

(6) ((SATZZ (.)) (SATZZ (.)) (SATZZ (.)) (PRP (VOR)) (DET (D-)) (NOM (TISCH)) (SATZZ (.)) (SATZZ (.)) (SATZZ (.)) (SATZZ ( ))(SATZZ ( )) (DET (E-)) (ADJ (BRAUN)) (NOM (STUHL)) (SATZZ ( )) (SATZZ ( )) (SATZZ ( )) (SATZZ (.)) (SATZZ (.)) (SATZZ (.)))

The category 'number' is at present restricted to singular in the generation component of HAM-RPM. The preterminal string of an underlying utterance having thus been generated, the only category still to be determined for nouns and adjectives is surface case.

In German there are two sources yielding information about surface case. Prepositions determine the case of the following noun phrase, although frequently not uniquely; syntactic dominance relations and/or specified verb frames such as those employed by GOLDMAN 1975 supply clues to the case of noun phrases. Syntactic information might be obtained by parsing the underlying utterance. This would, however, run into difficulties, as most of the system's utterances are not complete sentences, nor are they always grammatically well formed. A parser would, for example, have great trouble analysing 'thinking aloud' utterances.

In HAM-RPM a different method has therefore been developed which employs parts of speech in the delimitation of parts of sentences. Methodologically this heuristic approach resembles the rules introduced by GERSHMAN 1977 for identifying noun groups, especially in the respect that syntactically incorrect sentences should not necessarily be rejected.

The table below shows some rules based on parts of speech which introduce boundaries into underlying utterances. Square brackets in the examples indicate the boundaries between parts of sentences.

(7) The current word is a noun. A boundary is established behind it, if the word which follows is neither a verb, an interjection (ITJ) nor a filler word (FUELLW).

Examples:

([(DET (D-)) (ADJ (ROT)) (NOM (STUHL))] [(DET (D-)) (ADJ (BRAUN)) (NOM (TISCH))] . . .)
the red chair the brown table . . .

([(DET (D-)) (ADJ (KLEIN)) (NOM (TISCH)) (ITJ (ALSO))])
the small table then

([(DET (E-)) (NOM (PARKZONE)) (VRB (SEIN))] [(DET (E-)) (NOM (TEIL)) . . .])
a parking zone is a part . . .

(8) The current word is a verb. A boundary is set up if the word which follows is neither an adverb nor an adjective. If it is a pronoun the boundary is set up after this pronoun.

Examples:

([. . . (PRP (AUF)) (DET (D-)) (NOM (TISCH)) (VRB (STEH))] [(DET (D-)) . . .])
. . . on the table is . . .

([(NOM (WASSER)) (VRB (SEIN)) (ADV (GRUNDSAETZLICH)) ...])
water is by definition ...

([(PRP (VOR)) (DET D-)) (NOM (HAUS)) (VRB (SEH)) (PRN (ICH))] [(NEG (KEIN)) (NOM (FRAU))])
in front of the house I do not see a woman

(9) Successive punctuation marks, such as the dots and blanks in (6), are grouped between boundaries.

As a result of the application of these rules to the underlying utterance from left to right, it is partitioned into smaller sections. Each of these sections is now subject to morphological transformation processes.

Depending on the parts of speech inflectional suffixes are added to the canonical forms. For nouns, adjectives and articles case information is supplied by examining the surroundings of the word, looking especially for prepositions, and the partitioning of the underlying utterance.

Rather simple rules describe the inflectional suffixes added to noun stems according to case and gender as shown in Fig. 13. The terminal symbol '∅' represents a zero morpheme. The 'E' of the genitive suffix is elided if the last letter of the stem is a vowel.

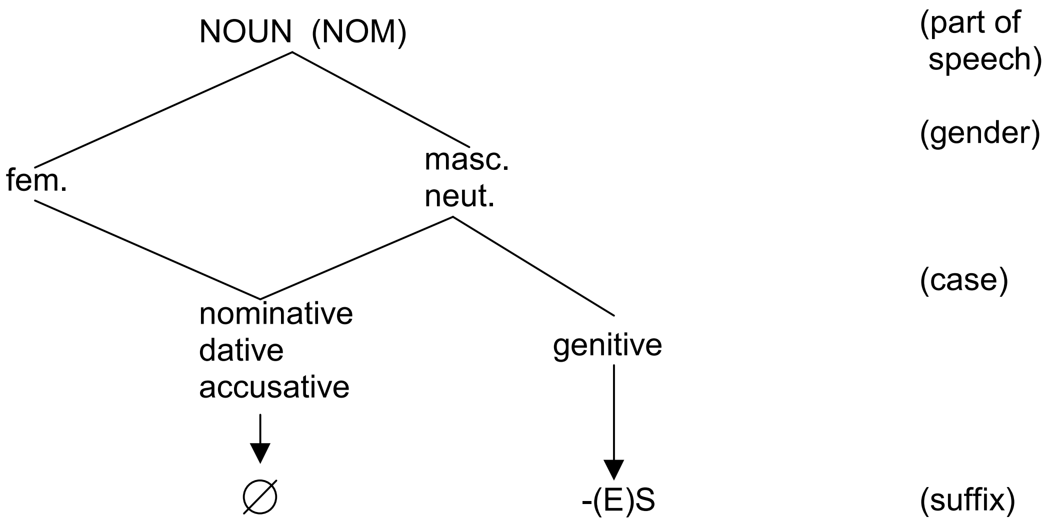


Fig. 13

In German the inflectional suffixes of adjectives also depend on the type of the article, i.e. definite or indefinite article. Fig. 14 represents the rules for generating inflected adjectives in HAM-RPM.

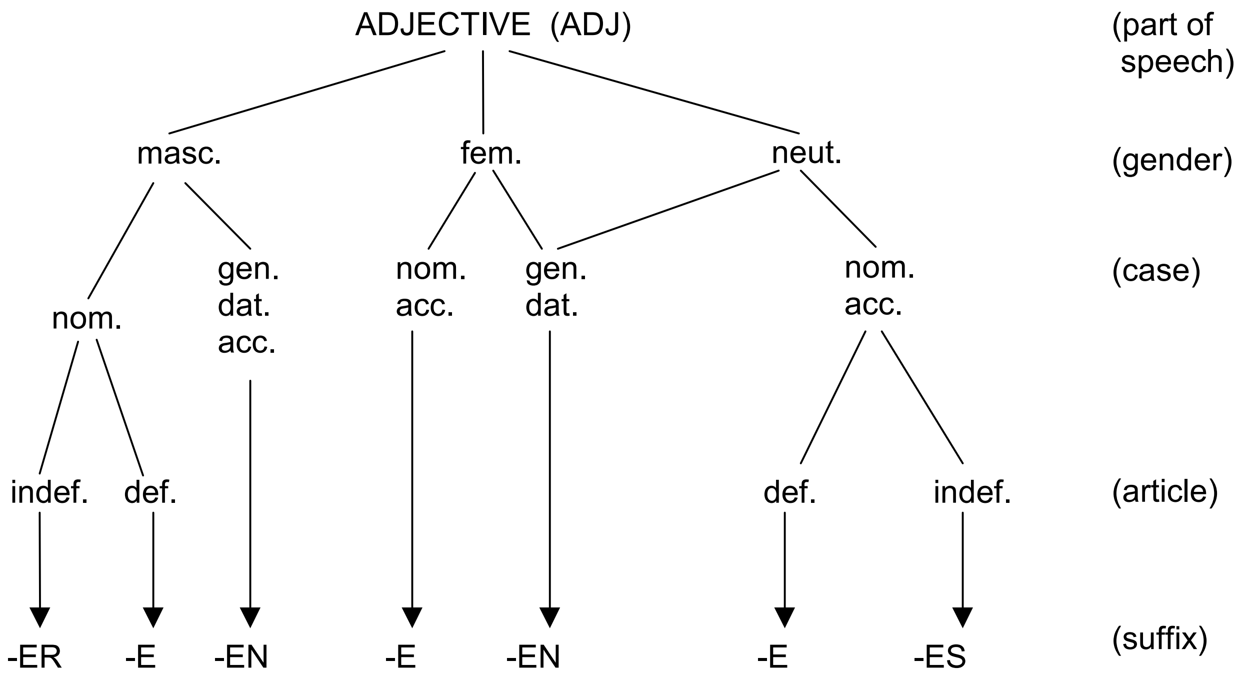


Fig. 14

Articles in German also vary according to gender and case. For the purpose of generating inflected words the canonical form of the definite article is 'D', for the indefinite article 'EIN' which are the maximal common letters of all article forms.

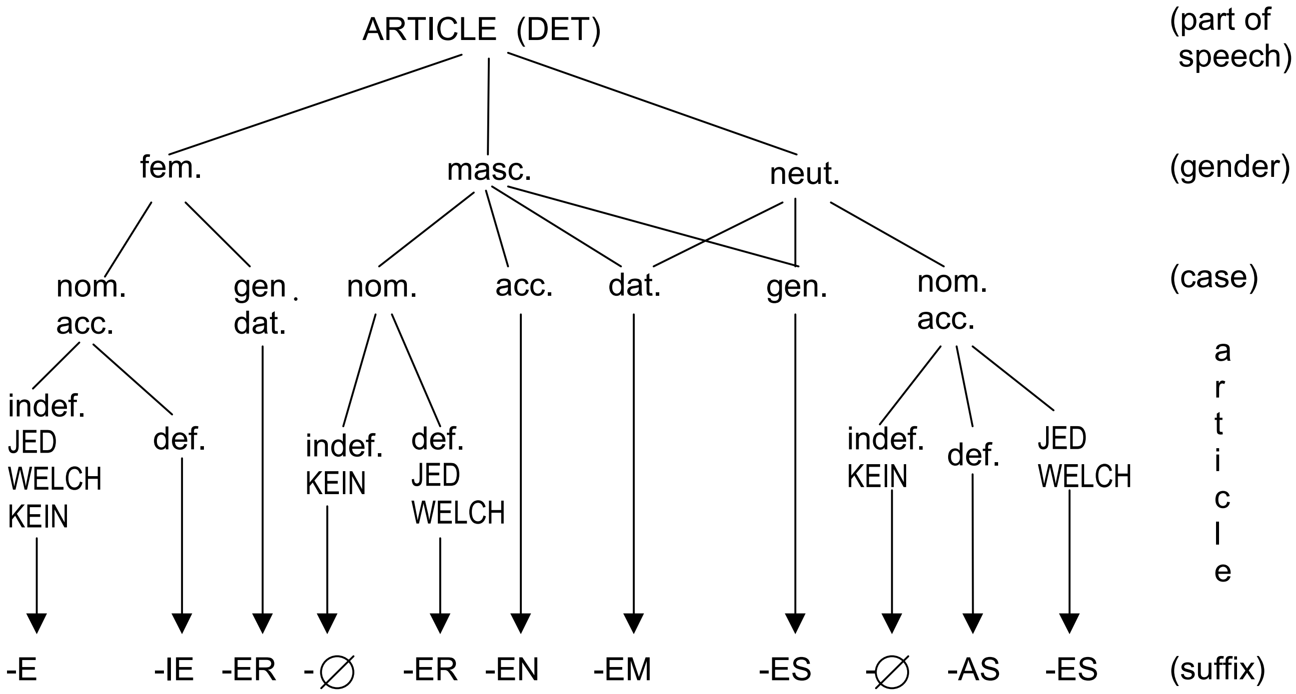


Fig.15

As seen from Fig. 15 by the same rules suffixes are added to the canonical forms of the quantifiers KEIN (no) and JED (each) and to the relative or interrogative pronoun WELCH (which). In the case of relative pronouns the gender and the case of the preceding noun determines the choice of the inflectional suffix.

Preliminary rules supply verb forms with inflectional suffixes. Tense and number are at present restricted to present tense and singular. The category an far as surface transformations are concerned is therefore restricted to the three possibilities in the singular.

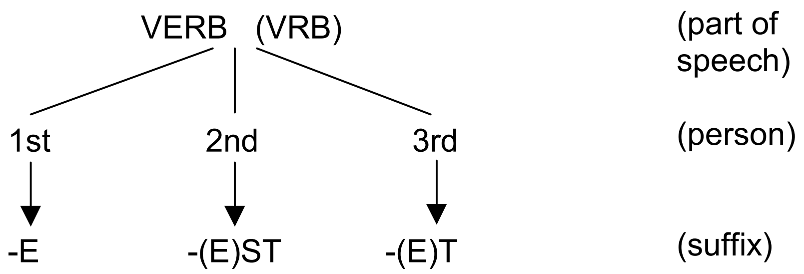


Fig. 16

The person is determined by looking for either ICH (I), DU (you) or something else in the part of the underlying utterance being investigated.

Generally speaking the rules described above only apply to regularly inflected words. Irregularities such as vowel mutation will not be treated correctly until lexical structures are developed to represent them.

Having transformed word stems into inflected words in all parts of the underlying utterance the last task of surface transformations is to introduce punctuation marks joining the parts together. This is achieved by classifying the single parts and establishing connection rules over these classes. Four classes of parts are defined:

- VST - The part is complete (VST: vollständig) if it consists of at least a verb and a noun group.
- UNV - The part is incomplete (UNV: unvollständig) if there is no verb.
- REL - The part is a relative clause (generated by NP-generation, see 2.15). Note that subclauses in German are normally delimited by commas.
- ZCH - The part consists solely of punctuation marks.

For combinations of these classes a decision is made as to whether a comma or an UND (and) is to be inserted between them. The following combinations are examined:

combination	comma ?	and ?
VST UNV > VST	yes	yes
VST UNV > REL	yes	no
REL < VST UNV	yes	only if there are no further parts.
UNV — UNV — UNV	yes	no
UNV — UNV — VST	no	yes
UNV — UNV < REL ZCH	yes	no
UNV — UNV	no	yes

After inserting commas and UND according to the rules above the complete underlying utterance has been transformed into a natural language utterance. The final punctuation mark is a full stop if no other mark, such as question mark or an exclamation mark, had terminated the underlying utterance.

Starting from the rather heuristic approach which is implemented at present to transform canonical utterances into surface structures a major task is to elaborate a complementary component of the lexical-analysis component (see 2.4). Note that the problems for surface transformations originate not only from the rich morphological structure of German but also from the limited capabilities of the current version of HAM-RPM in semantic and pragmatic processing.

Our system finally has to be able to analyse each of its own utterances by the same process as is already done for utterances of the natural partner. This postulate resembles the previously stated analogy between NP-resolution and NP-generation. In addition to eliminating restrictions already mentioned in this section important topics to be worked on are the use of idiomatic expressions and pronominalization in surface transformations and stylistic variation guided by conversational dynamics.

3 TRANSCRIPTS OF ACTUAL DIALOGUE SESSIONS

This section contains transcripts of actual dialogue sessions with HAM-RPM. The domain of discourse underlying the first session (see fig. 17) is WORLD2, the traffic scene (see 2.1.1). The last sentence in this session is traced, since the user has changed the output type via (IO-OPTIONS) (see 2.2.1).

In the second dialogue session (fig. 18) the domain of discourse is WORLD3 (see 2.1.1).

All system messages and the interaction with the user which follow the command DO WELT N until HAM-RPM signals that it is ready to begin the conversation by greeting the user have been removed from the actual protocol. Further annotated transcripts of actual dialogue sessions in which WORLD1 and WORLD2 are used as underlying discourse worlds are contained in Memo No. 4.

Fig. 19 summarizes all possible speech-act sequences which in the present version of HAM-RPM may occur in the communication with the system.

.DO WELT 2

⋮

GUTEN MORGEN!
Good morning!

(1) ? WAS BEFINDET SICH IM MOMENT NEBEN DEM FACHBEREICH FUER INFORMATIK?
What's next to the Department of Computer Science at the moment?
DER KLEINE BAUM.
The small tree.

(2) ? SIEHST DU AUCH SIE?
Do you see her too?
AUF WAS BEZIEHT SICH 'SIE'? AUF DIE FRAU.
What does 'her' refer to? To the woman.
JA.
Yes.

(3) ? STEHT NEBEN DEM BAUM EIN ROTER WAGEN?
Is there a red car standing next to the tree?
NEIN.
No.

(4) ? IST DENN EIN ROTER WAGEN VOR DER AMPEL RECHTS VORNE IM AUGENBLICK?

Is there a red car in front of the traffic light in front and to the left at the moment?

JA.

Yes.

(5) ? IST DER BAUM, WELCHER RECHTS NEBEN DER AMPEL STEHT,GROSS?

Is the tree which is next to the traffic light big?

. . . ES SOLL HIER RECHTS NEBEN EINER AMPEL EINEN BAUM GEBEN . . . DA MUSS ICH MAL GENAUER HINSEHN . . .

. . . behind a traffic light there's supposed to be a tree . . . I've got to take a closer look …

JA.

Yes.

(6) ? WELCHE FARBE HAT SIE MOMENTAN?

What colour is it (gender: feminine) at the moment?

GRUEN

Green.

(7) ? IST EINE PARKZONE GETEERT?

Is a parking zone tarred?

JA.

Yes.

!!! DIE AMPELN WURDEN UMGESCHALTET !!!

!!! The traffic lights have changed !!!

(8) ? WIE KOMMST DU DARAUF?

What makes you think so?

EINE PARKZONE IST MEIST EIN TEIL VON EINER STRASSE, UND EINE STRASSE IST OFT GETEERT.

A parking zone is usually a part of a street, and a street is often tarred.

(9) ? BEFINDET SICH IN DER NAEHE VOM HAUS HINTEN LINKS EIGENTLICH EINE PARKZONE?

Is there in fact a parking zone near the house in back and to the left?

JA.

Yes.

(10) ? WAS FUER EINE FARBE HAT DIE AMPEL VORNE RECHTS DENN JETZT?
What colour is the traffic light in front and to the right now?
ROT.
Red.

(11) ? DAS BLAUE AUTO, WELCHES IN DER NAEHE VOM HAUS HINTEN RECHTS PARKT, IST ALT?
The blue car that's parked near the house in back and to the right is old?
. . . EIN BLAUES AUTO . . . UND ZWAR IN DER NAEHE VOM HAUS HINTEN RECHTS . . .
. . . a blue car . . . next to the house in back and to the left . . .
IN DER NAEHE VON DEM HAUS RECHTS HINTEN SEHE ICH KEIN BLAUES AUTO.
I don't see any blue car near the house to the right and in back.

(12) ? WELCHE FAHRZEUGE SIND DENN UEBERHAUPT ALT?
Which vehicles are old then?
DER BLAUE PERSONENWAGEN, DER PARKENDE UND DUNKELE PERSONENWAGEN LINKS VORNE, DER PARKENDE UND WEISSE PERSONENWAGEN LINKS VORNE, DER PARKENDE UND GELBE PERSONENWAGEN RECHTS VORNE UND DER ROTE PERSONENWAGEN.
The blue car, the dark parked car in front and to the left, the white parked car in front and to the left, the yellow car in front and to the right and the red car.

? (I0-OPTIONS)

UEBLICHE IO-OPTIONS?	NEIN
Standard IO-options?	No

AUSGABE ALLER TEILERGEBNISSE?	JA
Trace mode?	Yes

STRUKTURBAUM AUCH?	NEIN
Parse tree too?	No

LINELENGTH ('J' = 72) :	J

ZEITANGABE?   NEIN
Time measurement? No

TTY-I-0?   JA

(13) ? WAS STEHT HINT%ER DER DAHME, WELCHE SICH NEBEN DEM GELBEN UND ALTEN AUTO BEFINDET?
What's behi%nd the lahdy which is next to the yellow and old car?

(IDIOMENBEHANDLUNG:)
(WAS STEH HINTER D- DAHME WELCH NEBEN D- GELB UND ALT AUTO STEHT)
WAS HEISST 'DAHME'?   ICH MEINTE DAME
(L VERBESSERUNG DAHME (DAME))
WAS HEISST 'DAME'?   DASSELBE WIE FRAU
(L SYNONYM DAME (FRAU))
(WAS STEH HINTER D- FRAU WELCH NEBEN D- GELB UND ALT PERSONENWAGEN STEH)
(W-FRAGE)
. . . NEBEN DEM GELBEN UND ALTEN PERSONENWAGEN . . . EINE FRAU . . .
. . . next to the yellow and old car . . . a woman . . .
((IPRN (WAS)) (VRB (STEH)) (PRP (HINTER)) (NOM (FRAU1)))
(KEINE PRAETRANSFORMATION MOEGLICH)
(SATZ AKZEPTIERT :)
(S (NP (IPRN (WAS)))   (VP (VRB (STEH)) (PP (PRP (HINTER)) (NOM (FRAU1))))
(NIL (FRAU1) (HINTER))
(PERSONENWAGEN6)   0.875
(AMPEL1)   0.35

DER BRAUNE PERSONENWAGEN.
The brown car.

(14) ? TSCHUESS.
Good bye.

DIE NEU GELERNTEN WOERTER:
New words:
(DAME (SYNONYM (FRAU)))   NEIN
DAS WAREN ALLE.
KEINE NEUEN WOERTER.
TSCHUESS BIS ZUM NAECHSTEN MAL!
See you later!
*
.DO WELT 3
:
GUTEN ABEND!
Good evening!

(1) KANNST DU MIR BITTE MAL SAGEN, WO SICH HIER EIN GETRAENK BEFINDET?
Could you please tell me where there's a drink?
AUF DEM COUCHTISCH.
On the couch table.
(8.9 SEC)

(2) ? KANNST DU DEN WHISKY TRINKEN?
Can you drink the whisky?
ES GIBT HIER KEINEN WHISKY.
There's no whisky here.
(2.7 SEC)

(3) ? WAS IST DENN IM GLAS?
What's in the glass then?
DAS APFELSAFTGETRAENK.
The apple juice. (6.1 SEC)

(4) ? MAGST DU DEN APFELSAFT GERNE?
Do you like the apple juice?
ICH GLAUBE JA.
I think so.
(7. 6 SEC)

(5) ? WIESO DENN?
Why?

EIN APFELSAFTGETRAENK IST OFT SUESS UND EIN APFELSAFTGETRAENK IST ALKOHOLFREI.
Apple juice is often sweet and apple juice is non-alcoholic.

(0. 9 SEC)

(6) ? WAS IST AUF DEM ROTEN BILD?
What's in the red picture?
DIE MARILYIN MONROE.
Marilyn Monroe.

(10.2 SEC)

(7) ? FINDEST DU SIE SCHOEN?
Do you find her pretty?
JA.
Yes.

(3.8 SEC)

(8) ? WARUM DENN EIGENTLICH?
Really, why?
IST DOCH KLAR!
It's obvious!

(0. 9 SEC)

(9) ? WELCHE HAARFARBE HAT SIE DENN?
What colour hair has she got?
BLOND. (1.5 SEC)

(10) ? WELCHE FARBE HAT EIN KLAVIER?
What colour is a piano?
MEISTENS SCHWARZ.
Usually black. (2.3 SEC)

(11) ? SIEHST DU EIGENTLICH EIN PIANO HIER?
Do you actually see a pianoforte here?
WAS HEISST 'PIANO'? DAS IST DASSELBE WIE KLAVIER.
What's the meaning of 'pianoforte'? The same as piano.
JA.
Yes.

(6. 9 SEC)

(12) ? WELCHE FARBE HAT DIESES KLAVIER?
What colour is this piano?
WEISS.
White.

(2.9 SEC)

(13) ? IST DER TISCH GROSS?
Is the table big?

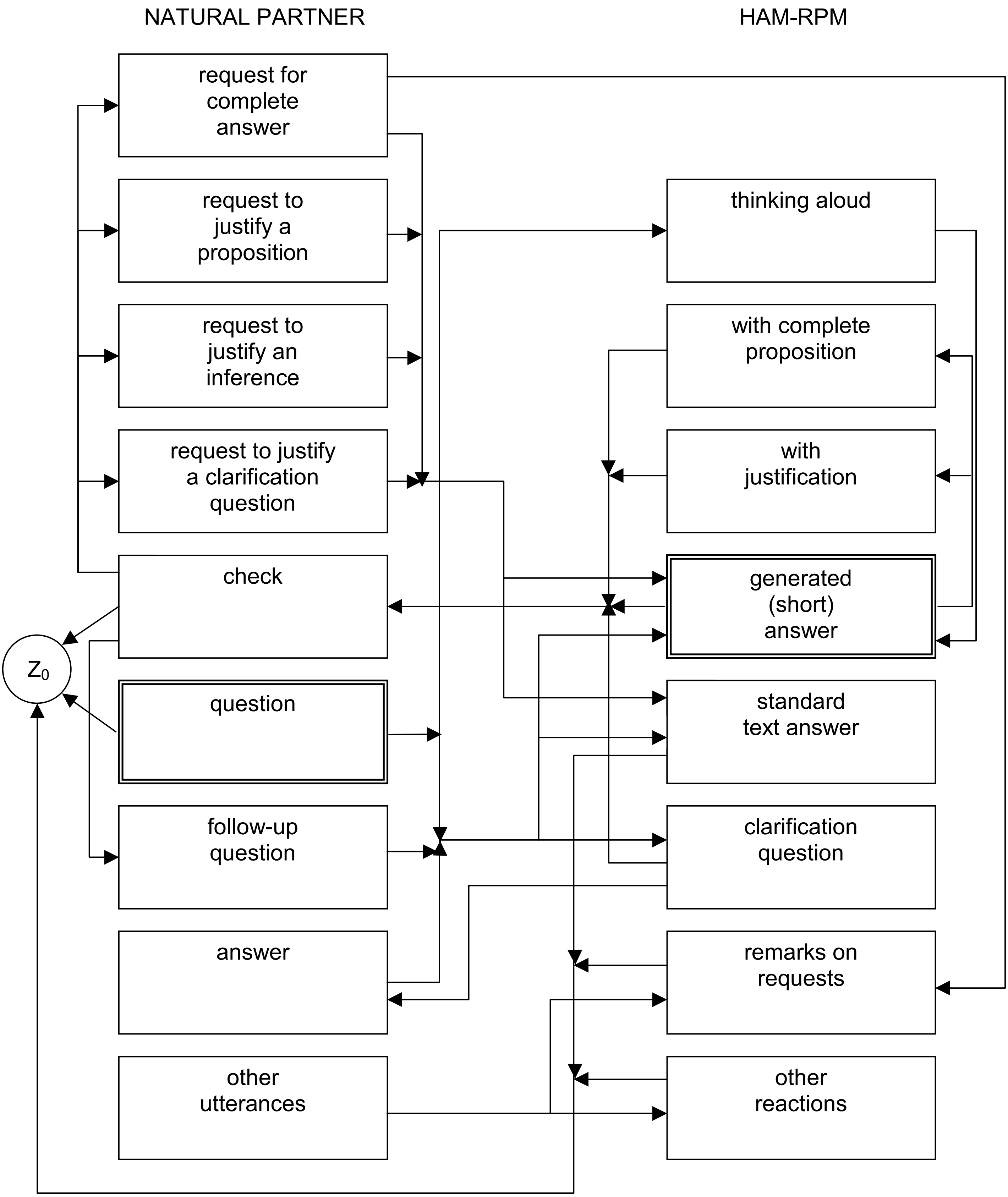
MEINST DU DEN COUCHTISCH?	WIESO WILLST DU DAS WISSEN?
Do you mean the couch table?	Why do you want to know?

WEIL ES ZWEI GIBT!
Because there are two!
MEINTEST DU DEN COUCHTISCH? NEE.
Did you mean the couch table? Nope.
DEN ESSTISCH ALSO.
The dining table then.
JA, DER ESSTISCH IST GROSS.
Yes, the dining table is big.

(6.3 SEC)

(14) ? TSCHUESS.
Good bye.
⋮
*

Fig. 18



NATURAL PARTNER
HAM-RPM
request for complete answer
request to justify a proposition
request to justify an inference
request to justify a clarification question
check
question
follow-up question
answer
other utterances
Z0
thinking aloud
with complete proposition
with justification
generated (short) answer
standard text answer
clarification question
remarks on requests
other reactions

Fig. 19

4 HAM-RPM'S PROGRAMMING ENVIRONMENT

4.1 Technical data

A non-compiled version of HAM-RPM is running on the DECsystem 1070 (PDP-10) of the Fachbereich für Informatik of the University of Hamburg under the TOPS10 Operating System. Comprising approximately 400 LISP/FUZZY procedures, the current version occupies 150K of 36-bit words and requires from one to fifteen seconds for a response.

4.2 Auxiliary software

Although Rutgers-UCI-LISP represents one of the most advanced interactive programming systems available, we have found it worthwhile to develop additional auxiliary software to help to overcome the problems which arise when several designers work together on a large system like HAM-RPM. In such a situation the designers, as well as the users, must constantly deal with procedures and data whose design they have not directly participated in.

Complete documentation and manuals could, in principle, provide the necessary information, and in fact reference material does exist for most parts of the system. But on the one hand, the experimental nature of HAM-RPM limits the value of such documents, as it means that any particular version of the system is likely to be used for only a relatively short time. On the other hand, the fact that HAM-RPM is implemented in an interactive programming language makes available other means of making the system understandable and manageable. These fall mainly into two categories:

- techniques which make parts of the system largely self-explanatory, so that the use of reference material is in general unnecessary.
- software which makes possible efficient, semi-automatic communication among the designers of the system, so as to help them to keep abreast of new developments both on a large and on a small scale.

A technique which falls into the first category is a simple but effective abbreviation system for LISP. This has increased greatly the readibility of the code of HAM-RPM. It is always useful to be able to assign long, self-explanatory names to procedures and variables, such as INFERENZ-REGELN-NACH-PRAEFERENZEN-ORDNEN (order inference rules according to preferences). The use of such names often makes it possible, for example, to understand a call to a procedure within another procedure without looking at either the definition of the called procedure or its documentation. Self-explanatory names are in this respect even better than comments within a procedure's definition which explain its function. The use of such names on a large scale is only practical, however, if the designer can, while the procedure is being modified and tested, type in an abbreviation which is immediately expanded by the system into the full form.

At a number of points the system must request some sort of input from the user, e.g. when the system is being started (see 2.2.1). The user must know what inputs are admissible and what consequences they will have. Such sets of possible inputs and their respective consequences are represented within HAM-RPM's programming environment as production systems which are handled by an interpreter with the following special properties, among others:

- It doesn't accept any input which doesn't match any of the patterns provided.
- If the user types HELP, the production system is printed in an easily readable form which in most cases permits it to be understood adequately even by a naive user.

The second category of techniques mentioned above comprises several packages of LISP procedures which allow the user to write messages of various sorts onto files directly from the LISP system. These packages make it possible for him to transmit useful information quickly without interrupting his work with the system. The efficiency of this type of communication is largely due to the fact that all of the facilities of the LISP system are available to facilitate the preparation of the message. Such things as formatting and the printing of the date and the user's name are done automatically. More important, data structures of interest can be printed and examples of interactions with the simulated dialogue partner can form part of the message.

One of these packages is used for writing short notes to a text file which is consulted regularly by all of the system's designers. This package is also used to generate semi-automatic protocols of faulty system performance during dialogues.

Another package in this category is a documentation system which elicits a standardized description of a procedure from its designer in prompting mode.

The very number of auxiliary procedures used in HAM-RPM's programming environment at first created a problem: on the one hand such procedures should be readily available when needed, but on the other hand it would be impractical to load them all every time the system was used. This problem was solved through the creation of a library system for LISP: Most of the auxiliary procedures are not loaded when the LISP system is initialized; instead, a catalogue is loaded which contains the names of the files on which they can be found. A special definition is assigned to all of the library procedures when the system is initialized.

When a library procedure is called, this definition in effect tells the interpreter to load the true definition of the procedure from the file listed in the catalogue before evaluating the call.

The usefulness of these techniques so far in keeping HAM-RPM manageable and modifiable suggests that they will be adequate to support the extensive development which is planned for the system in the future.

ACKNOWLEDGEMENT

This research is currently being supported by the Deutsche Forschungsgemeinschaft.

HAM-RPM PUBLICATIONS

Reports

1 v. Hahn, Walther, Henskes, Dieter, Hoeppner, Wolfgang, Wahlster, Wolfgang: HAM-RPM: Ein Redepartnermodel als Simulationsprogramm. In: Braunmüller, K., Kürschner, W. (eds.): Grammatik. Akten des 10. Linguistischen Kolloquiums Tübingen 1975. Band II. Tübingen: Niemeyer 1976. 337-357

2 Wahlster, Wolfgang, v. Hahn, Walther: Einige Erweiterungen des natürlichsprachlichen AI-Systems HAM-RPM. In: Laubsch, J., Schneider, H.-J. (eds.): Dialoge in natürlicher Sprache und Darstellung von Wissen. Arbeitstagung der Fachgruppe 'Künstliche Intelligenz' der Gesellschaft für Informatik. Freudenstadt 1976. 204-225

3 v. Hahn, Walther, Hoeppner, Wolfgang, Jameson, Anthony, Wahlster, Wolfgang: HAM-RPM: Natural Dialogues with an Artificial Partner. In: Proceedings of the AISB/GI Conference on Artificial Intelligence. Hamburg 1978. 122-131

4 v. Hahn, Walther: Überlegungen zum kommunikativen Status und der Testbarkeit von natürlichsprachlichen Artificial-Intelligence-Systemen. In: Sprache und Datenverarbeitung 1, 1978, 145-169

5 Wahlster, Wolfgang: Die Simulation vager Inferenzen auf unscharfem Wissen: Eine Anwendung der mehrwertigen Programmiersprache FUZZY. In: Ueckert, H., Rhenius, D. (eds.): Komplexe menschliche Informationsverarbeitung. Bern: Huber 1979. 249-259

6 v. Hahn, Walther: Probleme der Simulationstheorie und Fragepragmatik bei der Simulation natürlichsprachlicher Dialoge. In: Ueckert, H., Rhenius, D. (eds.): Komplexe menschliche Informationsverarbeitung. Bern: Huber 1979. 260-269

7 Wahlster, Wolfgang, Jameson, Anthony, Hoeppner, Wolfgang: Glancing, Referring and Explaining in the Dialogue System HAM-RPM. In: American Journal of Computational Linguistics. 1979

8 v. Hahn, Walther: Dialogkohärenz in natürlichsprachlichen AI-Systemen. Dezember 1978

9 Wahlster, Wolfgang: Algorithmen zur Beantwortung von 'Warum'-Fragen in Dialogsystemen. Januar 1979

10 v. Hahn, Walther: Überlegungen zum Handlungsrahmen von Fragen in Artificial-Intelligence-Systemen. Januar 1979

11 Wahlster, Wolfgang: ATNs und die semantisch-pragmatische Steuerung der Analyse und Generierung natürlicher Sprache. März 1979

Memos

0 Jameson, Anthony: Eine Einführung in die interaktive Arbeit und die Programmierung mit FUZZY. 1978

1 Wahlster, Wolfgang: Eine kurze Einführung in die Organisation der assoziativen Netze in der Programmiersprache FUZZY. Februar 1978

2 Hoeppner, Wolfgang: Nominalgruppen mit unbestimmtem Artikel. Februar 1978

3 Wahlster, Wolfgang: Zur Überführung eines Konstituentenstrukturbaums in eine semantische Repräsentationskonstruktion mit Hilfe von Pattern-Matching-Operationen. Mai 1978

4 Hoeppner, Wolfgang, Wahlster, Wolfgang: Die Performanz von HAM-RPM: Zwei kommentierte Beispieldialoge und ein Interaktionsbeispiel. Juni 1978

5 Hoeppner, Wolfgang: NP-Referenzanalyse in HAM-RPM: Kommentierte Performanzbeispiele. Juli 1978

6 Demnick, Gudrun, Marburger, Heinz: Benutzeranleitung für das System HAM-RPM. Februar 1979

REFERENCES

Bates, M. (1978): The theory and practice of augmented transition network grammars. In: Bolc, L. (ed.): Natural language communication with computers. Berlin: Springer, 191-259

Bobrow, D.G., Kaplan, R.M., Kay, M., Norman, D.A., Thompson, H., Winograd, T. (1977): GUS - A frame-driven dialog system. In: Artificial Intelligence, 8, 1, 155-173

Bobrow, D.G., Winograd, T. (1977): An overview of KRL - a knowledge representation language. In: Cognitive Science, 1, 1, 3-46

Borgida, A.T. (1975): Topics in the understanding of English sentences by computer. Univ. of Toronto, Dept. of Computer Science, Technical Report No. 78

Charniak, E. (1973): Context and the reference problem. In: Rustin, R. (ed.): Natural language processing. Englewood Cliffs: Prentice-Hall, 311-329

Charniak, E. (1976): Inference and knowledge I, II. In: Charniak E., Wilks, Y. (eds.): Computational Semantics. Amsterdam: Noth-Holland, 1-21, 129-154

Deutsch, W., Clausing, H. (1979): Das Problem der Eindeutikeit sprachlicher Referenz. In: Ueckert, H., Rhenius, D. (eds.): Komplexe menschliche Informationsverarbeitung. Bern. Huber, 369-377

Gershman, A.V. (1977): Analysing English noun groups for their conceptual content. Yale Univ., Dept. of Computer Science, Research Report No. 110

Goldman, N.M. (1975): Conceptual generation. In: Schank, R.C. (eds.): Conceptual information processing. Amsterdam: North-Holland, 289-371

Grice, H.P. (1975): Logic and conversation. In: Cole, P., Morgan J.L. (eds.): Syntax and semantics, Vol. 3: Speech acts, N.Y.: , Academic, 41-58

Grosz, B.J. (1976a): Discourse analysis. In: Walker, D.E. (ed.): Speech understanding research. Final Technical Report, Stanford Research Institute, Chapter VIII

Grosz, B.J. (1976b): Resolving definite noun phrases. In: Walker, D.E. (ed.): Speech understanding research. Final Technical Report. Stanford Research Institute, Chapter IX

Grosz, B.J. (1976c): Ellipsis. In: Walker, D.E. (ed.): Speech understanding research, Final Technical Report, Stanford. Research Institute, Chapter X

Hayes, P.J., Rosner, M.A, (1976): ULLY: A program for handling conversations. In: Proc. of the AISB Summer Conference, Edinburgh 137-147

Hendrix, G.G. (1977): Human engineering for applied natural language processing. In: Fifth International Joint Conference on Artificial Intelligence, 183-191

Herrmann, T., Deutsch, W. (1976): Psychologie der Objektbenennung. Bern: Huber

Herrmann, T., Laucht, M. (1976): On multiple verbal codability of objects. In: Psychological Research, 38, 355-368

Hobbs, J.R. (1976): Pronoun resolution. City Univ. of New York, Dept. of Computer Science, Research Report No. 76-1

Lakoff, G. (1975): Hedges: A study in meaning criteria and the logic of fuzzy concepts. In: Hockney, D., Harper, W., Freed, B. (eds.): Contemporary research in philosophical logic and linguistic semantics. Dordrecht: Reidel, 221-272

LeFaivre, R.A. (1977): FUZZY reference manual. Rutgers Univ., Dept. of Computer Science, March 1977

LeFaivre, R.A. (1978): Rutgers/UCI LISP manual. Rutgers Univ. , Dept. of Computer Science, June 1978

McDermott, D. (1978): The last survey of representation of knowledge. In: Proc. of the AISB/GI Conference on Artificial Intelligence, Hamburg, 206-221

Morton, J. (1976): On recursive reference. In: Cognition, 4, IV, 309

Norman, D.A., Rumelhart, D.E. (1975): Explorations in cognition. San Francisco: Freeman

Oomen, I. (1977): Determination bei generischen, definiten und indefiniten Beschreibungen. Tübingen: Niemeyer

Ritchie, G.D. (1977): Computer modelling of English grammar. Univ. of Edinburgh, Dept. of Computer Science, Report CST-1-77

Schank, R.C. (ed.) (1975): Conceptual information processing. Amsterdam: North-Holland

Siklossy, L. (1976): Let's talk LISP. Englewood Cliffs: Prentice-Hall

Sridharan, N.S. (ed): AIMDS - User manual. Version 2. Rutgers Univ., Dept. of Computer Science, June 1978

Wahlster, W. (1977): Die Repräsentation von vagem Wissen in natürlichsprachlichen Systemen der Künstlichen Intelligenz. Univ. Hamburg, Fachbereich Informatik, Report IfI-HH-B-38/77

Waltz, D.L. (ed.) (1977): Natural language interfaces. In: SIGART Newsletter, 61, 16-65

Waltz, D.L. (1978): An English language question answering system for a large relational database. In: CACM, 21, 7, 526-539

Winograd, T. (1972): Understanding natural language. N.Y.: Academic

Winograd, T. (1977): A framework for understanding discourse. Stanford Univ., Memo-AIM- 297

Winston, P.H. (1977): Artificial Intelligence. Reading: Addison-Wesley

Wong, H.K.T. (1975): Generating English sentences from semantic network structures. Univ. of Toronto, Dept. of Computer Science, Technical Report No. 84

Woods, W.A et al. (1976): Speech understanding systems. Final Report, Vol. V, BBN Report No. 3438, Cambridge: Bolt, Beranek & Newman

Zadeh, L.A. (1974): The concept of a linguistic variable and its application to approximate reasoning. San Jose: IBM Research Report RJ1355

Zadeh, L.A. (1978): Fuzzy sets as a basis for a theory of possibility. In: Fuzzy Sets and Systems, 1, 3-28