In: Rieger, B. (ed.): Empirical Semantics. Bochum: Brockmeyer.

IMPLEMENTING FUZZINESS IN DIALOGUE SYSTEMS

Wolfgang Wahlster
Projektgruppe: Simulation von Sprachverstehen,
Germanisches Seminar der Universität Hamburg, Germany

A problem which must be dealt with by any reasonably general computer simulation of the cognitive processes underlying natural language dialogue behaviour is the inherent fuzziness of natural language and human thought. Artificial Intelligence (AI) research has to date achieved nothing approaching a general formal treatment covering all aspects of fuzziness. Surveys of AI approaches to the problems of vagueness can be found in WAHLSTER (1977) and PRADE (1979).

1. THE TREATMENT OF FUZZINESS IN AI SYSTEMS

In most of the natural language AI systems implemented to date, the problem of fuzziness is not taken into account. WINOGRAD, for example, in his pioneering work on the SHRDLU system, employs the following representation of the meaning of the word 'millionaire' (WINOGRAD, 1972: 148):

(THGOAL (#POSSESS $?X1 $1,000,000))

A person who possesses $999,999 is thus not acknowledged by the system to be a millionaire; by contrast, the definition underlying our everyday use of the word stipulates vaguely that a millionaire must possess at least approximately a million dollars.

Most of the contributions of AI research to the problems of vagueness have come from the areas problem solving and expert systems. Special inference mechanisms have been developed for what has variously been referred to as 'fuzzy reasoning', 'common-sense reasoning', 'plausible reasoning', 'approximate reasoning', 'probabilistic reasoning', 'speculative reasoning', and 'vague reasoning'.

The linguistic capabilities corresponding to these cognitive capabilities have, however, received very limited attention in such systems as MYCIN (SHORTLIFFE 1976) and PROSPECTOR (HART/ DUDA 1977). In MYCIN, for instance, the uncertainty involved in the various steps of the diagnosis is skillfully taken into account by the model which underlies the system's reasoning. But this uncertainty is reflected linguistically only in the way in which conclusions are qualified, with formulations such as 'There is strongly suggestive evidence that... ' and 'There is weakly suggestive evidence that...' (SHORTLIFFE, 1976: 99).

From the beginning of our work on the dialogue system HAM-RPM, which converses with a human partner in colloquial German about limited, but interchangeable scenes, we have attempted to deal with vagueness (WAHLSTER/v. HAHN 1976) on three levels, giving each level approximately equal emphasis: Vague utterances are analysed and generated; the intervening reasoning is often fuzzy, as is the data on which it operates.

The LISP-embedded PLANNER-type programming language FUZZY (LeFAIVRE 1977) has proved to be a powerful implementation tool for programs which require facilities for the concise representation and high-level but efficient manipulation of fuzzy knowledge.

Figure 1 indicates some components of a dialogue system in which fuzziness can play a role. All of the processes and representation structures mentioned in Figure 1 have been realized to a certain degree in HAM-RPM, in some cases only in an extremely simplified form.

In the second part of this paper, examples exhibiting the corresponding capabilities of HAM-RPM will be discussed. In the third part of the paper, we shall present a new corroboration procedure for multiple derivations within the framework of a many-sorted fuzzy logic, define the associated derivation rules, and give an example of a possible application of this calculus in a natural language dialogue system.

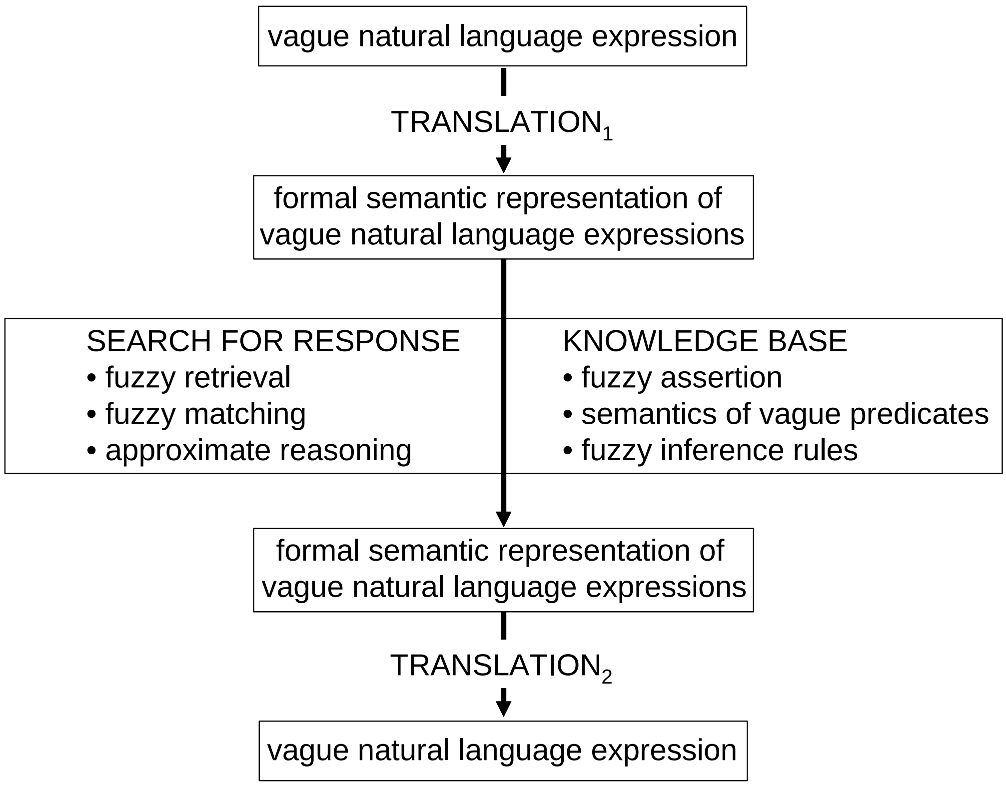

Fig. 1: Fuzziness in various components of a dialogue system

An informal discussion of some of the communicative and cognitive functions of vagueness will provide a framework within which the capabilities of HAM-RPM to be discussed in the next part of the paper can be evaluated appropriately. In particular, it will become clear that, although fuzziness is reflected more extensively in HAM-RPM than in most comparable AI systems, some of the most important functions of vagueness are neglected in the present version of the system.

1.1. SOME COMMUNICATIVE AND COGNITIVE FUNCTIONS OF VAGUENESS

If we assume the notion of the preciseness of an expression to be a relative concept, we can say that the expression A 'John earns about $30,000 a year' is more precise than the expression B 'John earns a lot of money'. Similarly, if we consider the inverse relation 'vaguer than', we can say that B is vaguer than A. According to EIKMEYER and RIESER (1978: 25), we may call a given expression X precise if and only if there exists no expression Y which is more precise than X. A consequence of this definition is that there are many expressions for which no corresponding precise expression exists (cf. WAHLSTER, 1977: 34). For A and B, however, the expression 'John makes $31,263,75 a year' would be considered a corresponding precise expression.

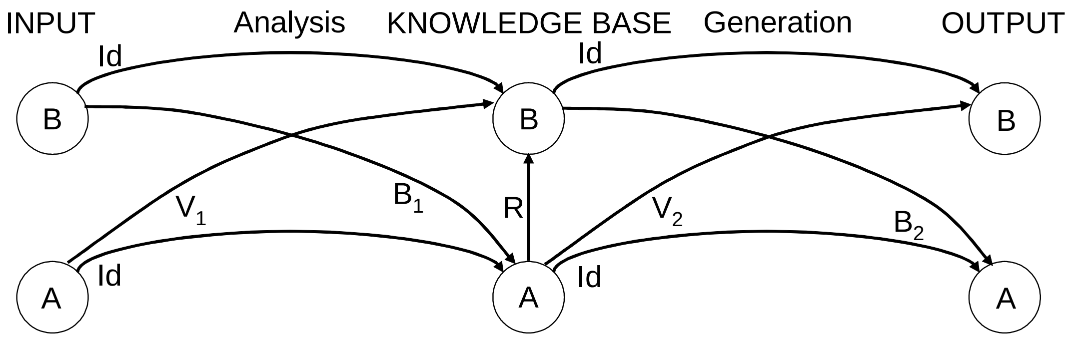

Fig. 2: Translations which change the degree of precision

Figure 2 shows some of the translations between the expressions A and B that might occur during the analysis and generation of utterances. Id is the identity mapping. If only Id is used during analysis, the information input is stored without alteration in the knowledge base. If Id is used during generation, the information returned by the answering routines is output without any change in its degree of fuzziness. This is the only sort of behaviour which is possible for traditional information systems.

By contrast, human speakers typically recode information in a variety of ways during the analysis and generation of utterances, some of these transformations involving changes in the degree of fuzziness.

During analysis, relatively precise statements are often translated into less precise ones which are easier to remember (V_1 in Figure 2). Such translations depend heavily on standards which may vary from person to person. For example, if the sentence 'It was 11:45 P.M. and he was driving 85mph' is read within a long text, it is likely to be encoded as 'It was late at night, and he was driving very fast'.

The arrow R in Figure 2 represents a kind of 'intelligent garbage collecting', which though potentially useful has not, to my knowledge, appeared in any AI program.

The translation of a relatively vague to a relatively precise assertion (B_1 in Figure 2) with the help of general inference rules is also possible. Default reasoning (REITER 1978) provides an example of this sort of reasoning. Such translations, which involve a high degree of uncertainty, are necessary when inferences presuppose a certain degree of precision. For example, if I wish to estimate the annual income of the married couple John and Mary, and I have the precise figure of $18,203.50 a year for Mary, but only B for John, I must use my general knowledge to translate B into something like A, so as to be able to arrive at the estimate of 'roughly $50,000 a year' for the two together.

When utterances are generated, relatively precise stored values are frequently mapped onto less precise values (V_2 in Figure 2). The function of this sort of transformation can be understood in the light of GRICE's (1975) conversational maxim 'Don't make your contribution more informative than is required'.

V_2 also plays a key role in intentional exaggeration and understatement, which usually require the use of vague terms. This particular use of vagueness can only be meaningfully simulated if the computer program assumes the role of a dialogue partner whose goals are different from the unqualified desire to be helpful which guides the behaviour of most dialogue systems including the present version of HAM-RPM. In the research group 'Simulation of Language Understanding' we are at present investigating this function of vagueness using the example of a room-booking dialogue: The program simulates a hotel manager who is anxious to rent one of his rooms. If the 'caller' asks how large the room is, and the manager knows perfectly well that the dimensions are 9 by 15 feet, he may prefer to express himself vaguely and exaggerate somewhat, saying that the room is 'quite large'. One reason why vague expressions are more suitable for this sort of thing than more precise ones e.g. 'The room is 11 by 18 feet', is that their use is less likely to result in negative sanctions: if challenged, the speaker can, in effect, appeal to individual differences in standards ('I consider this room to be quite large').

The arrow B_2 in Figure 2 represents a translation for output of a relatively vague assertion into a more precise one, often on the basis of more general beliefs. One use of such translations is to lend added weight to claims by giving the impression that the speaker has more detailed knowledge of his subject than he actually has: 'This pain-killer is 34.6% more effective than its rival'.

Since the ultimate goal of AI research in natural language understanding is a computer model of all of the cognitive processes which underlie intelligent dialogue behaviour, all of the mappings indicated in Figure 2 will have to be dealt with. Before models which can do justice to the various functions of these mappings can be constructed, much research will have to be done into the computer modelling of goal-driven understanding, memory organization, and speech-act planning.

1.2. PUTTING VAGUENESS TO PRACTICAL USE

On the other hand, some of the techniques which are already implemented in HAM-RPM to handle the analysis and generation of vague utterances and to represent vague knowledge could be put to practical use: Their incorporation in such application-oriented natural language interfaces as those described in WALTZ (1977) would increase the flexibility of these systems and their convenience for users. Consider the following five examples of possible applications:

- For reasons of data security the system could generate answers with varying degrees of precision, dependent on the status of the user.
- When a precise answer would require too much computing time, the system could quickly produce a vague answer.
- The user could include vague expressions in his requests; the system could interpret them in terms of the precise values in the data base.
- When the system lacked the information required to produce a precise answer, it might be able to arrive at a conjecture on the basis of fuzzy reasoning; this would at least be more helpful than the response 'I don't know'.
- The system could use vague expressions to keep its own utterances free of superfluous information.

These introductory remarks make it clear that vagueness must not be viewed as a defect of colloquial language which is - and should be - absent in formal query languages; on the contrary, vagueness can significantly increase efficiency of communication between man and machine.

2. SOME REFLECTIONS OF FUZZINESS IN HAM-RPM

The representation of fuzzy knowledge in HAM-RPM (a detailed description of HAM-RPM's knowledge base can be found in v. HAHN et al. 1980) relies heavily on FUZZY's assertional data base, whose entries have the form

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

The left-hand side is an arbitrary list structure. In (1) it is interpreted by HAM-RPM as ascribing sweetness to apple juice. The number on the right is the 'z-value': each assertion in the data base is modified by a z-value, usually in the range $[0, 1] \subset |R$. The z-values are interpreted by HAM-RPM in various ways, depending on the context and the processes which make use of the assertion in question. For (1), the interpretation is in terms of frequency: apple juice is usually sweet.

FUZZY's assertional data base is used to represent HAM-RPM's two semantic networks. The referential semantic network, which contains extensional information about the scene being discussed, is complemented by the conceptual semantic network, which contains intensional information relevant to the scene.

All of the arcs in the two semantic networks can be fuzzified. For example, the following two assertions from the conceptual semantic network represent the knowledge that a tree usually has branches, and that, more often than not, it has many branches:

(2) ((T TREE BRANCH). 0.8)

(3) ((NUMBER (T TREE BRANCH) MANY). 0.6)

The T-arc links part-concepts to whole-concepts, while the NUMBER-arc is used to represent the number of parts of a particular sort associated with a whole-concept. As can be seen in (3), fuzzy natural language quantifiers (cf. ZADEH 1978) like 'many' are sometimes used directly in the representation language.

2.1. LINGUISTIC HEDGES AND FUZZY SPATIAL RELATIONS

The present version of HAM-RPM uses procedural definitions of 23 linguistic hedges. These hedges may either be recognized within the input and mapped onto assertions in the knowledge base ($TRANSLATION_1$ in Figure 1), or used by the system to qualify its own utterances (TRANSLATION in Figure 1). The following exerpts from actual dialogues conducted with HAM-RPM will illustrate ways of dealing with these linguistic hedges.

Here are some examples of the generation of hedges:

(4) WIEVIELE BEINE HAT EIN MENSCH?
How many legs does a human being have?

FAST IMMER ZWEI.
Almost always two.

(5) UND EIN STUHL? WIEVIEL BEINE HAT DER?
And a chair? How many legs does it have?

NORMALERWEISE VIER.
Usually four.

(6) UND DIESER KAPUTTE STUHL?
and this damaged chair here?

NUR DREI.
Only three.

The information requested in (4) and (5) is obtained from the conceptual semantic network, where it is either retrieved directly or derived with the help of general rules. The z-values which modify the answer are then mapped onto appropriate linguistic hedges.

In the case of (6), it is the referential semantic network which contains the desired information. Since the value found there differs from the standard value stored in the conceptual semantic network, the adverb 'only' is used to modify the answer (The algorithms underlying this behaviour were designed by W. HOEPPNER).

Another class of adverbs, including words like 'directly' and 'roughly', is used within HAM-RPM to fuzzify or defuzzify expressions used to describe spatial relations.

(7) STEHT DAS FERNSEHGERÄT EIGENTLICH DIREKT LINKS NEBEN DER LAMPE, WELCHE SICH RECHTS NEBEN IHM BEFINDET?
Is the TV set in fact standing directly to the left of the lamp which is to its right?

When the question (7) is analysed, the vague spatial relation 'to the right of is used to determine which lamp is meant. Then the positions of the TV set and the lamp are compared to determine whether they satisfy the relation 'directly to the left of'. Note that the hedge 'directly' increases the precision of this expression, in the sense that the range of possible configurations of the two objects is narrower (The spatial-relations component of HAM-RPM was designed by A. JAMESON).

The system also uses hedges to modify spatial relations. The following noun phrase identifies the object 'ARMCHAIR4' uniquely within one of HAM-RPM's domains, a living-room scene:

(8) DER HINTERE, BRAUNE SESSEL, DER SICH UNGEFÄHR RECHTS VON DEM MITTLEREN ROTEN BILD BEFINDET.
The rear brown armchair which is located roughly to the right of the middle red picture.

In this example the spatial relations 'rear' and 'middle' are used to distinguish the two objects from other objects in their immediate neighbourhoods which have exactly the same properties (cf. WAHLSTER et al. 1979): there is a group of brown armchairs and a group of red pictures.

The translation of z-values into linguistic hedges in HAM-RPM is context-dependent, the following two factors being taken into account:

- The nature of the predicate modified by the z-value
- The knowledge source which supplied the information expressed in the modified predication

Consideration of the first factor is necessary to ensure, for example, that (BEHIND 0.2) is verbalised as 'roughly behind', but (DAMAGED 0.2) as 'slightly damaged'.

Here is an example of the influence of the second factor: (EXPENSIVE 0.6) is expressed either as 'rather expensive' or 'often expensive', depending on whether the corresponding assertion was retrieved from the referential or from the conceptual semantic network.

2.2. REPRESENTING THE REFERENTIAL MEANING OF RELATIVE ADJECTIVES

The use of a relative adjective such as 'old', 'big' or 'narrow' always involves a comparison with a standard. The particular standard used depends on the type of object being described and on other contextual factors. The applicability of such predicates is typically a matter of degree.

The referential meaning of 'big' for the reference set 'trees' is encoded in HAM-RPM in the following FUZZY procedure, which is invoked when the system needs to determine whether a particular tree is big or not.

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

Suppose that the assertion (10), which states that the height of TREE3 is about 15 meters, is stored in the referential semantic network.

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

When called upon to answer the question about this tree

(12) IST DER BAUM, DER RECHTS NEBEN DER AMPEL STEHT, GROSS?
Is the tree which is next to the traffic light big?

the system will attempt with the help of the procedure (9), to derive the assertion (REF TREE3 BIG) with a z-value of at least 0.2. The variable _>TREE in (9a) is implicitly typed, so that it can only be bound to names of objects of the type TREE (typed variables will be discussed in more detail in the next section). Line (9b) calls for the retrieval of the assertion (10) from the referential semantic network. Line (9c) provides for the 'bigness' of TREE3 to be computed using the standard function SFUNK (introduced by Zadeh). The standard size of trees is encoded in the second and third arguments to SFUNK.

The invocation of (9) with the argument TREE3 thus yields the assertion ((REF TREE3 BIG) . 0.5), which permits the question (12) to be answered in the affirmative.

There are several other modules of HAM-RPM in which fuzziness plays an important role, but this section will conclude with a brief mention of only one of them: The set of all previously mentioned objects which are considered to be of current relevance in the dialogue is represented as a fuzzy set. The degree of current relevance of an object is taken into account by the system in several contexts. For example, when the dialogue partner uses a pronoun which has more than one possible antecedent, it is sometimes safe to assume that the object with the greater current relevance was the one he had in mind.

3. A NEW APPROACH TO APPROXIMATE REASONING IN A MANY-SORTED FUZZY LOGIC

It is often easy to modify non-fuzzy representation structures and algorithms so that they can handle degrees of variation. The more interesting problems of fuzziness, on the other hand, require a more radical treatment, including the development of new algorithms and control structures.

The new approach to the formal reconstruction of approximate reasoning processes to be described in this section is an example of such a treatment. The following four properties characterize our model of fuzzy inference processes:

- A fuzzy inference rule represents a weak implication; a particular 'implication strength' must thus be associated with each such rule.

- The premisses of a fuzzy inference rule are often fulfilled only to a certain degree.
- The applicability of a fuzzy inference rule in the derivation of a particular conclusion is likewise a matter of degree.
- Several mutually independent fuzzy inference rules can corroborate each other in the derivation of a particular conclusion.

Before these properties can be elucidated with the help of examples, the concepts 'implicitly typed variable' and 'fuzzy matching' must be defined.

3.1. FUZZILY SORTED VARIABLES AND FUZZY MATCHING

Pattern matching usually results in the binding of certain variables which appear in the pattern. The evaluation of (13), for example, causes the variable LUXURY/ CAR to be assigned the value VW/ BUG1. Simple shove variables are

(13) (MATCH (?LUXURY/ CAR IS EXPENSIVE) (VW/ BUG1 IS EXPENSIVE))

prefixed in FUZZY with a question mark. In (13) the variable name 'LUXURY/ CAR' is arbitrary, in no way restricting the class of values that may be assigned to the variable. If we want the match in (13) to fail on the grounds that a VW bug is not a luxury car, we must make LUXURY/ CAR into a typed variable. (What are called 'types' in computer science are not types in the sense of mathematical logic, but only sorts. These two terms will be used interchangeable in the following).

The relation ISA is used to introduce sort identifiers:

(14) ((ISA CADILLAC123 LUXURY/ CAR). 1)

This assertion assigns the individual constant CADILLAC123 to the sort 'LUXURY/ CAR'. We have extended FUZZY's pattern matcher to permit the representation of typed variables as follows: If the name of a sort such as 'LUXURY/ CAR' is prefixed with the characters '_>', it can be bound only to the individual constants of the corresponding sort. The match attempted in (15) will thus be unsuccessful, providing there is no assertion in the data base assigning VW/ BUG1 to the sort LUXURY/ CAR.

(15) (MATCH (_>LUXURY/ CAR IS EXPENSIVE) (VW/ BUG1 IS EXPENSIVE))

This kind of variable-typing can be extended if relations between sorts are declared by means of assertions such as (16). In (16), 'AUTOMOBILE' is introduced as a superset of LUXURY/ CAR.

(16) ((U AUTOMOBILE LUXURY/ CAR). 1)

Such declarations create a hierarchy of types, and type-checking can involve a search through the hierarchy of types as well as direct retrieval. In (17), for example, the variable AUTOMOBILE is bound to

(17) (MATCH (_>AUTOMOBILE CONSUMES GASOLINE) (CADILLAC123
CONSUMES GASOLINE))

CADILLAC123, because the system can use the assertions (14) and (16) to prove that CADILLAC123 belongs to the sort 'AUTOMOBILE'.

In the examples discussed so far, only one typed variable corresponding to a particular type has been used, but it is obvious that, in general, an arbitrary number of variables of a given type must be available. One way to achieve this is to allow integers to serve as suffixes in the names of typed variables, as in (18) (A more general approach to typed variables has been developed by BOLEY (1978) for the language FIT).

(18) (_>AUT0MOBILE1 IS MORE EXPENSIVE THAN _>AUTOMOBILE2)

If we introduce fuzzy type declarations such as (19) it becomes evident that some sort of fuzzy matching is needed. The

(19) ((ISA BUICK321 LUXURY/ CAR). 0.7)

corresponding procedure - here called FMATCH - should return a value of 0.7 when (20) is executed, indicating that the match

(20) (FMATCH (_>LUXURY/ CAR IS EXPENSIVE) (BUICK321 IS EXPENSIVE))

was successful only to a certain degree. Since a matching operation in which the pattern consists of several elements can be viewed as a conjunction of simpler matching operations the z-value returned by the more complex operation should be equal to the minimum of the z-value returned by the simpler operations.

3.2. TWO DERIVATION RULES AND A CORROBORATION PROCEDURE

Although a complete definition of the syntax of the many-sorted fuzzy logic used in this section is not possible here, the two derivation rules of the calculus will now be defined formally.

Let $S = \{s_1, s_2, ..., s_n\}$ be a set of sort identifiers, and FS_O the set of individual constants. We introduce a fuzzy sorting by defining for each $s \in S$ a characteristic function:

$$\delta_S : FS_O \to [0,1] \subset \mathbb{R} \quad \text{and setting} \quad FS_O^S = \{a \in FS_O \mid \delta_S(a) > 0\}$$

Lower case t will be used to refer to terms, and v^{s_i} will stand for the individual variables belonging to the sort s_i. Let FO be the set of well-formed formulas and A and B be elements of FO.

We shall continue to use the neutral term 'z-value' to refer to the value $ZVAL(A)$ from the real interval $[0,1]$ associated with the formula *A*; epistemologically, this value can be interpreted as a fuzzy truth value, a possibility (ZADEH 1978), a measure of belief (SHORTLIFFE 1976), or an agreement probability (SCHEFE 1979), to mention but a few of the possible interpretations.
Since only a finite subset of the real numbers can be represented on a computer in any case, it is tempting to embed the calculus in a many-valued logic with a finite truth set. This would make it possible to apply the completeness and soundness results of MORGAN (1976) with only minor modifications. How can new formulas be derived from given formulas, and how are z-values to be assigned to the new formulas? The set of derivation rules $R = \{R_1, R_2\}$ is defined as follows:

(21) $$R_1 : \quad FO \times FO \times FO \to \{0,1\}$$

$$R_1(M,N,O) = 1 :\Leftrightarrow M = A,\ N = A \Rightarrow B,\ O = B$$

$$\text{where } ZVAL(O) = ZVAL(M) \cdot ZVAL(N)$$

R_1 is called fuzzy modus ponens.

(22) $$R_2 : \quad FO \times FO \to \{0,1\}$$
Let φ be a substitution $\{t_1/v^{s_1}, \ldots, t_n/v^{s_n}\}$

$$R_2(M,N) = 1 :\Leftrightarrow M = A,\ N = \varphi A$$

$$\text{where } ZVAL(N) = ZVAL(M) \cdot MIN(\omega_{s_1}(t_1), \ldots, \omega_{s_n}(t_n))$$

$$\text{where } \omega_{s_i}(t_i) = \begin{cases} 1 & \text{if } t_i \notin FS_O \\ \delta_{s_i}(t_i) & \text{if } t_i \in FS_O \end{cases}$$

R_2 is called the fuzzy

subsitution rule; it was implemented in FUZZY by means of a change in the system procedure which controls the matching of calling patterns with the characteristic patterns of FUZZY procedures.

The above derivation rules sometimes permit two derivations of the same formula B, using different axioms in the knowledge base. It remains to be specified how the z-values corresponding to the several derivations are to be combined to yield an overall z-value.

Suppose there are k implications $A_1 \overset{a_1}{\Rightarrow} B, \ldots, A_k \overset{a_k}{\Rightarrow} B$ where each $a_i \in [0,1]$ denotes the corresponding implication strength. *ZVAL(B)* is computed via the following recursive formula, which is a generalization of Shortliffe's combining function 1 (SHORTLIFFE, 1976: 179):

(23) $$ZVAL(B) = \sum_{v=1}^{k} (-1)^{v+1} \sum_{\{i_1, i_2, \ldots, i_v\} \subset \{1,2,\ldots,k\}} \prod_{j=1}^{v} ZVAL(A_{i_j}) \cdot \alpha_{i_j}$$

While it would exceed the scope of this paper to justify (21), (22), and (23) and to compare them with the corresponding derivation rules and combining functions of other calculi, an extended example of their use should suffice to give the reader an intuitive appreciation of the model as a whole. An example was selected which will illustrate the four properties of the model mentioned above.

3.3. MULTIPLE DERIVATIONS: AN EXAMPLE

First suppose that the following two assertions, which assign the individual constant MERCEDES123 to the sorts 'LUXURY/ CAR' and 'AUTOMOBILE' respectively, are stored in the data base:

((ISA MERCEDES123 LUXURY/ CAR) . 0.8)
((ISA MERCEDES123 AUTOMOBILE) . 1)

The data base also contains the axioms A, B, and C, which represent the fuzzy knowledge that the Mercedes in question is used, rather old, and somewhat rusty.

A = ((REF MERCEDES123 USED). 1)
B = ((REF MERCEDES123 OLD). 0.5)
C = ((REF MERCEDES123 RUSTY). 0.3)

Two other axioms that the data base contains are the inference rules D and E, which represent common-sense knowledge: D states that an old, used luxury car is likely to be cheap, while E makes the same prediction for any rusty automobile.

```
D =  (PROC DEMON: MDEMON ZVAL: '(0.4 0.7)
     (REF _>LUXURY/ CAR CHEAP)
     (GOAL (REF !LUXURY/ CAR OLD))
     (GOAL (REF !LUXURY/ CAR USED)) (END?))

E =  (PROC DEMON: MDEMON ZVAL: '(0.2 0.8)
     (REF _>AUTOMOBILE CHEAP)
     (GOAL (!AUTOMOBILE RUSTY))
     (END?))
```

MDEMON is a control procedure specially designed for multiple derivations. It supervises the evaluation of the FUZZY procedures to which it is assigned, combining the z-values resulting from the various derivations in accordance with (23). The second of the pair of numbers occurring in the first line of D and E represents the implication strength of the inference rule. The implication strength of D is thus .7, that of E .8.

The value of the two implicitly typed variables _>LUXURY/ CAR and _>AUTOMOBILE are referenced via a prefix '!'. We shall now use the derivation rules and the corroboration formula defined above to prove the following assertion:
Assertion: {A, B, C, D, E} $\vdash$ ((REF MERCEDES 123 CHEAP). 0.4528)

Proof: A) We first use R_2, where φ_1 = {MERCEDES123/_>LUXURY/ CAR}

$\varphi_1 D$ = (PROC DEMON: MDEMON ZVAL: ' (0.4 0.56 [= 0.7·0.8])
(REF MERCEDES123 CHEAP)
(GOAL (REF MERCEDES123 OLD))
(GOAL (REF MERCEDES123 USED))
(END?))

Because of the fuzzy pattern matching which is performed when D is invoked, the implication strength of $\varphi_1 D$ is weaker than that of D. This corresponds to the fact that the inference rule D is valid only for luxury cars, and MERCEDES123 can only be regarded as a luxury car to a certain degree. This illustrates the third of the four properties of the model listed above.

B) We apply R_1 and obtain $\{A, B, \varphi_1 D\} \vdash$ F

F = ((REF MERCEDES123 CHEAP) . 0.28 [= MIN(1, 0.5) · 0.56])

This illustrates the second property, since the premise that the car is old is only satisfied to a certain degree. It also illustrates the first property, as $\varphi_1 D$ is a weak inference rule.

C) The step B) completes one derivation of the formula which was to be proven. We now seek a second derivation in order to corroborate the results of the first one. We first apply R_2 once again, where $\varphi 2$ = {MERCEDES123/_>AUTOMOBILE}

$\varphi_2 E$ = (PROC DEMON: MDEMON ZVAL: '(0.2 0.8 [=1 · 0.8])
(REF MERCEDES 123 CHEAP)
(GOAL (REF MERCEDES123 RUSTY))
(END?))

D) Applying R_1 we then obtain $\{C, \varphi_2 E\} \vdash$ G
G = ((REF MERCEDES123 CHEAP). 0.24 [= 0.3·0.8])

E) The combining function (23) now yields {F, G} $\vdash$ H
H = ((REF MERCEDES123 CHEAP). 0.4528 [= 0.28 + 0.24 - 0.28 · 0.24])

This last step illustrates the fourth characteristic property of the model: two independent pieces of evidence combine to yield the desired z-value.
The reconstruction of fuzzy inference processes described in this section has been implemented in a specially extended version of FUZZY. Its performance on a number of test examples has been satisfactory.

In conclusion, it should be noted that the problem of incorporating such a model of fuzzy reasoning into a natural language dialogue system - such as HAM-RPM - is more difficult than it might appear to be at first glance. If the system draws unreliable conclusions such as the one in the above example while answering a question like 'Is that Mercedes cheap?', it is not enough for it to hedge its affirmative answer with an expression like 'I think so'. In many cases a dialogue partner will not be satisfied with such an answer unless the system can explain how it was derived ('Because it is used and rather old and ...'). One of the things that complicate the task of simulating such explanation of reasoning is the fact that the fuzziness enters into the reasoning in several different ways (cf. the four characteristic properties of the present model listed above, some of which are harder to verbalise colloquially than others).

The explanation module which has been implemented within HAM-RPM (WAHLSTER et al. 19 79) represents an attempt to deal with some of these problems.

ACKNOWLEDGEMENTS

I wish to thank Anthony Jameson for suggesting a number of improvements to an earlier draft of this paper. Research on the dialogue system HAM-RPM is currently being supported by the Deutsche Forschungsgemeinschaft.

REFERENCES

BOLEY, H. (1978): Five views of FIT programming. Universität Hamburg, Fachbereich für Informatik. (revised September 1979).

EIKMEYER, H.-J., RIESER, H. (1978): Vagheitstheorie. Universität Bielefeld, Projekt Vagheitstheorie.

HAHN V., W., HOEPPNER, W., JAMESON, A., WAHLSTER, W. (1980): The anatomy of the natural language dialogue system HAM-RPM. to appear in: BOLC, L. (Ed.) Natural Language Based Computer Systems. München: Hanser/Macmillan, 119-253.

HART, P.E., DUDA, R.D. (1977): PROSPECTOR - A computer-based consultation system for mineral exploration. SRI International, AI Center, Menlo Park, Technical note 155.

LeFAIVRE, R. (1977): FUZZY reference manual. Rutgers University. Computer Science Department. March 1977. (revised June 1978).

MORGAN, C.G. (1976): A resolution principle for a class of many-valued logics. Logique et Analyse, 19, 311-339.

PRADE, H.M. (1979): The application of fuzzy set theory to high-level artificial intelligence - survey and future developments . Université Paul Sabatier, Toulouse.

REITER, R. (1978): On reasoning by default. American Journal of Computational Linguistics, Microfiche 80.

SCHEFE, P. (1979): Foundations of reasoning with uncertain facts and vague concepts. Universität Hamburg, Fachbereich Informatik, Report IFI-HH-B-56-79 (also to appear in: Int. J. Man-Machine Studies, 1980).

SHORTLIFFE, E.H. (1976): Computer-based medical consultations: MYCIN. New York: Elsevier.

WAHLSTER, W. (1977): Die Repräsentation von vagem Wissen in natürlichsprachlichen Systemen der Künstlichen Intelligenz. Universität Hamburg, Fachbereich Informatik, Report IFI-HH-B-38/77.

WAHLSTER, W., v. HAHN, W. (1976): Einige. Erweiterungen des natürlichsprachlichen AI-Systems HAM-RPM. Laubsch, J.H., Schneider, H.-J. (Eds.) Dialoge in natürlicher Sprache und Darstellung von Wissen. Freudenstadt, 204-225.

WAHLSTER, W., JAMESON, A., HOEPPNER, W. (1979): Glancing, referring and explaining in the dialogue system HAM-RPM. American Journal of Computational Linguistics, Microfiche 77.

WALTZ, D.L. (1977): Natural language interfaces. SIGART Newsletter, 61, 16-65.

WINOGRAD, T. (1972): Understanding natural language. New York: Academic.

ZADEH, L.A. (1978): PRUF - A language for the representation of meaning in natural languages. Int. J. Man-Machine Studies, 10, 395-460.