L-FUZZY - AN AI LANGUAGE WITH LINGUISTIC MODIFICATION OF PATTERNS

by

Christian Freksa

Memorandum No. UCB/ERL M80/10

25 March 1980

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

ABSTRACT

L-FUZZY is a dialect of the AI-language FUZZY (LeFaivre, 1974). In this dialect, patterns are modified by linguistic labels (or "L-values") instead of numerical Z-values. This allows for representation of inherently imprecise information describing a fuzzy range of possibilities instead of a precise fuzzy set membership value. For example, the English statement "John is very tall" can be represented in L-FUZZY by

((JOHN IS TALL) very)

where the L-value "very" restricts the possibilities of tallness values that may correspond to John's height. Internally, L-values are represented by procedures which modify possibility distributions (Zadeh, 1978). Thus, given the possibility distribution representing the fuzzy set "tall" (i.e., an indication of the possibility that a person may have a particular height given that he is tall) the modifier "very" yields a possibility distribution indicating that a person may have a particular height given that he is "very tall".

For information retrieval, fuzzy assertions in the data base are matched against fuzzy requests. Three types of matching can be distinguished: trivial matching, relational qualitative matching, and detailed semantic matching. Which type is selected by the system depends on whether the request pattern is identical to, closely related to, or remotely related to the assertion pattern, respectively. Depending on the outcome of the match, ten different answers are possible: "absolutely", "indeed", "yes", "quite possibly", "possibly", "not quite", "no", "not at all", "on the contrary", "don't know".

A system for object identification by description matching is being implemented in L-FUZZY on PDP-10 and DECsystem20 computers.

---

* Supported by NESC Contract N00039-78-C-0013, NSF Grant ENG 78-23143 and by German Academic Exchange Service (DAAD).

L-FUZZY
- an AI language with linguistic modification of patterns *

Christian Freksa
Department of Electrical Engineering & Computer Sciences
and Electronics Research Laboratory
University of California
Berkeley

I. INTRODUCTION

In 1974, the programming language FUZZY by Rick LeFaivre became available [9,10]. This language combines many useful features of other 2nd generation AI-languages like MICRO-PLANNER, CONNIVER, and QLISP [2,6]. The most important features are a system of associative nets, pattern matching mechanisms, pattern-directed procedure invocation, global control by procedure demons, controlled backtracking. In addition, FUZZY provides facilities for efficient storage, retrieval and manipulation of information which is imprecise or uncertain. In particular, variables and procedures may return both a value and a so-called Z-value. The Z-value - typically a real number in the interval [0,1] - modifies the associated value and may be degree of set membership in a fuzzy set, a certainty value, a possibility or probability measure, or an indication of confidence in a decision. All system functions know about Z-values and use them for local or global control purposes.

FUZZY is directly embedded in LISP and therefore much more efficient than languages like MICRO-PLANNER which require a run-time monitor. LISP and FUZZY primitives may be freely intermixed and FUZZY functions may be called from compiled LISP code. FUZZY has been successfully used in two major projects: for the AIMDS system at Rutgers University [13] and for the HAM-RPM system at Hamburg [4].

A problem with FUZZY is, however, that there is no natural way of representing modified fuzzy sets. For example, if the Z-value is interpreted as a fuzzy set membership value, a fuzzy statement modified by a Z-value represents a (crisp) element of a fuzzy set; if the Z-value is interpreted as a threshold value, we can represent fuzzy sets in which low-membership elements have been eliminated. In those situations in which we do not have or do not need precise information it is desirable to have a way of directly representing general modified fuzzy sets. Typical modifications of fuzzy sets are shift, precisiation, or fuzzification.

L-FUZZY is an extension of FUZZY, in which numerical Z-values are replaced by linguistic modifiers ("L-values") [3,15,17]. For example, the English statement "John is very tall" may be viewed as a fuzzy assertion "John is tall" modified by the linguistic qualification "very" and represented in L-FUZZY by

((JOHN IS TALL) very)

The implementation of linguistic modifiers is based on a model in which a fuzzy assertion is viewed as descriptor of a set of possibilities [18,19]. This set is modified by L-values, which are internally represented by procedures.

II. FUZZY STATEMENTS AND LINGUISTIC MODIFIERS

II.1 Labels of fuzzy concepts.

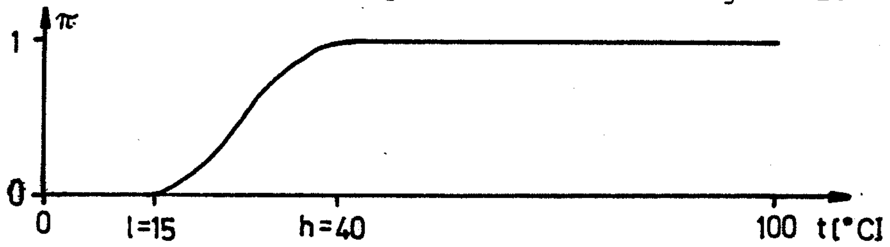
Consider the English statement "the water is warm". What does this mean? It is a statement describing the temperature of some mass of water. The label "warm" indicates, in the given context

1. a feature dimension (in this case: "temperature")
2. a range of feature values (the range of temperatures at which water is considered "warm")
3. a relative property (i.e., a property which can be emphasized or deemphasized)

The statement is not precise, because it does not specify the temperature of the water; but the statement restricts the possibilities of temperatures that the water may have. In any given context, there is no particular temperature above which water is considered "warm" and below which it is considered "not warm". However, some temperatures definitely can be called "warm", others definitely can be called "not warm". In between, there is a gradual transition and at any given temperature point one of the two labels may be more appropriate than the other. For this reason, we call the restriction on the possibilities of the water temperature imposed by the label "warm" a fuzzy restriction.

This can be illustrated by a possibility distribution over the range of water temperatures: if no situation-specific information about water temperature is given, all temperatures between 0'C and 100'C are equally possible: i.e., the possibility may be said to be of value one over the entire range [0,100] 'C. If we assume a context such that water is definitely considered "warm" at temperatures above 40'C and definitely not considered "warm" at temperatures below 15'C, we obtain a possibility restriction which may be represented as in figure 1:

Fig. 1



In our model, we want to say little about the transition of the possibility function from one to zero. It is a monotonic transition but we will not worry about its specific shape; we will use a standard function* to describe the transition. For information retrieval, only relative possibility values are necessary. These are independent of the particular monotonic function that is used.

II.2 Modifiers of fuzzy concepts.

Linguistic modifiers in English may have various effects [5,7,8]. In the following discussion we will consider two cases:

1. precisiation or fuzzification of statements

2. alteration of statements

For clarification of the two cases consider the following situations:

1. Lisa intends to go for a swim in the lake, but before jumping into the water she asks her friend "is the water warm?" Her friend answers "indeed, the water is very warm!"

2. Lisa asks "is the water cold, warm, or very warm?" and her friend mumbles "it is very warm." Lisa does not quite understand the response and asks again

---

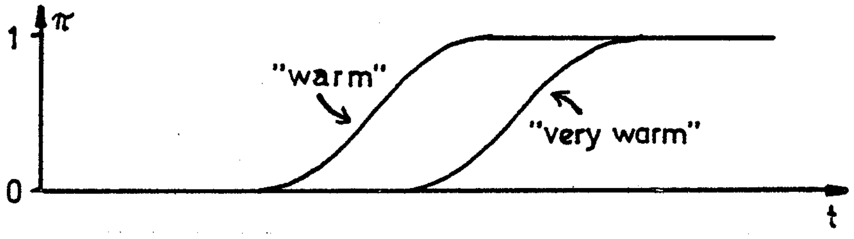
* S-function (compare [16]):

```
  S (t) = 2(t - l)**2        for t between l and (h + l)/2
  S (t) = 1 - 2(h - t)**2    for t between (h + l)/2 and h
```

"you say it is warm?" Her friend answers "no, it is very warm!"

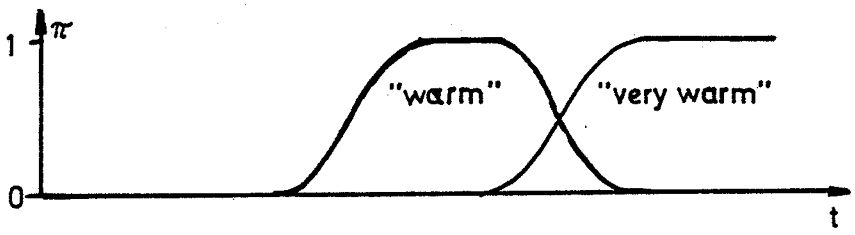
In both situations, "warm" is used to label a fuzzy range of water temperatures. In the first situation, however, "very warm" denotes a fuzzy sub-range of the range "warm", thus it restricts the possibilities of the actual temperature that the water may have. In other words, "the water is very warm" is a more precise (or less fuzzy) statement than "the water is warm". This situation is depicted in figure 2:

Fig.2



In contrast, in the second situation, "very warm" denotes a range of higher water temperatures than "warm". The modifier "very" alters the statement but does not necessarily make it more precise. Figure 3 shows alteration by fuzzy set shifting:

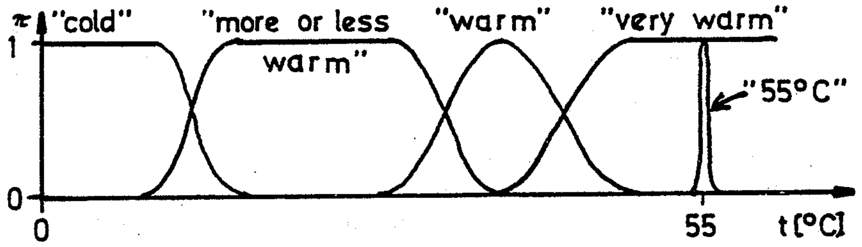
Fig.3



Correspondingly, modifiers like "more or less" may be used to fuzzify or to alter a statement. It must be determined from the context which type of modification is to be used. Alteration can be expressed in terms of precisiation or fuzzification by being more explicit. For example, "warm" in situation 2 can be expressed by "warm but not very warm" in situation 1.

The degree of fuzziness of a linguistic statement is related to the area under the corresponding possibility curve. For example, a precise temperature measurement would correspond to an impulse function in the possibility distribution. Figure 4 shows a set of linguistic labels denoting possibility distributions of various degrees of fuzziness:

Fig.4

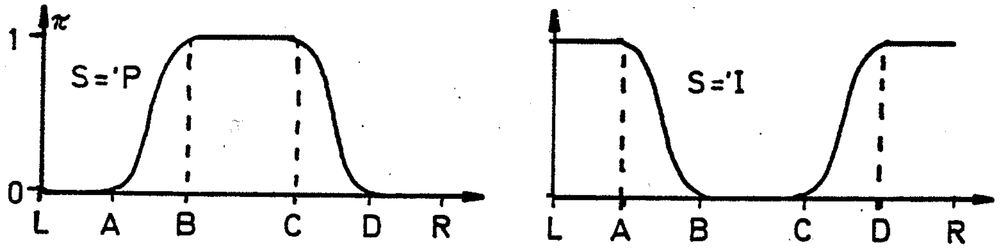


III. IMPLEMENTATION OF FUZZY CONCEPTS AND MODIFIERS

III.1 Possibility distributions.

In L-FUZZY, fuzzy sets representing descriptors like "warm" or "tall" are defined by the user. They are represented by a label and a 7-tuple {S, L, A, B, C, D, R} which characterizes the possibility distribution as illustrated in figure 5:

Fig. 5



S indicates whether the corresponding possibility distribution has standard representation (S = 'P) or is inverted (S = 'I). [L , R] denotes the discourse interval (for example, in case of water temperatures L = 0, R = 100). In standard representation, A marks the transition point of the possibility, π, from π = 0 to π > 0, B the transition from π < 1 to π = 1, C from π = 1 to π < 1, and D from π > 0 to π = 0 when proceeding through the discourse interval from L to R. In the inverted representation, π = 0 and π = 1 are interchanged. If S = 'P and π (L) = 1 or S = 'I and π (L) = 0 then we set A := B := L, and if S = 'P and π (R) = 1 or S = 'I and π (R) = 0 then we set C := D := R. Thus, in the example of figure 1, the possibility distribution for "warm" would be characterized by the 7-tuple

(P 0 15 40 100 100 100)

This representation only allows for unimodal possibility distributions, or in case of inverted distributions, for unimodal "impossibility distributions".

III.2 Modifiers.

A modifier is represented in L-FUZZY in three ways:

1. by its label,
2. by relations comparing it to other modifiers,
3. by a procedure which modifies a possibility distribution.

A set of standard modifiers is provided in L-FUZZY, but the user can easily define his own set of modifiers according to his specific requirements.

III.3 Associative net.

In FUZZY, assertions in the associative net are sorted by Z-value [11]. Thus, when searching a pattern in the associative net, assertions with high fuzzy set membership are found first. For L-values, there is no implicit sorting criterion; therefore, a partial ordering is defined for linguistic modifiers according to the emphasis they express on the modified dimension. Accordingly, the following linguistic modifiers may be ordered:

absolutely > very > : * > more or less > not

The order can be given by the user or automatically computed from the modifier function when this function is defined.

IV. MATCHING FUZZY ASSERTIONS WITH FUZZY REQUESTS

The threefold representation of linguistic modifiers allows for matching of assertions on various levels. Suppose, the fuzzy associative net contains assertions of the following form:

(<basic assertion> <linguistic assertion modifier>)

For example:

((JOHN IS TALL) very)

((BOB IS TALL) :)

((TOM IS MEDIUM-SIZED) more-or-less)

A request to the data base has the following form:

(GOAL <basic request> <linguistic request modifier>)

For example:

(GOAL (BOB IS TALL) very)

The request modifier has the function of specifying the

* ":" is the neutral modifier, called "unitor". It does not modify the associated possibility distribution.

range of possibilities which should be searched to arrive at the answer to the request. We can distinguish three types of requests to the data base:

1. basic assertion = basic request
   assertion modifier = request modifier

2. basic assertion = basic request
   assertion modifier ≠ request modifier

3. basic assertion ≠ basic request

In the first case, the request can be satisfied by "trivial matching", i.e., only the labels must be compared, the possibility distributions involved are irrelevant.

In the second case, the relative effect of modifiers must be considered. For example, in figure 2, the modifier "very" has the effect of precisiation (or selection of a fuzzy subset of possibilities) with respect to the unitor (i.e., the identity modifier). Thus, an assertion which holds when modified by "very" also holds when not modified. Conversely, an assertion which holds when not modified, possibly may hold when modified by "very". The following table shows the compatibility between four modifiers in linguistic terms:

request assertion	very	:	more or less	not
very	absolutely	indeed	no	not at all
:	possibly	absolutely	indeed	on the contrary
more or less	not quite	possibly	absolutely	no
not	not at all	on the contrary	no	absolutely

In the third case, where the basic assertion does not agree with the basic request, the possibility distributions must be analyzed in detail. For example, given that the data base contains the assertion

((TOM IS MEDIUM-SIZED) more-or-less),

the answer to the request

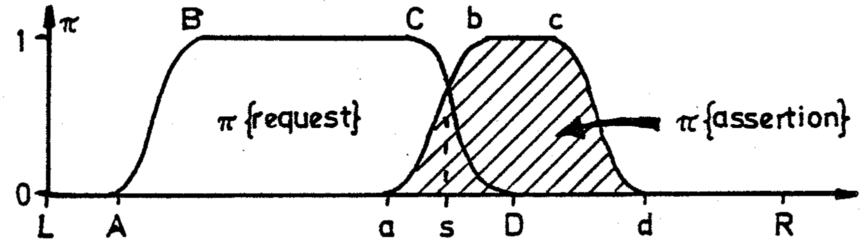
```
    (GOAL (TOM IS TALL) :)
```

requires a comparison of the possibility distributions of "medium sized" and "tall" over the height of a person. In this case, the comparison yields the answer "not quite". Ten standard answers can be generated this way: "absolutely", "indeed", "yes", "quite possibly", "possibly", "not quite", "no", "not at all", "on the contrary", and "don't know". These answers are obtained according to the following rules:

```
     if $\pi$ {assertion} (L) | $\pi$ {request} (L)
    and $\pi$ {assertion} (R) | $\pi$ {request} (R)
   then if $\pi$ {assertion} (u)
            = 1 - $\pi$ {request} (u) $\forall$ u $\in$ [L,R]
      then                                           "on the contrary"
      else                                                "don't know"
   else
     if assertion = request
   then                                                   "absolutely"
   else
     if $\pi$ {assertion} (u)
        [ $\pi$ {request} (u) $\forall$ u $\in$ [L,R]
   then     if $\pi$ {assertion} (L) = 1
            or $\pi$ {assertion} (R) = 1
          then                                                "indeed"
          else                                                   "yes"
   else
     if [b,c]  $\subset$  [B,C]
   then                                               "quite possibly"
   else
     if b  $\in$  [B,C] or c  $\in$  [B,C]
   then                                                     "possibly"
   else
     if $\pi$ (s) > 0.5
   then                                                    "not quite"
   else
     if a  $\in$  [A,D] or d  $\in$  [A,D]
   then                                                           "no"
   else                                                   "not at all"
```

For explanation of the variables refer to figure 6:

Fig.6



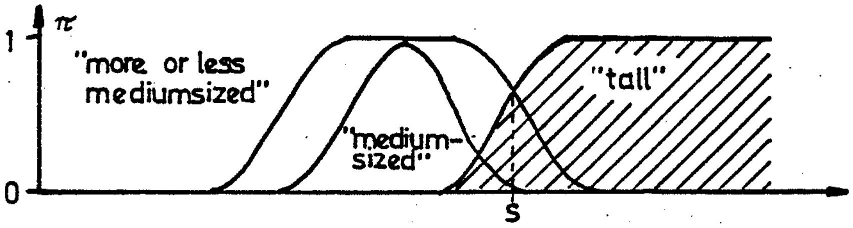
Thus, the response "absolutely" is only given, if the possibility distribution of assertion and request are identical; "on the contrary" is given, if they are complementary. If they are partially complementary, the system responds "don't know". Some fuzzy labels denote relative properties whose possibility distributions have a value of 1 at one of the endpoints of their discourse interval. Examples are "warm" (as in figure 2), "tall", "short". Emphasizing these labels corresponds to selection of a subset of the possibility distribution which coincides at the end region with the original label. In these cases, the system responds "indeed", if the assertion distribution is subset of the request distribution and it responds "possibly", if the request distribution is subset of the assertion distribution.

"Quite possibly" is the response if the assertion distribution is subset of the request distribution without having possibility one at one of the end points of the discourse interval and "not quite" if the extreme possibilities of assertion and request do not overlap, but if they have overlapping possibilities of at least 0.5. If the latter is not the case the answer is "no" and if the possibility distributions do not overlap at all, the response is "not at all".

Observe that there is relatively little qualitative difference from one possible answer to the next. This is an indication of "graceful degradation" of performance when program data degrades gradually.

L-FUZZY not only returns the final conclusion about the match between request and assertion, but also the assertion from which the conclusion was drawn. Thus, if our linguistic labels are related as shown in figure 7,

Fig.7



the request

```
    (GOAL (TOM IS TALL))
```

yields

```
    (not-quite:  ((TOM IS MEDIUM-SIZED) more-or-less)
```

V. SUMMARY

A programming language for direct representation of imprecise concepts is presented. As in the language FUZZY, a fuzzy pattern consists of two parts: a basic statement and a statement modifier. In L-FUZZY, fuzzy statements are represented by labels associated to possibility distributions, and modifiers by labels associated to procedures and by relations to other modifiers. The possibility distributions characterize the range of values which a representative of the imprecise concept may assume. The multiple representations of patterns and modifiers allow for high-level comparison of imprecise concepts. In particular, if the basic statements of two assertions agree, only relations between modifiers must be compared, the possibility distributions can be ignored. The result of a comparison is given in linguistic terms and reflects the quality or conclusiveness of the answer. This is more informative than a yes/no response.

L-FUZZY is used for object identification from incomplete and imprecise linguistic descriptions and may find further application in information retrieval systems. A possible extension of L-FUZZY is the use of the qualitative answer to requests as control information for the program flow.

ACKNOWLEDGEMENT

I wish to thank Lotfi Zadeh for convincing me that imprecision may be desirable, Rick LeFaivre for providing a wonderful language to build on, Andrew Clarke, Gerhard Dirlich and Ulrich Furbach for valuable comments on earlier drafts of this paper.

REFERENCES
and related papers

1. Adamo, J.M. (1980). L.P.L. A fuzzy programming language: 1. syntactic aspects. Fuzzy Sets and Systems 3, 151-179.

2. Bobrow, D. & Raphael, B. (1974). New programming languages for Artificial Intelligence Research. Computing Surveys, 6, 153-174.

3. Bonissone, P. (1979). The problem of linguistic approximation in system analysis. Ph.D. thesis, Dept. of Electrical Engineering & Computer Sciences, University of California, Berkeley.

4. v.Hahn, W., Hoeppner, W., Jameson, A., Wahlster, W. (1979). The anatomy of the natural-language dialogue system HAM-RPM. In: Bolc, L. (ed.) Natural language based computer systems. Hanser/Macmillan Verlag Muenchen, London.

5. Hersh, H.M. & Caramazza, A. (1976). A fuzzy set approach to modifiers and vagueness in natural language. J. Exp. Psych. 105, 254-276.

6. Kling, R. (1974). Fuzzy PLANNER: reasoning with inexact concepts in a procedural problem-solving language. Journal of Cybernetics, 4, 105-122.

7. Kochen, M. & Badre, A.N. (1974). On the precision of adjectives which denote fuzzy sets. Journal of Cybernetics 4, 1, 49-59.

8. Lakoff, G. (1973). Hedges: a study in meaning criteria and the logic of fuzzy concepts. Journal of Philosophical Logic, 2, 458-508.

9. LeFaivre, R.A. (1974). Fuzzy problem solving. Ph.D. thesis, Dept. of Computer Science, University of Wisconsin, Madison.

10. LeFaivre, R.A. (1974). The representation of fuzzy knowledge. Journal of Cybernetics, 4, 57-66.

11. LeFaivre, R.A. (1977). FUZZY reference manual. Computer Science Department, Rutgers University.

12. Noguchi, K., Umano, M., Mizumoto, M., Tanaka, K. (1976). Implementation of fuzzy artificial intelligence language FLOU. Technical Report in Automation and language of IECE.

13. Sridharan, M.J. et al. (1978). AIMDS user manual version 2. Department of Computer Science, Rutgers University, New Brunswick, N.J.

14. Umano, M., Mizumoto, M., Toyoda, J., Tanaka, K. (1975). On interpretation and execution of fuzzy programs. Technical Report on Automation and Language of IECE, AL 75-27.

15. Wenstop, F. (1975). Application of linguistic variables in the analysis of organizations. Ph.D. thesis, School of Business Administration, University of California, Berkeley.

16. Zadeh, L.A. (1972). A fuzzy-set-theoretic interpretation of linguistic hedges. Journal of Cybernetics, 2, 4-34.

17. Zadeh, L.A. (1975). The concept of a linguistic variable and its application to approximate reasoning.
Part I. Information Sciences, 8, 199-249.
Part II. Information Sciences, 8, 301-357.
Part III. Information Sciences, 9, 43-80.

18. Zadeh, L.A. (1977). Fuzzy sets as a basis for a theory of possibility. Journal of Fuzzy Sets and Systems.

19. Zadeh, L.A. (1978). PRUF - a meaning representation language for natural languages. Int. J. Man-Machine Studies 10, 395-460.

NOTE

A copy of the L-FUZZY system for PDP-10 and DECsystem-20 computers may be obtained by sending a magtape to

Christian Freksa
EECS Department
University of California
275 Cory Hall
Berkeley, Ca 94720

phone: (415) 642-1989