Workshop on Coping with
Complexity in Systems,
London, May 14-15, 1981

(95)

APPLICATION OF POSSIBILITY THEORY TO FUZZY DATABASE

Masaharu MIZUMOTO*
Inst. f. Wirtschftswissen.
RWTH Aachen
5100 Aachen, West Germany

Motohide UMANO
Dept. of Appl. Mathematics
Okayama Univ. of Science
Okayama 700, Japan

INTRODUCTION

Database systems have been vigorously studied since Codd [1] proposed the relational model of data in 1970. Such database systems can only deal with well-defined and unambiguous data. In the real world, however, there exist uncertain or ambiguous data and information which cannot be defined in certain and well-defined form by any means. Since in everyday life we often make decisions based on such fuzzy data, the formulation and construction of a database which can represent and manipulate fuzzy data will increase the application areas of database systems and improve the interface for the smooth communication between men and machines.

Based on the theory of possibility distribution by Zadeh [2], we have constructed a fuzzy database system called FREEDOM (Fuzzy Relational Extension for Data Organization and Manipulation) [3,4] which is a fuzzy version of Codd's relational model and can represent and manipulate fuzzy data. The data manipulation language of FREEDOM provides QUERY, INSERT, DELETE, DEFR (DEfine Fuzzy Relations) and DEFR (DEfine Fuzzy Predicates) statements. The system FREEDOM is implemented in a fuzzy set manipulation language "FSTDSL/FORTRAN" [5] and currently running on a FACOM 230-45S Computer.

This paper gives an outline of FREEDOM using some examples of programs using QUERY statements.

---

* On leave from Osaka Electro-Communication Univ., Osaka, Japan, until Aug. 31, 1981.

POSSIBILITY DISTRIBUTIONS

If F is a fuzzy set in U characterized by its membership function $\mu_F: U \rightarrow [0,1]$, then a fuzzy proposition

$$x \text{ is } F \tag{1}$$

induces a possibility distribution $\Pi_{A(x)}$ which is equal to F, where A is the attribute of an object x. That is to say,

$$x \text{ is } F \rightarrow \Pi_{A(x)} = F \tag{2}$$

Such a distribution is characterized by a possibility distribution function $\pi_{A(x)}: U \rightarrow [0,1]$ (identified to μ_F) which associates with each u in U the possibility that A(x) may take u as a value. Thus,

$$\Pi_{A(x)} = \{\pi_{A(x)}(u)/u : u \in U\}_p \tag{3}$$

$$= \{\mu_F(u)/u : u \in U\}_p \tag{4}$$

where the suffix p is used to emphasize that the fuzzy set F represents a possibility distribution.

As a simple example, consider a fuzzy proposition:

Q: Tom is about 20 years old

in which "about 20" is a fuzzy set in the domain of age defined by

$$\underset{\sim}{20} = \{0.8/18,\ 1/19,\ 1/20,\ 1/21,\ 0.8/22\}.$$

Then, from the fuzzy proposition Q we can obtain a possibility distribution $\Pi_{AGE(Tom)}$ for the age of Tom such as

$$\Pi_{AGE(Tom)} = \{0.8/18,\ 1/19,\ 1/20,\ 1/21,\ 0.8/22\}_p$$

For example, the possibility that Tom's age AGE(Tom) may be 18 is equal to 0.8 and likewise for other ages.

We shall next introduce some special possibility distributions which are needed in the later discussion.

$$\text{UNKNOWN} = \{1/u : u \in U\}_p \tag{5}$$

This possibility distribution $\Pi_{A(x)}$ = UNKNOWN represents that there is possibility that A(x) could be any value in U but we can obtain no information about A(x) from the possibility distribution $\Pi_{A(x)}$.

$$\text{UNDEFINED} = \{0/u : u \in U\}_p \tag{6}$$

This represents that there is no possibility that the value of A(x) could exist in U. For example, let PROF(x) be the profession of x, then

$$\Pi_{PROF(Tom)} = UNDEFINED \tag{7}$$

means that the profession of Tom is not defined, that is, Tom has no profession.

For the possibility distribution $\Pi_{A(x)} = \{1/u_i\}_P$ which consists of one element u_i (which means $A(x) = u_i$), the possibility distribution

$$\Pi_{A(x)}^{*} = \{1/u_i\}^{*}_{p} \tag{8}$$

represents that we cannot assert that A(x) is u_i but we can affirm that A(x) is probably equal to u_i.

Finally, the possibility distribution NULL is used to represent that we do not even know whether A(x) may take a value in U or not. For example,

$$\Pi_{PROF(Tom)} = NULL$$

means that we do not even know whether Tom has a profession or not.

FUZZY DATABASE

We shall consider a fuzzy relational model of the fuzzy database system FREEDOM. Some of the data in the fuzzy relational model are characterized by a possibility distribution.

As a simple example, let us consider a fuzzy relation PERSON in Table 1 whose attributes are NAME, AGE and CHILD-NAME. This

TABLE 1 Fuzzy Relation PERSON

PERSON

NAME	AGE	CHILD-NAME
Tom	23	Ted
Susan	35	John
Susan	35	Mike
Richard	40	$\{Judy, Anna\}_p$
Raymond	YOUNG	UNKNOWN
Victor	UNKNOWN	UNDEFINED
Smith	$\{1/50, .5/49, .5/51\}_p$	NULL
Jack	OLD	$\{Betty\}^{*}_{p}$

fuzzy relation PERSON represents the following meaning: We know that Tom is 23 years old and has a child named Ted. Susan is 35 years old and has two children whose names are John and Mike, whereas Richard's age is 40 and he has one child whose name is either Judy or Anna. The possibility distribution $\{Judy, Anna\}_p$ means $\{1/Judy, 1/Anna\}_p$. Raymond's age is young and characterized by a possibility distribution YOUNG as in Fig.1, and the value UNKNOWN means that we know he has a child but we don't know his child's name. As for Victor, we do not know his age and the value UNDEFINED represents that he has no children. Smith's age is about 50 and the value NULL means that we don't know even whether he has any children or not. Finally, Jack is old and his child's name is probably Betty.

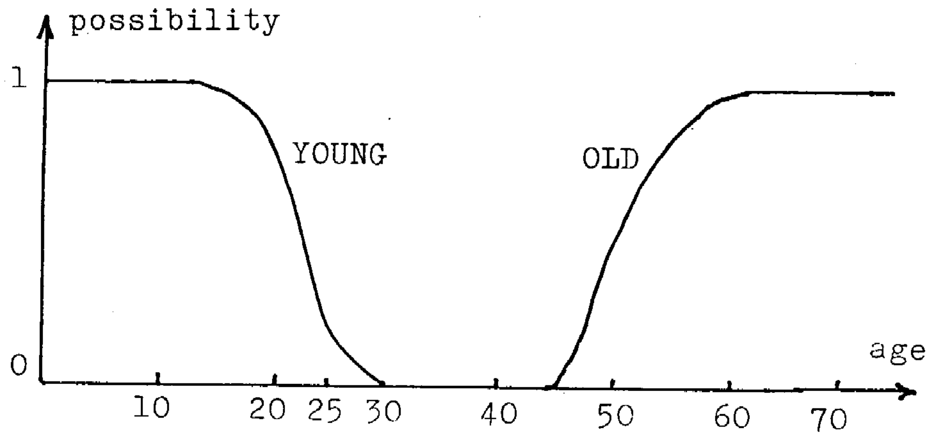

Fig.1 Possibility distributions YOUNG and OLD

It is found from the above example that we can express uncertain and ambiguous information by using a possibility distribution in the fuzzy database. It is impossible, however, to cope with such fuzzy data by traditional two-valued and multi-valued logic systems. In [3,4] we proposed a logic system which can deal with such fuzzy data represented by a possibility distribution.

In the following we shall illustrate a simple example of how to retrieve data from a fuzzy database by using Table 1. Let us consider the following query:

$$\text{Find a person whose age is greater than 25.} \qquad (9)$$

The results to be obtained will be divided into three types: (1) the results which certainly satisfy the condition of the given query. (2) the results which probably satisfy it. (3) the results which do not satisfy it. In fact, Tom does not satisfy the condition, while

Susan and Richard clearly satisfy it. As for Raymond, if the possibility distribution YOUNG is given as in Fig.1, then the possibility is positive at the age greater than 25. Thus, Raymond probably satisfies the condition of the query. Victor has also the possibility of satisfaction of the condition in the light of UNKNOWN. Smith clearly satisfies the condition because his age is between 49 and 51. Jack also satisfies the condition if the possibility distribution OLD is as shown in Fig.1.

Based on the above point of view, we have designed and implemented a data manipulation language of the fuzzy database system FREEDOM. For example, the query (9) for the relation PERSON of Table 1 can be written in the FREEDOM language as

```
QUERY A (NAME = X):
  PERSON (AGE = ?Y, NAME = ?X);
  GE(*Y, 25);
QEND
```

This represents that the value of X (its attribute name is NAME) which satisfies the condition (2-3 lines) between QUERY and QEND statements is inserted into a set A.

The retrieval result is:

A@1 = {Susan, Richard, Smith, Jack}
A@2 = {Raymond, Victor}

where the set A@1 contains the elements which <u>certainly</u> satisfy the the condition, while A@2 contains the elements which <u>probably</u> satisfy it.

EXAMPLES OF QUERY STATEMENTS IN FREEDOM LANGUAGE

FREEDOM language provides QUERY, INSERT, DELETE, DEFR (DEfine Fuzzy Relation) and DEFP (DEfine Fuzzy Predicate) statements and is embedded into FSTDSL/FORTRAN [5]. It is currently running on a FACOM 230-45S computer. The QUERY statement retrieves data from a fuzzy database. The INSERT statement inserts and the DELETE statement deletes several tuples into and from the specified fuzzy relation, respectively. The DEFR statement declares fuzzy relations together with its attributes and their types before the INSERT statements insert tuples into them. The DEFP statement defines fuzzy predicate to be used in a QUERY statement. A more detailed exposition of FREEDOM language is provided in [3,4].

We shall next describe some facilities of our data manipulation language by writing several queries in it. Let us consider three fuzzy relations CANDIDATE, PROF and BODY in Table 2 which are defined by DEFR and INSERT statements. The basic set of the attribute SEX is {MALE, FEMALE}. The attribute SCAREER stands for a school career and its basic set is {U, H}, where U denotes a university and H a high school. The element which starts with Yen mark ¥ represents a possibility distribution. Especially, the possibility distribution ¥An means "about n." ¥¥U denotes $\{1/U\}^*_p$ in (8).

(1) Retrieve the name of a person who graduates from a university.

```
QUERY A (NAME = X):
  CANDIDATE (NAME = ?X, SCAREER = U);
QEND
```

```
Result: A@1 = FSET(1/SMITH, 1/MARY);
        A@2 = FSET(1/RICHARD, 1/SUSAN);
```

The result is output in the form of fuzzy set, where FSET is the fuzzy-set construction operator in FSTDSL/FORTRAN. As for the result, SMITH and MARY in A@1 clearly satisfy the condition of the query (1), whereas RICHARD's SCAREER is UNKNOWN and he may graduate from a university, so he is in A@2. SUSAN may probably graduate from a university because of ¥¥U (= $\{1/U\}^*_p$) and thus she is in A@2.

(2) (i) Retrieve the name and age of a person who is 25 years old.
(ii) Retrieve the name and age of a person who is about 25 years old.

```
(i) QUERY AO (NAME=U, AGE=V):
      CANDIDATE (NAME=?U, AGE=?V);
      EQ(*V, 25);
    QEND
```

```
Result: AO@1 = EMPTY;
        AO@2 = FSET(1/<RICHARD, ¥A25>, 0.7/<MARY, ¥A24>);
```

```
(ii) QUERY A1 (NAME=U, AGE=V):
       CANDIDATE (NAME=?U, AGE=?V);
       FEQ(*V, @A25);
     QEND
```

```
Result: A1@1 = FSET(1/<RICHARD, ¥A25>, 0.8/<MARY, ¥A24>);
        A1@2 = EMPTY;
```

TABLE 2 Fuzzy Relations CANDIDATE, PROF and BODY

(a) Fuzzy Relation CANDIDATE

CANDIDATE

NAME	SEX	AGE	SCAREER
SMITH	MALE	30	U
JOHN	MALE	¥A28	H
RICHARD	MALE	¥A25	¥UNKNOWN
ANNA	FEMALE	22	H
MARY	FEMALE	¥A24	U
LUCY	FEMALE	¥A20	H
SUSAN	FEMALE	23	¥¥U

(b) Fuzzy Relation PROF

PROF

NAME	PROFES	INCOME
SMITH	PROGRAMMER	300
JOHN	ENGINEER	¥A200
RICHARD	¥UNDEFINED	¥UNDEFINED
ANNA	CLERK	¥A150
MARY	TEACHER	¥UNKNOWN
LUCY	¥UNDEFINED	¥UNDEFINED
SUSAN	PROGRAMMER	¥¥250

(c) Fuzzy Relation BODY

BODY

HEIGHT	WEIGHT	NAME
180	¥A60	SMITH
¥A175	80	JOHN
165	¥A60	RICHARD
¥A170	¥A55	ANNA
165	55	MARY
170	60	LUCY
¥A175	¥A55	SUSAN

(i) is a query asking for a person whose age is just 25. There is not such a person and thus AO@1 is empty. The attribute value of AGE for MARY is given as a possibility distribution ¥A24 which is assumed to be

¥A24 = $\{0.7/23,\ 1/24,\ 0.7/25\}_p$

Thus, the possibility that she may be 25 years old is given by 0.7.

In (ii) the symbol @ followed by A25 in the 3rd line means that the name A25 is a fuzzy set name and this fuzzy set is defined as

A25 = $\{0.7/24,\ 1/25,\ 0.7/26\}$

The predicate EQ in (i) compares two parameters by elements, while the predicate FEQ (fuzzy equal?) compares by fuzzy sets. For example, in the case of MARY in (ii), the grade 0.8 attached to <MARY,¥A24> represents the compatibility of A25 and ¥A24.

(3) Retrieve the name of a person who is young.

```
QUERY B (NAME = X):
   CANDIDATE (NAME=?X, AGE=?Y);
   YOUNG(*Y);
QEND
```

```
Result:  B@1 = FSET(1/ANNA, 1/LUCY, 0.8/SUSAN);
         B@2 = FSET(0.6/RICHARD, 0.8/MARY);
```

A fuzzy predicate YOUNG is given by

YOUNG = (1/19, 1/20, 1/21, 1/22, 0.8/23, 0.6/24, 0.3/25)

which is defined by the DEFP statement. Using the predicate YOUNG, the truth value that ANNA and SUSAN is YOUNG is obtained as 1 and 0.8, respectively. For LUCY, since her age is given as

¥A20 = $\{0.7/19,\ 1/20,\ 0.7/21\}_p$

and the truth value of YOUNG for 19, 20 and 21 is 1, LUCY is in B@1. This is the representative case where ambiguous data and ambiguous query make a certain answer. For RICHARD, the grade value 0.6 is obtained as the maximum value of 0.6 and 0.3 which are truth values of YOUNG for his possible ages 24 and 25, respectively.

(4) Retrieve the name of a female whose profession is a programmer.

```
QUERY C (NAME = X):
   CANDIDATE (NAME=?X, SEX=FEMALE);
   PROF (NAME=*X, PROFES=PROGRAMMER);
QEND
```

```
Result:  C@1 = FSET(1/SUSAN);
         C@2 = EMPTY;
```

(5) Retrieve the name and weight of a female who graduates only a high school and whose weight is greater than or equal to the average of all persons.

```
QUERY E (NAME=U, WEIGHT=V):
   QUERY W (NAME=X, WEIGHT=Y):
      BODY (NAME=?X, WEIGHT=?Y);
   QEND
   CANDIDATE (NAME=?U, SEX=FEMALE, SCAREER=H);
   BODY (NAME=*U, WEIGHT=?V);
   GE(*V, AVG(W@1, 2));
QEND
```

```
Result:  E@1 = EMPTY;
         E@2 = FSET(1/<LUCY, 60>);
```

AVG is a function which computes the average. Its first parameter W@1 is a set of tuples <<u>name</u> , <u>weight</u>> obtained by the QUERY statement of the 2-4 lines. The second parameter 2 indicates the 2nd element of the tuple, that is, the value of the attribute WEIGHT. Thus, AVG computes the average of the values of WEIGHT.

(6) Retrieve the name of every profession in which a university graduate is engaged, if there exists a profession by a university graduate other than programmer, lawer and engineer.

```
QUERY PROFESSION (PROFES=P):
   QUERY U (PROFES=PU):
      CANDIDATE (NAME=?X, SCAREER=U);
      PROF (NAME=*X, PROFES=?PU);
   QEND
   NOT(CONTAIN(PR,U@1));
   CANDIDATE (NAME=?N, SCAREER=U);
   PROF (NAME=*N, PROFES=?P);
QEND
```

```
Result: PROFESSION@1 = FSET(1/PROGRAMMER, 1/TEACHER);
        PROFESSION@2 = FSET(1/¥UNDEFINED);
```

PR in 6th line is a set defined by

$$PR = \{ \text{PROGRAMMER, LAWER, ENGINEER} \}$$

CONCLUSION

Our fuzzy relational database system FREEDOM can represent and manipulate uncertain or ambiguous data represented by possibility distributions. This system facilitates the representation of uncertainty and ambiguity contained in data itself, but does not have a facility of representing uncertainty and ambiguity in the relationship between fuzzy data. The data model which overcomes this problem is investigated in connection with a fuzzy version of relational algebra by the authors [6,7,8].

REFERENCES

1. Codd, E.F. (1970). A relational model of data for large shared data banks. Comm. ACM, 13, 377-387.
2. Zadeh, L.A. (1978). PRUF - A meaning representation language for natural languages. Int. J. of Man-Machine Studies, 10, 395-460.
3. Fukami, S., M. Umano, M. Mizumoto & K. Tanaka (1979). Fuzzy database retrieval and manipulation language. Technical Reports on Automaton and Language of IECE of Japan, AL78-85, 65-72.
4. Umano, M., M. Mizumoto & K. Tanaka (1979). Fuzzy database systems. Proc. of Working Conf. on Database Engineering (13th IBM Computer Science Symp., Shizuoka, Japan, Nov. 17-19, 1979), 33-55.
5. Umano, M., M. Mizumoto & K. Tanaka (1978). FSTDS system: A fuzzy-set manipulation system. Inform. Sciences, 14, 115-159.
6. Umano, M. (1979). Representation and Manipulation of Fuzzy Data. Ph.D. Thesis, Osaka University.
7. Umano, M., M. Mizumoto & K. Tanaka (1981). Applications of alpha expressions to fuzzy relations. In B. Rieger (Ed), Empirical Semantics, Bochum (forthcoming).
8. Umano. M., S. Fukami, M. Mizumoto & K. Tanaka (1979). On fuzzy relational algebra. 20th National Convention Records of IPS of Japan, 1D-6, 693-694.