Fuzzy Database Query Languages and Their Relational Completeness Theorem

Yoshikane Takahashi

Abstract—Two fuzzy database query languages are proposed. They are used to query to fuzzy databases that are enhanced from relational databases in a way that fuzzy sets are allowed in both attribute values and truth values. A fuzzy calculus query language is constructed based on the relational calculus and a fuzzy algebra query language is also constructed based on the relational algebra. In addition, this paper proves a fuzzy relational completeness theorem such that both the languages have equivalent expressive power to each other.

Index Terms—Fuzzy database, query languages, relational algebra, relational calculus, relational completeness.

I. INTRODUCTION

DATABASE technology has been advanced up to the relational database stage with the purpose that user interfaces with databases may approach a level of human interfaces. It is recognized that the fuzzy theory is suitably applied to some human-oriented engineering fields, one of which is information processing, in particular database retrieval. In fact, fuzzy database models that allow fuzzy attribute values and fuzzy truth values in enhanced relational databases have been studied in [3] and [4]. However, these studies are restricted to just some particular applications and not grounded on theories of fuzzy database query languages. Thus fuzzy database systems would not be systematically developed on the basis of these studies; it is due to Codd's relational database theory that relational database systems have been systematically developed. It is desirable that theoretical foundations of fuzzy databases be established in order to systematically develop fuzzy database systems.

Recently, an excellent work has been done in the field of the fuzzy database theory; it develops a theoretical foundation for the fuzzy functional dependencies of fuzzy databases [1]. The work encourages further research for the rest of theoretical foundations of fuzzy databases. This paper thus aims to develop a theoretical foundation of query languages to fuzzy databases. It proposes two fuzzy database query languages: a fuzzy calculus query language and a fuzzy algebra query language. In addition, it proves a relational completeness theorem such that both the languages are equivalent in expressive power to each other. With these theoretical foundations, fuzzy database query systems will be developed systematically.

Fig. 1. A query to a fuzzy database.

II. A FUZZY DATABASE MODEL

A fuzzy database is defined as an enhanced relational database that allows fuzzy attribute values and fuzzy truth values; both of these are expressed as fuzzy sets. An example of the fuzzy database is shown in Fig. 1.

A. Fuzzy Data Model

A fuzzy database consists of relations: a relation is a relation $R(t_1, \ldots, t_n)$ in a Cartesian product $P_1 \times P_2 \times \cdots \times P_n$ of domains $F_i$; each $P_i$ is a set of fuzzy sets $t_i$ over an attribute domain $D_i$ $(1 \leq i \leq n)$. It is assumed that key attributes take ordinary nonfuzzy values. For the notational convenience, fuzzy sets are identified with their representative membership functions; for example, $t_i$ also denotes a membership function.

B. Fuzzy Attribute Values

Attribute values such as age have nonfuzzy values such as 20 in the relational database; attribute values are defined as fuzzy predicates such as "young" and "about forty" in the fuzzy database. For example, a fuzzy attribute value of "age of Dr. X is young" is expressed as a possibility distribution $P(\text{age of } X) = \text{YOUNG}$; YOUNG denotes a fuzzy set that represents the fuzzy predicate "young." Thus attribute values are identified with fuzzy sets such as YOUNG.

C. Fuzzy Truth Values

Truth values of any tuples are either 1 (= true) or 0 (= false) in the relational database; truth values of any tuples are defined as fuzzy predicates such as "0.7" and "completely true" in the fuzzy database. Consider, for example, a tuple $t$ that asserts a fuzzy proposition: "It is completely true that Dr. X is very much older than twenty." The truth value of $t$ is expressed as a possibility distribution $P[T(t)] = N$; $T(t)$ denotes a truth value of $t$ and $N$ denotes a fuzzy set that represents the fuzzy...
A. Tuple Fuzzy Calculus

A tuple fuzzy calculus (query language) is constructed as an enhancement of the tuple relational calculus. Formulas in the tuple fuzzy calculus are of the form \( t(f(t)) \); \( f \) is a fuzzy tuple variable each \( i \)th component \( t_i \), which is a fuzzy set in \( P_i \); \( f \) is a tuple fuzzy well-formed formula (WFF).

Tuples fuzzy WFF's are enhanced from those of the tuple relational calculus as follows.

1) Atomic Tuple Fuzzy WFF's: An atomic tuple fuzzy WFF consists of fuzzy sets and a fuzzy comparison operator \( * \). The fuzzy comparison operator \( * \) is one of the operators: equal; not equal; proper inclusion; inclusion. The fuzzy comparison operator \( * \) is an enhancement from the arithmetic comparison operator \( (=, \neq, <, >, \leq, \geq) \) in the relational calculus. Then the atomic tuple fuzzy WFF's are either of the following two types:

1) \( (t_i) * (s_j) \); here, it is assumed that \( t \) and \( s \) are fuzzy tuple variables such that \( D_i = D_j \) (1 \( \leq i, j \leq n \)).

2) \( (t_i) * (c) \); \( (c) * (t_i) \); here, it is assumed that \( c \) is a fuzzy set over \( D_i \).

2) Logical Connectives and Quantifiers: Logical connectives ("AND", "OR," and "NOT") are used for tuple fuzzy WFF's. Also, quantifiers ("for all" and "there exists") are used for tuple fuzzy WFF's.

3) Others: Other definitions concerning tuple fuzzy WFF's are the same as in the tuple relational calculus.

Thus tuples in any relation \( R(t_1, \ldots, t_n) \) that satisfy the formula \( t(f(t)) \) form a set of Cartesian products of fuzzy sets.

It should be considered further whether or not to include fuzzy comparison operators \( * \) expressed by fuzzy relations such as "much greater than," "is close to," "is similar to," and "is relevant to."

B. Query Evaluation

Queries expressed in the tuple fuzzy calculus are evaluated by two steps as follows.

**Step 1** Calculating truth values of resultant tuples: Let any resultant tuple \( r \) be expressed as \( r_{11} \cdots r_{kj} \cdots r_{km} \), and \( r \) be a projection of \( t \in R(t_1, \ldots, t_n) \) onto the components \( k_1, \ldots, k_j, \ldots, k_m \). Then the truth value \( T(r) \) is defined as a projection of \( T(t) \) onto the components \( k_1, \ldots, k_j, \ldots, k_m \), which is a fuzzy set over \( \{0, 1\} \). Then the maximum is taken over those components \( t_k \) (1 \( \leq k \leq n \)), such that \( t_k \neq t_{kj} \).

Duplicate removal schemes are out of the scope of this paper and left for future work: if two tuples \( r_1, r_2 \) have different truth values \( T(r_1), T(r_2) \) are found to be duplicated, it is left up to fuzzy database designers which one will be selected. The fuzzy database designers will also choose which tuples from the resultant tuples \( r \) should be returned to the users:

1) full sets or appropriate subsets of resultant tuples \( r \) should be returned;

2) tuples \( r \) that contain truth values \( T(r) \) should be returned; or

when users need not make use of truth values \( T(r) \), tuples \( r \) from which truth values \( T(r) \) are removed, should be returned.

IV. QUERY BY DOMAIN FUZZY CALCULUS

A domain fuzzy calculus (query language) is obtained from the tuple fuzzy calculus through the following replacements:

1) replacement of tuple variables \( t \) with domain variables, \( u_1; u_2; \ldots; u_m \);

2) replacement of the \( i \)th tuple component \( t_i \) with a domain variable \( u_i \) (1 \( \leq i \leq n \)).

V. QUERY BY FUZZY ALGEBRA

A. Fuzzy Algebra

A fuzzy algebra (query language) is constructed as an enhancement of the relational algebra. Fundamental fuzzy algebraic operations are union, set difference, Cartesian product, projection, and selection, which are defined as follows.

1) Union: Let \( R \) and \( S \) denote any relations in the fuzzy database. The union of \( R \) and \( S \) is a set of tuples that belongs to \( R \) or \( S \). The union is equal to that in set theory.

Any resultant tuple \( t \) by the union of \( R \) and \( S \) inherits the truth value \( T(t) \) from its original tuple in \( R \) or \( S \).

2) Set Difference: The difference is equal to that in set theory.

Any resultant tuple \( t \) by the set difference \( R - S \) inherits the truth value \( T(t) \) from its original tuple in \( R \).

3) Cartesian Product: The Cartesian product \( R \times S \) of \( R \) and \( S \) is a set of tuples, \( \{(r, s) | r \in R, s \in S\} \). The Cartesian product is equal to that in set theory.

The truth value \( T(t) \) of the resultant tuple \( t = (r, s) \) by the Cartesian product \( R \times S \) is the minimum of \( T(r) \) and \( T(s) \) where \( T(r) \) and \( T(s) \) are truth values of \( r \) and \( s \), respectively.

4) Projection: The projection \( \text{Proj}(k_1, \ldots, k_j, \ldots, k_m)(R) \) of \( R \) onto the \( k_j \)th attributes is a set of tuples of the \( k_j \)th attribute values. The projection is equal to that in set theory.

Let \( r \) denote any resultant tuple of the projection \( \text{Proj}(i_1, i_2, \ldots, i_m)(R) \) of \( t \in R \). Then the truth value \( T(r) \) is
the maximum of \( T(t) \) taken over those components \( t_k \), such that \( t_k \neq t_{kj} \).

5) **Selection:** Let \( G \) denote a fuzzy WFF involving the following constituents:
   i) operands that are constant fuzzy sets and attribute item numbers of the relation \( R \);
   ii) the fuzzy set comparison operators * (equal, not equal, proper inclusion, inclusion);
   iii) logical connectives “OR,” “AND,” and “NOT.”

The selection \( \text{Sel}_G(R) \) of the relation \( R \) is a set of tuples \( t \in R \) each of which satisfies the fuzzy WFF \( G \) when any occurrences of the number \( i \) in \( G \) are replaced with the \( i \)-th component of \( r \) in \( R \).

When any resultant tuple \( r \) is made by the selection \( \text{Sel}_G(R), t \in R \) inherits the truth value \( T(t) \) from the original tuple \( t \in R \): \( T(r) = T(t) \).

Some additional fuzzy algebraic operations such as intersection, quotient, \( \theta \)-join, and natural join are defined as combinations of the fundamental fuzzy algebraic operations defined previously in the same way as in the relational algebra. For example, the \( \theta \)-join and the natural join are defined as follows.

6) **\( \theta \)-Join:** The \( \theta \)-join of \( R \) and \( S \) is defined as a combination of two fundamental fuzzy algebraic operations: the Cartesian product and the selection where \( \theta \) is enhanced to a fuzzy comparison operator * (equal, not equal, proper inclusion, inclusion). Truth values of resultant tuples by the \( \theta \)-join are calculated as those of combinations of the two fundamental fuzzy algebraic operations.

7) **Natural Join:** The natural join of \( R \) and \( S \) is defined as a combination of three fundamental fuzzy algebraic operations: the Cartesian product, the selection, and the projection. Truth values of resultant tuples by the natural join are calculated as those of combinations of the three fundamental fuzzy algebraic operations.

**B. Query Evaluation**

Any query by the fuzzy algebra is expressed as a combination of the fundamental fuzzy algebraic operations. Thus the resultant tuples \( r \) and their truth values \( T(r) \) by this query are obtained as combinations of its constituent fundamental fuzzy algebraic operations.

Duplicate removal schemes and return methods of resultant tuples to users are the same as described in the fuzzy calculus.

**VI. RELATIONAL COMPLETENESS THEOREM FOR FUZZY DATABASE QUERY LANGUAGES**

The relational database theory establishes the relational completeness theorem such that the relational calculus is equivalent in expressive power to the relational algebra [2]. A similar theorem in the fuzzy database is given.

**Theorem:** The following three fuzzy database query languages have the same expressive power:
1) tuple fuzzy calculus;
2) domain fuzzy calculus;
3) fuzzy algebra.

**Proof:** The fundamental idea of the proof of this theorem is given by Ullman [2, pp. 114–122]; it presents the proof of the relational completeness theorem for the relational database query languages. Ullman’s proof techniques consist of the following three techniques:
   i) reduction of the relational algebra to the tuple relational calculus;
   ii) reduction of the tuple relational calculus to the domain relational calculus;
   iii) reduction of the domain relational calculus to the relational algebra.

The reduction technique ii) is just the transformation between variable expressions, and thus is not influenced by the enhancements of the fuzzy database query languages. Therefore, it should be proved here that the reduction techniques i) and iii) can also be extended to cover the enhancements of the fuzzy database query languages.

There are two essential enhancements in the fuzzy database query languages from the relational database:
1) The fuzzy database allows fuzzy sets as attribute values; the fuzzy comparison operators * (equal, not equal, proper inclusion, inclusion) are used in the fuzzy database query languages instead of the arithmetic comparison operators (=, ≠, <, >, ≤, ≥) used in the relational database query languages.
2) The fuzzy database allows fuzzy sets as truth values \( T(t), t \in R \); truth values \( T(r) \) of resultant tuples \( r \) are inherited from \( T(t) \) of original tuples \( t \in R \), or calculated as combinations of Cartesian products or projections of \( T(t) \) of original tuples \( t \in R \).

The enhancement 1) is easily incorporated into the reduction techniques i) and iii) by replacing the arithmetic comparison operators with the fuzzy comparison operators.

Next, consider the enhancement 2). Remember that the truth value \( T(t) \) of the tuple \( t \) is defined just depending on the fuzzy set and fuzzy set operations that have been established in fuzzy theory; calculations of \( T(r) \) are made independently of any of the definitions of the fuzzy database query languages. Thus the reduction techniques i) and iii) can be extended to incorporate the enhancement 2). This completes the proof.

QED

**VII. CONCLUDING REMARKS**

This paper proposes two fuzzy database query languages (fuzzy relational calculus and fuzzy relational algebra) based on the relational database query languages. In addition, it proves the relational completeness theorem such that both the languages are equivalent in expressive power to each other. As in the case of the relational database, this relational completeness theorem in the fuzzy database is expected to provide a criterion for the minimum fuzzy database query capability that must be implemented in any reasonable real fuzzy database query languages.

There are interesting further theoretical studies still left. More complicated fuzzy queries, including more general fuzzy comparison operators such as “much greater than,” and “is close to,” need to be studied. Such queries include, for example, a statement “select several persons where their age is...
a little over than that of young boys." Other studies need to be devoted to duplicate removal schemes and query optimization techniques to improve execution efficiency of the fuzzy query languages; both of these are completely out of the scope of this paper though these are essential to the fuzzy database. Practically, there also should be an interesting further study how to implement the fuzzy database query languages in this paper by extending the existing real relational database query languages, such as the international standard database language SQL.

REFERENCES


Yoshikane Takahashi received the M.Sc. degree in mathematics from the University of Tokyo, Tokyo, Japan in 1975. He is currently with NTT Network Information Systems Laboratories, Kanagawa, Japan. His research fields include communications protocol, fuzzy theory, neural networks, nonmonotonic logic, genetic algorithms, and knowledge information theory.

Mr. Takahashi was awarded the Moto-oka Commemorative Award in 1986. He is a member of the Japanese Institute of Electronics, Information, and Communication Engineers, and the Information Processing Society of Japan.