An Evidential Reasoning Approach to Attribute Value Conflict Resolution in Database Integration

Ee-Peng Lim, *Member, IEEE Computer Society*
Jaideep Srivastava, *Member, IEEE Computer Society,*
and Shashi Shekhar, *Member, IEEE Computer Society*

Abstract—Resolving domain incompatibility among independently developed databases often involves uncertain information. DeMichiel [1] showed that uncertain information can be generated by the mapping of conflicting attributes to a common domain, based on some domain knowledge. In this paper, we show that uncertain information can also arise when the database integration process requires information not directly represented in the component databases, but can be obtained through some summary of data. We therefore propose an extended relational model based on *Dempster-Shafer theory of evidence* [2] to incorporate such uncertain knowledge about the source databases. The extended relation uses evidence sets to represent uncertainty in information, which allow probabilities to be attached to subsets of possible domain values. We also develop a full set of extended relational operations over the extended relations. In particular, an extended union operation has been formalized to combine two extended relations using *Dempster's rule of combination.* The **closure** and **boundedness** properties of our proposed extended operations are formulated. We also illustrate the use of extended operations by some query examples.

Index Terms—Attribute value conflict, database integration, semantic heterogeneity, evidential reasoning.

1 Introduction

The increasing need for applications that access data from multiple independent databases has posed a great challenge to the database research community to solve the *data heterogeneity* problem. Chatterjee and Segev [3] define data heterogeneity to be the incompatibility that occurs among similar attributes resulting in the same data being represented differently in different databases. Two types of incompatibilities may occur, namely *structural* and *semantic*. Structural incompatibility arises when attributes are defined differently in different databases, while semantic incompatibility arises when similarly defined attributes have different values/meanings in different databases. The former may be caused by differences in the attributes' *domain, format, units,* and *granularity*. The latter can be caused by *synonyms, homonyms, different coding methods, incomplete information,* etc. Differing values of an attribute called A, of tuples t_1 and t_2, coming from databases DB_1 and DB_2 respectively, can have one of following meanings:

1) *Entity type incompatibility*: Tuples t_1 and t_2 represent instances from different entity types, and it is coincidental that they possess properties represented by A. For example, the *height* of a person is incompatible with the *height* of a building.
2) *Attribute homonym problem*: A represents different properties of the same entity type in DB_1 and DB_2. For example, the attribute *address* of the entity type *Employee* can mean the office address in one database but home address in another.
3) *Entity identification*: t_1 and t_2 represent distinct real world instances of the same entity type.
4) *Attribute value conflict*: t_1 and t_2 represent the same real world instance, and A models the same property in t_1 and t_2, but there is a conflict in the A values stored in the two databases.

The first two cases are schema level incompatibility problems. Several approaches have been developed to resolve them [4], [5] and we do not intend to discuss them in this paper. The solution to both entity identification and attribute value conflict problems requires the use of attributes from the two databases. Solutions to the entity identification problem usually compare attributes between tuples from different relations in order to decide whether they represent the same real world entity [6], [3]. Attribute value conflict resolution needs to be performed only when a pair of tuples (from different databases) representing the same real world entity are found to conflict in some attribute values [7], [1], [8]. In this paper, we assume that entity identification precedes attribute value conflict resolution.

It has been observed that relying on definite and precise semantic information alone to perform integration cannot resolve all data heterogeneity problems. For example, two relations R_A and R_B, storing restaurant information and coming from different databases, may not have any definite values for the *specialty* attribute, see Fig. 2. However, it is possible that some knowledge specific to each database

• *E.-P. Lim is with the School of Applied Science, Nanyang Technological University, Singapore. E-mail: aseplim@ntuix.ntu.ac.sg.*
• *J. Srivastava and S. Shekhar are with the Department of Computer Science, University of Minnesota, Minneapolis, Minn. E-mail: {srivasta, shekhar}@cs.umn.edu.*

Manuscript received Sept. 20, 1995; revised Oct. 16, 1995.
For information on obtaining reprints of this article, please send e-mail to: transkde@computer.org, and reference IEEECS Log Number K96059.

may enable us to determine the range of values for that attribute. With some knowledge about the menu items of each restaurant, different weights can be assigned to the possible specialty values for each tuple in the relations. For example, the restaurant *garden* in R_A may have 20 items in its food menu. Among them, 10 are from Sichuan cuisine and five are from Hunan cuisine. Using a simple voting model, we can assign weights of $\frac{1}{2}$ and $\frac{1}{4}$ to the specialties Sichuan and Hunan, respectively. By explicitly modeling uncertainty, it is now possible to utilize further semantic information to resolve attribute value conflicts. In the last decade, a few approaches have been proposed for the attribute value conflict problem as discussed in Section 1.2. However, approaches that explicitly consider uncertainty have been considered only in the recent past.

In this paper, we use the **Dempster-Shafer theory of evidence** [2] to model the uncertainty faced in resolving the attribute conflicts. We examine the problem of combining the tuples in two sets of relations, each from a distinct database, sharing a relation definition generated based on the global schema. Our approach has been inspired mainly by work in the artificial intelligence and expert systems communities [2], [9], [10], [11]. Essentially, the problem of resolving data heterogeneity between databases can be formulated as the problem of combining evidence supplied by different sources. As a result, the traditional relation concept is extended in the following aspects:

1) the use of evidence sets to model the uncertain attribute values produced by the mapping from actual attribute to virtual attributes, and
2) the introduction of a tuple membership value for each tuple to indicate the support for it being a member of the relation.

In order to perform attribute value conflict resolution on two extended relations, an extended union operation has been defined. Other extended relational operations have also been given for processing queries on the extended relations.

Fig. 1 shows our proposed database integration framework involving entity identification and attribute value conflict resolution. We assume that schema integration has already been performed on the relations R_A and R_B. The knowledge that is useful to entity identification and attribute value conflict resolution is extracted during schema integration. The knowledge includes schema mapping, attribute domain information, and integration methods. Schema mapping establishes correspondences between attributes from different relations. Attribute domain information defines the mapping between attribute values from different domains. Attribute integration methods are specified for deriving the attributes in the integrated relation. Fig. 1 shows that we first preprocess each source relation to make both relations compatible in their attributes. This usually involves mapping the actual attributes from the source relations into virtual attributes of the appropriate domain types. With the tuple matching information provided by entity identification, tuple merging essentially combines the attribute values of matched tuples based on the specified attribute integration methods. It also produces the integrated relation on which users can pose queries.

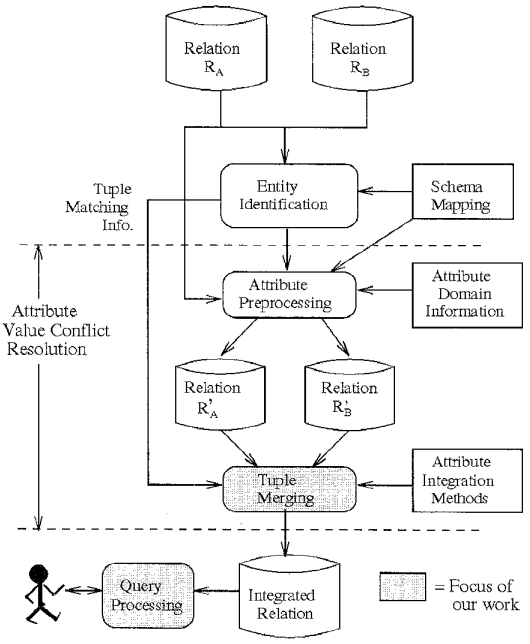


Fig. 1. Entity identification and attribute value conflict resolution framework.

Focus and Scope: In this paper, we focus on the shaded boxes in Fig. 1, i.e., tuple merging and query processing. We assume that the relations to be integrated are identical in their attributes and domains, i.e., attribute preprocessing has been performed. We will examine situations where the preprocessed relations contain uncertain information. Uncertain information may arise mainly because some attributes in the integrated database do not have their direct corresponding attributes in the component databases. The process of deriving them using statistical or history information may introduce uncertainty. We illustrate this using an integration example. To appropriately represent this uncertainty, an extended relational model is introduced. For simplicity, we assume that the preprocessed relations share a common key which determines the matched tuples. An extended relational algebra for uncertain attributes is introduced for merging attributes of matched tuples and for query processing. The **closure** and **boundedness** properties of our extended operations are formulated.

1.1 Example Databases to Illustrate Data Integration

To facilitate our explanation, we adopt the following integration example throughout this paper:

Let DB_A and DB_B be two online databases maintained by two local news agencies, Minnesota Daily and Star Tribune, respectively, for restaurant information in Minneapolis/St.Paul. In order to provide a comprehensive service to future tourists, the Minnesota Tourist Bureau decides to integrate the two databases. Since the information stored in the two databases was collected by independent surveys conducted by the two news agencies, there exists some conflict in the attribute values collected about the same restaurant.

For purely illustrative purposes, we assume that schema

TABLE R_A

rname	street	bldg-no	phone	†speciality	†best-dish	†rating	†(sn,sp)
garden	univ.ave.	2011	371-2155	$[si^{0.5}, hu^{0.25}, \Theta^{0.25}]$	$[d31^{0.5}, \{d35, d36\}^{0.5}]$	$[ex^{0.33}, gd^{0.5}, avg^{0.17}]$	(1,1)
wok	wash.ave.	600	382-4165	$[si^1]$	$[d6^{0.33}, d7^{0.33}, d25^{0.34}]$	$[gd^{0.25}, avg^{0.75}]$	(1,1)
country	plato.blvd	12	293-9111	$[am^1]$	$[d1^{0.5}, d2^{0.33}, \Theta^{0.17}]$	$[ex^1]$	(1,1)
olive	nic.ave.	514	338-0355	$[it^1]$	$[d1^1]$	$[gd^{0.5}, avg^{0.5}]$	(1,1)
mehfil	9th-street	820	333-4035	$[mu^{0.8}, ta^{0.2}]$	$[d24^{0.4}, d31^{0.6}]$	$[ex^{0.8}, gd^{0.2}]$	(0.5,0.5)
ashiana	univ.ave.	353	371-0824	$[mu^{0.9}, \Theta^{0.1}]$	$[d34^{0.8}, d25^{0.2}]$	$[ex^1]$	(1,1)

TABLE RM_A

rname	mname	position	†(sn,sp)
garden	hwang	owner	(1,1)
garden	lim	pub-rel	(1,1)
wok	hwang	owner	(1,1)
country	jim	executive	(1,1)
mehfil	jaideep	executive	(0.5,0.5)

TABLE M_A

mname	phone	†(sn,sp)
hwang	624-7807	(1,1)
lim	625-9631	(1,1)
jim	951-1234	(1,1)
jaideep	625-4012	(1,1)

TABLE R_B

rname	street	bldg-no	phone	†speciality	†best-dish	†rating	†(sn,sp)
garden	univ.ave.	2011	371-2155	$[si^{0.5}, hu^{0.3}, \Theta^{0.2}]$	$[d31^{0.7}, d35^{0.3}]$	$[ex^{0.2}, gd^{0.8}]$	(1,1)
wok	wash.ave.	600	382-4165	$[ca^{0.2}, si^{0.7}, \Theta^{0.1}]$	$[d6^{0.5}, d7^{0.25}, d25^{0.25}]$	$[gd^1]$	(1,1)
country	plato.blvd	12	293-9111	$[am^1]$	$[d1^{0.2}, d2^{0.8}]$	$[ex^{0.7}, gd^{0.3}]$	(1,1)
olive	nic.ave.	514	338-0355	$[it^1]$	$[d1^{0.8}, d2^{0.2}]$	$[gd^{0.8}, avg^{0.2}]$	(1,1)
mehfil	9th-street	820	333-4035	$[mu^1]$	$[d24^{0.1}, d31^{0.9}]$	$[ex^1]$	(0.8,1)

TABLE RM_B

rname	mname	position	†(sn,sp)
garden	hwang	owner	(1,1)
garden	lim	pub-rel	(0.6,1)
wok	hwang	owner	(1,1)
olive	shashi	executive	(1,1)
mehfil	jaideep	executive	(0.8,0.8)

TABLE M_B

mname	phone	†(sn,sp)
hwang	624-7807	(1,1)
lim	625-9631	(1,1)
shashi	625-1234	(0.7,0.9)
jaideep	625-4012	(1,1)

integration has been performed and the databases share a common global schema as shown in Fig. 2. DB_A consists of relations R_A, RM_A, and M_A. DB_B consists of relations R_B, RM_B, and M_B. In this example, the source databases after schema integration contain attributes which may be assigned uncertain values. Attributes which may involve uncertainties are prefixed by "†," e.g., †specialty.

The contents of the database are shown below.[1] Note that an additional attribute (*sn, sp*) has been included in each relation to represent the membership of tuples in the relation. The detailed definition of these relations containing uncertain information is given in Section 2.

Consider Table R_A.[2] It consists of seven attributes among which three attributes, i.e., *best–dish, specialty* and *rating,* may contain uncertain values. Each tuple modeling restaurant has been obtained from some survey information on the restaurant's food and services. In a survey, a panel of

1. To save space, the *speciality* and *rating* attribute values have been abbreviated.

2. For simplicity, we assume that the uncertain attributes of relation R_B are determined in the same way except that a different panel of reviewers may conduct the survey.

six food reviewers examines the food and service provided by each restaurant. Each reviewer then casts one vote in favor of a dish and a vote on the overall rating. The values for the two attributes †*best–dish* and †*rating* are derived by consolidating the voting results. For example, a voting statistics of the reviewers on one restaurant's best dish and rating, together with the consolidated attribute values, are shown below:

VOTE STATISTICS ON BEST DISH

name of dish	number of votes
d1	3
d2	2
d3	1

$\dagger best-dish = [d1^{0.5}, d2^{0.33}, d3^{0.17}]$

VOTE STATISTICS ON RATING

rating	number of votes
excellent	2
good	4

$\dagger rating = [excellent^{0.33}, good^{0.67}]$

The restaurants' *specialty* attribute can be obtained in a similar manner by classifying the items in the restaurant menus. Tuples from DB_A and DB_B can be matched by comparing their common key which is definite, e.g., *rname* is the key used to match tuples in R_A and R_B. The integrated relation contains all the attributes in both local relations.

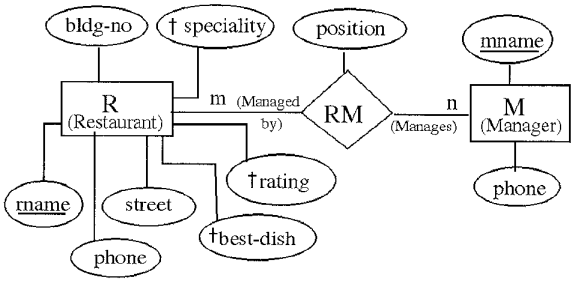


Fig. 2. Example global schema.

1.2 Related Work and Our Contributions

Two kinds of research efforts are related to our proposed attribute value conflict resolution approach, namely

1) other proposed approaches in resolving attribute value conflict, and
2) research in representing uncertain data.

In this subsection, we first describe related work of Type 1, followed by related work of Type 2. Finally we compare and contrast our approach with these efforts.

Several approaches to the attribute value conflict problem have been proposed in the past:

- *Dayal's Aggregate Attributes*: Dayal [12] proposed the use of aggregate functions, e.g., average, maximum, minimum, etc. to resolve discrepancies in attribute values. For instance, if the salary attribute values of record instances in two employee relations do not agree, an average is defined over them to derive the correct salary attribute value for the integrated relation. While aggregate functions [12] are useful in resolving numeric attribute values, our approach is appropriate when an aggregate function cannot be defined over attribute values which are either nonnumeric or uncertain. In this case, we can treat aggregate function approach and our approach as separate classes of attribute integration methods which can coexist in the integration framework (Fig. 1).
- *DeMichiel's Virtual Attributes and Partial Values*: The use of partial values to represent uncertain information from source databases was first proposed by DeMichiel [1]. DeMichiel handles mismatched domains by allowing one-many and many-many mappings between actual attribute values and virtual attributes. When an actual attribute value cannot be mapped into a single definite value, a partial value may result. A *partial value* can be characterized as a set of values of which exactly one must be correct. The combination of two partial values involves removing the noncommon elements.
- *Tseng, Chen, and Yang's Probabilistic Partial Values*: The notion of partial values was generalized by Tseng et al., . to capture uncertainty in attribute values [8]. The possible values of an attribute are listed and given probabilities to indicate their likelihood. Extended selection and join operations are provided to filter out tuples which do not satisfy the query condition with the desired certainty. The possibilities of tuples satisifying a query are given as part of the query result.

In the following, we briefly discuss the relationship between our extended data model and two other related models which have been proposed lately. A more in-depth comparisons will be given in Section 5.2.

- An earlier version of extended relational model, also based on Dempster-Shafer theory has been proposed by Lee [9]. While this model is similar to ours, we have further defined a generalized closed world assumption for interpreting tuples not contained in the extended relation so that query evaluation on our extended relations is finite. To be consistent with this interpretation, our proposed operations have to satisfy the closure and boundedness properties defined in Section 3.6. We have also incorporated Dempster's rule of combination into the extended union operation for the purpose of resolving attribute conflict.
- A probabilistic data model (PDM) has been proposed by Barbara et al., [13] to represent database entities whose properties cannot be deterministically classified. Their model attaches probabilities to the attribute values. However, the model allows probabilities to be assigned only to individual values, and not their subsets. PDM model does not capture tuple membership information. Interestingly, in [13] Barbara et al., discuss the potential need of a *COMBINE* operator to combine two probability distributions of an attribute. We believe

that such an operator has been realized in our model by the use of Dempster's rule of combination.

Contributions: We propose an evidential reasoning approach to resolve attribute value conflict. Our approach is different from the other approaches to attribute value conflict [12], [1] in that it can combine attribute values which contain quantified uncertainties. Furthermore, Dempster's rule of combining uncertainties provides our approach a formal and well founded theory of combining information. Our approach generalizes the partial value concept [1] to capture extra uncertainty information. In DeMichiel's approach, querying relations containing partial values may produce a set of *true* tuples and another set of *may-be* tuples. True tuples are those that definitely qualify as the answers to the query, while may-be tuples are those that may or may not qualify as the answers. With the tuple membership attribute, our model effectively allows a query to return tuples with a full range of certainty. As a result, only a single result set is needed. In precisely the cases that DeMichiel's approach returns *true* tuples, our approach returns tuples with full membership support. There are also some major differences between the probabilistic partial value approach by Tseng et al., [8] and ours. First, our approach, along with DeMichiel's, assumes that source databases provide consistent information, while Tseng's approach does not. As a result, their proposed rule of combining uncertain information is different and the integration result retains inconsistent information. Second, their model does not capture the uncertainty in information related to the membership of tuples within a relation.

1.3 Outline of Paper

This paper is organized as follows. In Section 2, we describe the Dempster-Shafer approach to representing and manipulating uncertain information, and introduce the extended relation concept. We then define our proposed extended relational operations in Section 3. The two important properties of the extended relational operations, namely *closure* and *boundedness* properties, are discussed. We also show that our extended operations correctly extend the standard relational operations. In Section 4, we illustrate our proposed extended relational operations using an extensive query example. Some comments on the proposed model are given in Section 5. Conclusions are given in Section 6.

2 Extended Relation: Representation of Uncertain Information

In this section, we introduce the concept of extended relation, which allows us to represent various forms of uncertain information. This concept is based on the evidential theory by Shafer [2]. We first describe the theoretical foundation of Dempster-Shafer theory and then present the extended relation concept.

2.1 Basic Concepts

We denote the domain of an attribute A by Θ_A which is a set of values A can possibly be assigned. To represent an uncertain A value, *mass values* are assigned to subsets of Θ_A to denote the portions of belief committed to the sets. The function that allocates these probabilities is called the *mass function*(m) [2]. A mass function satisfies the following properties:

$$m(\phi) = 0 \ (\phi \text{ represents empty set})$$
$$\Sigma_{A \subseteq \Theta} m(A) = 1$$

Every subset of the environment which has a mass greater than 0 is a *focal element*, i.e., A is a focal element if $m(A) > 0$.

EXAMPLE. Let $\Theta_{specialty}$ be the set of all possible specialties offered by a restaurant, $\Theta_{specialty}$ = {*american, hunan, sichuan, cantonese, mughalai, italian*}. Let *villagewok* be a Chinese restaurant whose specialty is not completely determined but we may assign mass values to subsets of $\Theta_{specialty}$ as follows:

$$m(\{cantonese\}) = \frac{1}{2}$$
$$m(\{hunan, sichuan\}) = \frac{1}{3}$$
$$m(\{\Theta_{specialty}\}) = \frac{1}{6}$$

The above mass value assignment can be interpreted based on a group voting model. The assignment indicates that half of the dishes on menu are pure Cantonese, and $\frac{1}{3}$ of the dishes on menu are in the set {*hunan, sichuan*}, which cannot be classified as pure Hunan or pure Sichuan. The left over mass value is assigned to $\Theta_{specialty}$ to denote *nonbelief*, representing the fraction of dishes about which no classification information is available. Note that the amount of mass value assigned to a subset of domain values is independent of the size of the subset. For example, in the above mass assignment, m({*cantonese*}) > m({*cantonese, hunan*}) (since m({*cantonese, hunan*}) = 0).

DEFINITION (EVIDENCE SET). *Let Θ_A be the domain of values for an attribute A. An evidence set is a collection of subsets of Θ_A associated with a mass function assignment* [9].

For example, for the restaurant *villagewok*, *ES*1 = [{*cantonese*}$^{1/2}$, {*hunan,sichuan*}$^{1/3}$, $\Theta^{1/6}_{specialty}$] is an evidence set associated with the *specialty* attribute.

The mass function assignment, m, indicates the distribution of belief among the set of possible values in the attribute A of some entity. The m value of a subset of Θ_A is shown as a superscript over the subset. When the subset contains only one element, we may drop the curly brackets for simplicity, e.g., {*cantonese*}$^{0.5}$ can be written as *cantonese*$^{0.5}$. Also, to simplify the notation, we use Θ to denote the appropriate domain of any attribute in the relation. If an evidence set has only one singleton subset assigned with mass value 1, then it represents a definite value (also known as atomic value in relational model).

DEFINITION (BELIEF FUNCTION). *A **belief function**, denoted by Bel, corresponding to a specific mass function m, assigns to every subset A of Θ_A the sum of beliefs committed exactly to every subset of A by m, i.e.,*

$$Bel(A) = \sum_{X \subseteq A} m(X)$$

For example, $Bel(\{cantonese, hunan, sichuan\}) = m(\{cantonese\}) + m(\{hunan\}) + m(\{sichuan\}) + m(\{cantonese, hunan\}) + m(\{cantonese, sichuan\}) + m(\{hunan, sichuan\}) + m(\{cantonese, hunan, sichuan\}) = \frac{1}{2} + 0 + 0 + 0 + 0 + \frac{1}{3} + 0 = \frac{5}{6}$.

The belief function, above, indicates the minimum degree to which $specialty(villagewok) \in \{cantonese, hunan, sichuan\}$, based on the evidence set $ES1$.

DEFINITION (PLAUSIBILITY FUNCTION). *A* ***plausibility function****, denoted by Pls, corresponding to a specific mass function m, determines the maximum belief that can possibly contribute to a subset of A. That is,*

$$Pls(A) = \sum_{A \cap X \neq \phi} m(X) = 1 - Bel(\overline{A})$$

$$\text{where } \overline{A} = \Theta_A - A$$

A plausibility function is defined to indicate the degree to which the evidence set fails to refute a subset A.

For example, $Pls(\{cantonese, hunan, sichuan\}) = m(\{cantonese\}) + m(\{hunan, sichuan\}) + m(\{\Theta_{specialty}\}) = 1$.

Alternatively, $Pls(\{cantonese, hunan, sichuan\}) = 1 - Bel(\{Bel(\overline{\{cantonese, hunan, sichuan\}})\}) = 1 - Bel(\{american, mughalai, italian\}) = 1 - 0 = 1$.

The above plausibility function indicates the maximum degree to which $specialty(villagewok) \in \{cantonese, hunan, sichuan\}$, based on the evidence set $ES1$. In other words, $specialty(villagewok) \in \{cantonese, hunan, sichuan\}$ cannot be disproved based on $ES1$ and is therefore plausible [14].

From the definition, $Bel(A) \leq Pls(A)$. Their difference, $Pls(A) - Bel(A)$ indicates the degree to which the evidence set is uncertain whether to support A or $\overline{A}$.

2.2 Combining Evidence Sets

A mass function is treated as some belief assignment on a domain of values. It is possible to have multiple mass functions on the same domain, which correspond to different evidence sets. Given two evidence sets ES_1 and ES_2, with mass functions m_1 and m_2 respectively, *Dempster's Rule of Combination* can be used to combine them[2]. The *combined mass*, denoted $m_1 \oplus m_2$, is defined as follows:

$$m_1 \oplus m_2(Z) = \sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y)$$

To satisfy the two properties of mass function, normalization may be required to ensure that $m_1 \oplus m_2(\phi) = 0$, and sum of nonzero $m_1 \oplus m_2$ values equals 1. We denote the combined evidence set as:

$$ES_1 \oplus ES_2$$

EXAMPLE. Continuing the example in Section 2.1, we now assume that the mass function m comes from source database DB_1. For clarity, we rename m to m_1. Another source database DB_2 offers a mass function m_2 for the same restaurant entity type, where:

$m_2(\{cantonese, hunan\}) = 1/2$
$m_2(\{hunan\}) = 1/4$
$m_2(\Theta) = 1/4$

The following table shows the intersection of the focal elements associated with the mass functions m_1 and m_2.

	$m_2(\{cantonese, hunan\}) = \frac{1}{2}$	$m_2(\{hunan\}) = \frac{1}{4}$	$m_2(\Theta_{specialty}) = \frac{1}{4}$
$m_1(\{cantonese\}) = \frac{1}{2}$	$\{cantonese\}$ $\frac{1}{4}$	ϕ $\frac{1}{8}$	$\{cantonese\}$ $\frac{1}{8}$
$m_1(\{hunan, sichuan\}) = \frac{1}{3}$	$\{hunan\}$ $\frac{1}{6}$	$\{hunan\}$ $\frac{1}{12}$	$\{hunan, sichuan\}$ $\frac{1}{12}$
$m_1(\Theta_{specialty}) = \frac{1}{6}$	$\{cantonese, hunan\}$ $\frac{1}{12}$	$\{hunan\}$ $\frac{1}{24}$	$\Theta_{specialty}$ $\frac{1}{24}$

In the table, each internal entry is the intersection of a pair of evidence set members. The number attached to the entry is a product of the m_1 and m_2 values of the two evidence set members. The null set, ϕ, occurs because $\{hunan\}$ and $\{cantonese\}$ have no element in common. Since the mass value of a null set has to be zero, a normalization is performed to allocate the mass value $\frac{1}{8}$ to the other focal elements of the combined mass function $m_1 \oplus m_2$. The normalization involves dividing the nonzero $m_1 \cdot m_2$ values by $1 - \kappa$ where

$$\kappa = \sum_{x \cap Y = \phi} m_1(X) \cdot m_2(Y)$$

Since κ in our example is $\frac{1}{8}$, we derive the following $m_1 \oplus m_2$ values for our example:

$$m_1 \oplus m_2(\{cantonese\}) = \left(\frac{1}{4} + \frac{1}{8}\right) / \left(1 - \frac{1}{8}\right) = \frac{3}{7}$$

$$m_1 \oplus m_2(\{hunan\}) = \left(\frac{1}{6} + \frac{1}{12} + \frac{1}{24}\right) / \left(1 - \frac{1}{8}\right) = \frac{1}{3}$$

$$m_1 \oplus m_2(\{cantonese, hunan\}) = \frac{1}{12} / \left(1 - \frac{1}{8}\right) = \frac{2}{21}$$

$$m_1 \oplus m_2(\{hunan, sichuan\}) = \frac{1}{12} / \left(1 - \frac{1}{8}\right) = \frac{2}{21}$$

$$m_1 \oplus m_2(\phi) = 0 \text{ (by the definition of mass function)}$$

$$m_1 \oplus m_2(\Theta_{specialty}) = \frac{1}{24} / \left(1 - \frac{1}{8}\right) = \frac{1}{21}$$

Note that after the combination of evidence sets, the mass value allocated to the set $\{hunan\}$ has increased due to merging larger focal elements, i.e., $\{cantonese, hunan\}$ and $\{hunan, sichuan\}$. The mass value allocated to the set $\{cantonese\}$ has decreased due to conflict in merging the focal elements $\{cantonese\}$ and $\{hunan\}$. It is also a general trend that large focal elements have smaller mass values after the combination. This is due to Dempster's rule which reduces uncertainties after combining uncertain information from two sources.

Considering the normalization step, the general form of Dempster's Rule of Combination is,

$$m_1 \oplus m_2(Z) = \frac{\sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y)}{1 - \kappa}$$

In case none of the focal elements of two mass functions intersect, we use Δ to denote the conflicting information provided by the source databases. Some actions may be necessary to inform the data administrators or integrators about the conflict. Note that the combination rule is both associative and commutative. This implies that the order of combining evidence is not important.

2.3 Extended Relations

Traditional relations capture only precise and certain information. When uncertain information is involved, as in our case of modeling information from difference sources, an extended relation concept is required. In this section, we define an extended relation concept that models the uncertainty within attribute values, as well as the uncertainty about the membership of tuples. This is a step beyond the *partial relation* proposed by DeMichiel [1]. Her partial relation is, in fact, a special case of our extended relation.

Our extended relation differs from the traditional relation in the following ways:

- As we use extended relations to represent entity and relationship instances, each extended relation has definite key values.[3] To represent the properties of entity and relationship instances, *nonkey* attributes are allowed to assume uncertain values. Let $D(A)$ be the domain of a nonkey attribute A. For uncertain attribute A, the A value of a tuple t is an **evidence set**. That is, a collection of subsets of $D(A)$ can be a value for A such that each of these subsets is assigned a **mass** (m) value, i.e.,

$$t.A \subseteq 2^{D(A)}, \text{ and}$$

$$m : t.A \rightarrow [0, 1]$$

Recall that m has to satisfy the following constraint:

$$\sum_{x \in t.A} m(x) = 1$$

- Each extended relation has a *tuple membership attribute* that models the necessary and possible degrees to which a tuple belongs to the relation. Similar to the other nonkey attributes of a tuple, we also assign mass values to the hypotheses about the membership of a tuple in a relation. The domain of tuple membership attribute is the Boolean set Ψ = {true, false}. There are three possible subsets to which mass values can be assigned, namely {true}, {false}, and Ψ. The evidence set for tuple membership can be denoted by a pair of numbers (sn, sp), where:

$$sn = m(\{true\})$$
$$sp = m(\{true\}) + m(\Psi) = 1 - m(\{false\})$$
$$\text{with property } 0 \leq sn \leq sp \leq 1$$

A tuple with (sn, sp) = (1, 1) corresponds to one that is believed to exist with full certainty. A tuple with (sn, sp) = (0, 0) corresponds to one that is believed not to exist with full certainty. A tuple with (sn, sp) = (0, 1) corresponds to complete ignorance about the tuple's membership. The range of legal tuple membership values is shown as the shaded region in Fig. 3

The tuple membership value (sn, sp), may be obtained in two ways. Like other uncertain information, it may be acquired by voting statistics about the tuple's existence in the relation. In many cases, the membership of tuples in a relation is definite. Therefore, their tuple membership values are (1, 1). Nevertheless, during the query evaluation, it is possible that tuples with uncertain tuple membership are generated from relations with definite tuple membership. The uncertain tuple membership arises because some tuples may not qualify fully as part of a query result. In other words, they do not fully satisfy the selection predicate specified in the query.

3. Generalization to uncertain key values is outside the scope of this paper.

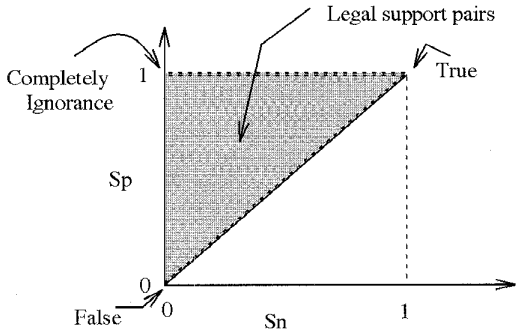


Fig. 3. Tuple membership values (sn, sp).

Generalization of the Closed World Assumption: Traditionally, the *closed world assumption* (*CWA*) is used to model information about entities not represented in a relation. By explicitly assuming that facts not found in a relation are considered to be false, *CWA* provides a means to make query processing finite, since it only has to be performed on the stored database (i.e., the extension). Since tuple membership values in our extended relational model vary in ($0 \leq sn \leq sp \leq 1$), *CWA* needs to be extended to CWA_{ER}, i.e., to "closed world assumption for extended relations." There are two possible ways to generalize *CWA*, namely:

1) "Any tuple not in the database must have $sn = 0$ and $sp = 1$.," i.e., we assume the membership of tuples not in the database to be completely unknown.
2) "Any tuple not in the database must have $sn = 0$.," i.e., tuples not present in the database are assumed to have no necessary support to their existence.

In choosing the first alternative, we would have to store tuples which are completely determined to be a nonmember of a relation. For example, if a restaurant is closed, its tuple must still be maintained in the restaurant relation except that its tuple membership is changed to (0, 0). Since such tuples are usually of no interest to the database users and will be an unnecessary burden to query processing, we choose the latter approach in generalizing the *CWA*. In other words, the integrated database will store information about an entity iff there is some positive evidence to support its membership. Thus, if an entity is not represented in an extended relation, its tuple membership value is (0, sp), such that $sp \leq 1$. Observe that the standard *CWA*, i.e., for regular logic, is a special case of this where $sn = sp = 0$. Thus, our generalization of *CWA* is consistent with its standard meaning. Furthermore, CWA_{ER} also provides finiteness of query processing since, as shown in Section 3.6, the result of query processing on a tuple with $sn = 0$ can never produce a result with $sn > 0$. Thus, query processing on the extension, i.e., stored portion, of an extended relation is sufficient.

By attaching mass values to the subsets of attribute domain, and by allowing a whole range of tuple membership values, we can effectively capture quantitative information about the uncertainty not represented in the *partial values* and *may-be tuples* proposed by DeMichiel [1].

3 Operations on Extended Relations

In this section, we define the operations over the extended relations. We adopt the convention of having a "*" over a relational operator to denote the corresponding extended operator. The new operations differ from the traditional relational operations in several ways:

- The selection/join condition of the operations may be composed of new Boolean predicates on attributes whose values are evidence sets.
- *Membership threshold condition* may be specified within selection/join condition to constrain the number of result tuples.
- The results of extended relational operations either retain or generate new tuple membership values for the result tuples.

3.1 Selection

Our selection operation can involve Boolean predicates more expressive than those allowed by the traditional selection operation, since it is based on logic with support values.

Let R be an extended relation, and $\vec{A}$ be its set of attributes, excluding the tuple membership attribute. We define the extended selection operation as follows:

$$\overset{*Q}{\sigma}_P R \equiv \left\{\left(r.\vec{A}, t_{TM}\right) \middle| r \in R \wedge t_{TM} = F_{TM}\left(r.(sn, sp), F_{SS}(r, P)\right) \wedge Q(t_{TM})\right\}$$ [4]

P : *selection condition* on the attribute value of tuples in R,

$F_{SS}(r, P)$: *selection support function* that returns a (sn,sp) pair indicating the support level of the tuple r for the selection condition P,

F_{TM} : *tuple membership derivation function* that revises the tuple membership

Q : *membership threshold condition* that determines whether a tuple can be included in the result set. The condition is specified on the elements of the support pair produced by F_{TM}

The process of obtaining the new tuple membership of the result extended relation is shown in Fig. 4. We now examine how $F_{SS}(r,P)$, F_{TM}, and Q are evaluated.

3.1.1 Selection Condition

A selection condition is either an *atomic predicate* or a *compound predicate*. The latter is constructed from atomic predicates using conjunction ($\wedge$). An atomic predicate is either an **is-predicate** or **θ-predicate**. The former is of the form A *is* $\{c_1, c_2, \cdots, c_n\}$, and the latter is of the form $A\ \theta B$ where A and

4. Note that the original attribute values are retained in the result. This is different from DeMichiel's approach which modifies the attribute values in the selection operation.

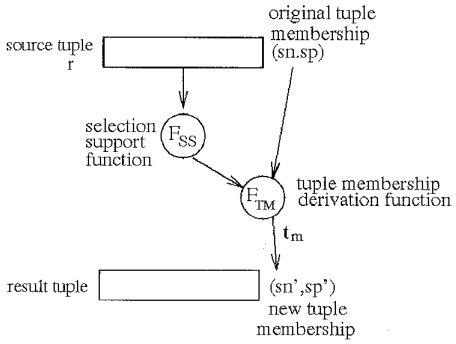


Fig. 4. Process to compute the new tuple membership.

B are evidence sets, $c_i \in \Theta_A$, and $\theta \in \{=, >, <, \leq, \geq\}$. As the attribute values involved in a selection condition may be evidence sets, the degree to which each tuple satisfies the selection condition must be quantified by a support pair. Here, we present an approach to assign support pairs to selection conditions which are atomic predicates. In cases that selection conditions are compound predicates, we adopt a strategy to combine the support pairs of their component atomic predicates.

- **Atomic predicate**:
  is-predicate
  The support of an is-predicate is evaluated based on the degrees to which an evidence set is committed to a given set of domain values. Let P be A **is** $\{c_1, c_2, \cdots, c_n\}$, where A is an evidence set and $c_i \in \Theta_A$. Let $F_{SS}(r, P) = (sn, sp)$. According to Dempster-Shafer theory, F_{SS} is determined as follows:
  $sn = Bel(\{c_1, c_2, \cdots, c_n\})$
  $sp = Pls(\{c_1, c_2, \cdots, c_n\})$

  θ-predicate
  Let P be the predicate $A\ \theta\ B$ where A and B are evidence sets, and $\theta \in \{=, >, <, \leq, \geq\}$.
  Let A be

$$\left[a_1^{m_A(a_1)}, a_2^{m_A(a_2)}, \ldots, a_n^{m_A(a_m)}\right],$$

  and B be

$$\left[b_1^{m_B(b_1)}, b_2^{m_B(b_2)}, \ldots, b_m^{m_B(b_n)}\right]$$

  where $a_i \subseteq \Theta$ and $b_j \subseteq \Theta$.
  Let $F_{SS}(r, P)=(sn, sp)$. The support pair (sn, sp) is computed as follows:

$$sn = \sum_{(a_i \theta b_j \text{ is TRUE})} m_A(a_i) \cdot m_B(b_j)$$

$$sp = \sum_{(a_i \theta b_j \text{ may be TRUE})} m_A(a_i) \cdot m_B(b_j)$$

  Let $a_i = \{a_{i1}, \cdots, a_{iv}\}$ and $b_j = \{b_{j1}, \cdots, b_{jw}\}$
  $(a_i\ \theta\ b_j$ *is TRUE*) if and only if
  $(\forall s \in \{1, \cdots, v\}), (\forall t \in \{1, \cdots, w\}), a_{is}\ \theta\ b_{jt}$.

TABLE $\overset{*sn>0}{\sigma}_{specialty\ is\ \{si\}} R_A$

rname	street	bldg-no	phone	†speciality	†best-dish	†rating	†(sn,sp)
garden	univ.ave.	2011	371-2155	$[si^{0.5}, hu^{0.25}, \Theta^{0.25}]$	$[d31^{0.5}, \{d35, d36\}^{0.5}]$	$[ex^{0.33}, gd^{0.5}, avg^{0.17}]$	(0.5,0.75)
wok	wash.ave.	600	382-4165	$[si^1]$	$[d6^{0.33}, d7^{0.33}, d25^{0.34}]$	$[gd^{0.25}, avg^{0.75}]$	(1,1)

TABLE $\overset{*sn>0}{\sigma}_{(specialty\ is\ \{mu\})\wedge(rating\ is\ \{ex\})} R_A$

rname	street	bldg-no	phone	†speciality	†best-dish	†rating	†(sn,sp)
mehfil	9th-street	820	333-4035	$[mu^{0.8}, ta^{0.2}]$	$[d24^{0.4}, d31^{0.6}]$	$[ex^{0.8}, gd^{0.2}]$	(0.32,0.32)
ashiana	univ.ave.	353	371-0824	$[mu^{0.9}, \Theta^{0.1}]$	$[d34^{0.8}, d25^{0.2}]$	$[ex^1]$	(0.9,1)

TABLE $R_A \dot{\cup}_{(rname)} R_B$

rname	street	bldg-no	phone	†speciality	†best-dish	†rating	†(sn,sp)
garden	univ.ave.	2011	371-2155	$[si^{0.655}, hu^{0.276}, \Theta^{0.069}]$	$[d31^{0.7}, d35^{0.3}]$	$[ex^{0.143}, gd^{0.857}]$	(1,1)
wok	wash.ave.	600	382-4165	$[si^1]$	$[d6^{0.5}, d7^{0.25}, d25^{0.25}]$	$[gd^1]$	(1,1)
country	plato.blvd	12	293-9111	$[am^1]$	$[d1^{0.25}, d2^{0.75}]$	$[ex^1]$	(1,1)
olive	nic.ave.	514	338-0355	$[it^1]$	$[d1^1]$	$[gd^{0.8}, avg^{0.2}]$	(1,1)
mehfil	9th-street	820	333-4035	$[mu^1]$	$[d24^{0.069}, d31^{0.931}]$	$[ex^1]$	(0.83,0.83)
ashiana	univ.ave.	353	371-0824	$[mu^{0.9}, \Theta^{0.1}]$	$[d34^{0.8}, d25^{0.2}]$	$[ex^1]$	(1,1)

($a_i\ \theta\ b_j$ may be *TRUE*) if and only if
$(\exists s \in \{1, \cdots, v\}), (\exists t \in \{1, \cdots, w\}), a_{is}\ \theta\ b_{jt}$.

EXAMPLE. Let P be $([\{1, 4\}^{0.6}, \{2, 6\}^{0.4}] \leq [\{2, 4\}^{0.8}, 5^{0.2}])$, $F_{SS}(r, P) = (sn, sp)$ where
$sn = 0.6 \cdot 0.8 + 0.6 \cdot 0.2 = 0.6$
$sp = 0.6 \cdot 0.8 + 0.6 \cdot 0.2 + 0.4 \cdot 0.8 + 0.4 \cdot 0.2 = 1$

- **Compound predicate:**

  Recall that a compound predicate is formed by a conjunction of two or more atomic predicates. In this paper, we assume that the atomic predicates are mutually independent. A discussion on combining the supports of dependent predicates is given in [10].

  Let S and T be predicates with support values (sn_S, sp_S) and (sn_T, sp_T), respectively. Let P be the compound predicate $S \wedge T$. The support of P, (sn_P, sp_P), is computed based on the multiplicative rule in [10], [15] as shown below:
  $sn_P = sn_S \cdot sn_T$
  $sp_P = sp_S \cdot sp_T$

3.1.2 Tuple Membership Derivation Function

So far, we have defined $F_{SS}(r, P)$, the support for predicate P, based on the attribute values involved in the predicate. The $F_{SS}(r, P)$ of a tuple has to be incorporated into the original tuple membership in order to derive the tuple membership for the result tuple.

An obvious way to interpret the new tuple membership value is that it should reflect the satisfaction of both the predicate P and the membership of the original tuple. We therefore treat the selection predicate and tuple membership as independent events, and define the tuple membership derivation function F_{TM} as follows.

$$F_{TM}((sn_1, sp_1),(sn_2, sp_2)) = (sn_1 \cdot sn_2, sp_1 \cdot sp_2)$$

3.1.3 Membership Threshold Condition

A membership threshold condition is a constraint on the revised tuple membership value of the selection result. In general, it can be query-dependent. However, to be consistent with the interpretation of our extended relations, the membership threshold condition must ensure that the tuple membership values in the selection result satisfy $(sn > 0)$. For example, if we want only tuples that definitely satisfy the selection condition, $(sn = 1)$ can be given as the membership threshold condition. As another example, if we want all tuples that satisfy the selection condition with or without uncertainty, we can give $(sn > 0)$ as the membership threshold condition.

TABLE $R_A^1 \overset{*}{\times} R_A^2$

$\underline{rname}^1$	$street^1$	$bldg-no^1$	$phone^1$	$\dagger speciality^1$	$\dagger best-dish$
garden	univ.ave.	2011	371-2155	$[si^{0.5}, hu^{0.25}, \Theta^{0.25}]$	$[d31^{0.5}, \{d35, d36\}^{0.5}]$
garden	univ.ave.	2011	371-2155	$[si^{0.5}, hu^{0.25}, \Theta^{0.25}]$	$[d31^{0.5}, \{d35, d36\}^{0.5}]$
garden	univ.ave.	2011	371-2155	$[si^{0.5}, hu^{0.25}, \Theta^{0.25}]$	$[d31^{0.5}, \{d35, d36\}^{0.5}]$
⋮	⋮	⋮	⋮	⋮	⋮
ashiana	univ.ave.	353	371-0824	$[mu^{0.9}, \Theta^{0.1}]$	$[d34^{0.8}, d25^{0.2}]$

$\dagger rating^1$	$\underline{rname}^2$	$street^2$	$bldg-no^2$	$phone^2$	$\dagger speciality^2$
$[ex^{0.33}, gd^{0.5}, avg^{0.17}]$	garden	univ.ave.	2011	371-2155	$[si^{0.5}, hu^{0.25}, \Theta^{0.25}]$
$[ex^{0.33}, gd^{0.5}, avg^{0.17}]$	wok	wash.ave.	600	382-4165	$[si^1]$
$[ex^1]$	country	plato.blvd	12	293-9111	$[am^1]$
⋮	⋮	⋮	⋮	⋮	⋮
$[ex^1]$	ashiana	univ.ave.	353	371-0824	$[mu^{0.9}, \Theta^{0.1}]$

$\dagger best-dish^2$	$\dagger rating^2$	†(sn,sp)
$[d31^{0.5}, \{d35, d36\}^{0.5}]$	$[ex^{0.33}, gd^{0.5}, avg^{0.17}]$	(1,1)
$[d6^{0.33}, d7^{0.33}, d25^{0.34}]$	$[gd^{0.25}, avg^{0.75}]$	(1,1)
$[d1^{0.5}, d2^{0.33}, \Theta^{0.17}]$	$[ex^1]$	(1,1)
⋮	⋮	⋮
$[d34^{0.8}, d25^{0.2}]$	$[ex^1]$	(1,1)

EXAMPLE. Consider the extended relation R_A in Section 1. Suppose we want to find the restaurants that specialize in *Sichuan* food. The selection operation ${}_{*}\sigma^{sn>0}_{speciality\ is\ \{si\}} R_A$ is evaluated and its result is shown in Table ${}_{*}\sigma^{sn>0}_{specialty\ is\ \{si\}} R_A$.

EXAMPLE. If we want to know the restaurants (in R_A) which specialize in *Mughalai* food and have been rated *excellent*, the following selection operation with complex predicate and its result is shown in Table ${}_{*}\sigma^{sn>0}_{(specialty\ is\ \{mu\})\wedge(rating\ is\ \{ex\})} R_A$.

3.2 Union

Let R, S be two *union-compatible*[5] extended relations with common key attributes $\vec{K}$ and common non key attributes $\vec{N}$. Let Ψ = {true, false}, and $F((sn_1, sp_1), (sn_2, sp_2)) = (sn, sp)$ where $\left(\text{true}^{sn}, \text{false}^{1-sp}\right) = \left(\text{true}^{sn_1}, \text{false}^{1-sp_1}, \Psi^{sp_1-sp_1}\right) \oplus \left(\text{true}^{sn_2}, \text{false}^{1-sp_2}, \Psi^{sp_2-sp_2}\right)$, where $\oplus$ is the Dempster's evidence combination operator (see Section 2.2.)

5. We say that two extended relations are union-compatible iff they share the same set of attributes including key attribute(s).

$$
\begin{aligned}
R \overset{*}{\cup}_{\vec{K}} S \equiv & \left\{ r \,\middle|\, r \in R \wedge (\nexists s) \quad \left(s \in S \wedge s.\vec{K} = r.\vec{K}\right) \right\} \\
& \cup \left\{ s \,\middle|\, s \in S \wedge (\nexists r) \quad \left(r \in R \wedge s.\vec{K} = r.\vec{K}\right) \right\} \\
& \cup \left\{ t \,\middle|\, (\exists r)(\exists s) \left(r \in R \wedge s \in S \wedge t.\vec{K} = r.\vec{K} = s.\vec{K}\right) \right. \\
& \qquad \wedge (\forall C)\left(C \in \vec{N} \Rightarrow t.C = r.C \oplus s.C\right) \\
& \qquad \left. \wedge \left(t.(sn, sp) = F_{TM}\left(r.(sn, sp), s.(sn, sp)\right)\right) \right\}
\end{aligned}
$$

The extended union operation combines both the attribute values and tuple membership values of matching tuples using Dempster's rule of combination. Note that for a tuple in a relation, whose key value does not match that of any tuple in the other extended relation, we assume that the latter relation has total uncertainty about the membership of the entity modeled by this tuple. Thus, the extended union simply retains the tuple from the first relation in the integrated relation. Like the ordinary union, the extended union is both commutative and associative.

EXAMPLE. The extended union, $R_A \overset{*}{\cup}_{(rname)} R_B$, is shown in Table $R_A \overset{*}{\cup}_{(rname)} R_B$.

3.3 Projection

Let R be an extended relation, and $\vec{A}$ be a set of attributes including the key attributes and the tuple membership at-

tribute. We define the extended projection similar to the conventional projection as follows:

$$\overset{*}{\pi}_{\vec{A}} R \equiv \left[r.\vec{A} \,\middle|\, r \in R \right]$$

The extended projection does not modify the uncertainty information within both the selected attributes and the tuple membership attribute.

EXAMPLE. The projection of *rname, phone, specialty, rating* and tuple membership attributes over R_A is shown in Table $\overset{*}{\pi}_{(rname,\ phone,\ specialty,\ rating,\ (sn,sp))} R_A$.

TABLE $\overset{*}{\pi}_{(rname,phone,speciality,rating,(sn,sp))}\ R_A$

rname	phone	†speciality	†rating	†(sn,sp)
garden	371-2155	$[si^{0.5}, hu^{0.25}, \Theta^{0.25}]$	$[ex^{0.33}, gd^{0.5}, avg^{0.17}]$	(1,1)
wok	382-4165	$[si^{1}]$	$[gd^{0.25}, avg^{0.75}]$	(1,1)
country	293-9111	$[am^{1}]$	$[ex^{1}]$	(1,1)
olive	338-0355	$[it^{1}]$	$[gd^{0.5}, avg^{0.5}]$	(1,1)
mehfil	333-4035	$[mu^{0.8}, ta^{0.2}]$	$[ex^{0.8}, gd^{0.2}]$	(0.5,0.5)
ashiana	371-0824	$[mu^{0.9}, \Theta^{0.1}]$	$[ex^{1}]$	(1,1)

3.4 Cartesian Product

Let R, S be two extended relations with attributes (excluding the tuple membership attribute) $\vec{A}$ and $\vec{B}$ respectively. We define the extended Cartesian product similar to the conventional Cartesian product as follows:

$$\begin{aligned} \overset{*}{R \times S} \equiv \Big\{ \big(t, t.(sn,sp)\big) \,\Big|\, (\exists r)(\exists s)\big(r \in R \wedge s \in S \wedge t.\vec{A} = r.\vec{A} \\ \wedge\ t.\vec{B} = s.\vec{B} \\ \wedge\ t.(sn,sp) = F_{TM}\big(r.(sn,sp), s.(sn,sp)\big)\big) \Big\} \end{aligned}$$

In addition to concatenating all possible pairs of tuples from R and S, the extended Cartesian product also combines the tuple membership attribute of tuple pairs using the tuple membership derivation function F_{TM} as defined in Section 3.1.2.

EXAMPLE. The Cartesian product of Table R_A with itself is shown in Table $R_A^1 \overset{*}{\times} R_A^2$ (to conserve space, only some result tuples are shown; also, due to its width, the table is split into three parts).

3.5 Join

Let R, S be two extended relations, P be the join condition, and Q be the membership threshold condition. We define the extended join as an extended Cartesian product followed by an extended selection.

$$R \overset{*}{\bowtie}{}^{Q}_{P} S \equiv \overset{*}{\sigma}{}^{Q}_{P}\left(R \overset{*}{\times} S \right)$$

3.6 Closure and Boundedness Properties of Extended Relational Operations

As stated in Section 2.3, we have assumed that tuples found in an extended relation R must have at least some positive evidence of their membership, i.e., $sn > 0$. By performing an extended operation on R, we get another extended relation as the result. To produce result relations that are consistent with our interpretation of extended relations, the extended relational operations have to guarantee the *closure property* and *boundedness property*.

Closure Property. Let $\mathcal{R}$ be a list of extended relations, i.e., $\mathcal{R} = (R_1, R_2, \cdots, R_n)$, and o be an $n$ary operator. Now, $\forall t \in o(\mathcal{R}), t.sn > 0$.

Closure property says that given input extended relation(s) that do not contain tuples with $sn = 0$, an extended relational operation on the relation(s) cannot produce tuples with $sn = 0$.

Conceptually, for an extended relation R_i, we can consider its complement $\overline{R}_i$, which has (hypothetical) tuples for all entities about whom R_i has no positive evidence, i.e., $sn = 0$. We can imagine that tuples in $\overline{R}_i$ have unique key values but none of the key values appear in R_i.

Boundedness Property. Let $\mathcal{R}$ be a list of extended relations, i.e., $\mathcal{R} = (R_1, R_2, \cdots, R_n)$, $\mathcal{R} \overset{*}{\cup} \overline{\mathcal{R}}$ be the list $(R_1 \overset{*}{\cup} \overline{R}_1, R_2 \overset{*}{\cup} \overline{R}_2, \ldots, R_n \overset{*}{\cup} \overline{R}_n)$, and o be an $n$ary operator.

$$\left\{ t \,\middle|\, t \in o(\mathcal{R}) \wedge t.sn > 0 \right\} \equiv \left\{ t \,\middle|\, t \in o\left(\mathcal{R} \overset{*}{\cup} \overline{\mathcal{R}} \right) \wedge t.sn > 0 \right\}$$

Boundedness property says that the result of an extended relational operation when applied on some extended relation(s) and its complement(s), and the result of the same operation when applied on the extended relation(s) alone, contains exactly the same set of tuples with $sn > 0$. Now, since the result of query processing, itself being an extended relation, must contain only tuples with $sn > 0$, this means that query processing on $\overline{R}$ can add nothing to the result. This property ensures that query processing remains finite, since it never has to be performed on complements of extended relations.

The following observations are useful in proving the closure and boundedness properties of extended operators.

Observations.

1) If
$$\left(\text{true}^{sn}, \text{false}^{1-sp}, \Psi^{sp-sn}\right) = \left(\text{true}^{sn_1}, \text{false}^{1-sp_1}, \Psi^{sp_1-sn_1}\right) \oplus \left(\text{true}^{sn_2}, \text{false}^{1-sp_2}, \Psi^{sp_2-sn_2}\right),$$
then $[(sn_1 > 0) \wedge (sn_2 > 0) \Rightarrow (sn > 0)]$.
2) If $F_{TM}((sn_1, sp_1), (sn_2, sp_2)) = (sn_3, sp_3)$,
then $(sn_3 = 0) \Leftrightarrow [(sn_1 = 0) \vee (sn_2 = 0)]$.

Both Observations 1 and 2 can be directly verified from their definitions.

LEMMA 1. $\overset{*}{\sigma}$, $\overset{*}{\pi}$, $\overset{*}{\cup}$, $\overset{*}{\times}$, *and* $\overset{*}{\bowtie}$, *satisfy the closure property. In other words, let R and S be a pair of extended relations.*

$$\forall o \in \{\overset{*}{\sigma}, \overset{*}{\pi}\}, \forall t \in o(R), t.sn > 0, \text{ and}$$
$$\forall o \in \{\overset{*}{\cup}, \overset{*}{\times}, \overset{*}{\bowtie}\}, \forall t \in o(R, S), t.sn > 0.$$

PROOF. By its definition, $\overset{*}{\sigma}$ satisfies closure property.

$\overset{*}{\pi}$ satisfies closure property because it preserves the input tuple membership attribute.

$\overset{*}{\cup}$ combines the tuple membership attributes only for the tuples that have the same key values. By Observation 1, the combined tuple membership attribute has $sn > 0$. For those tuples which come from only one extended relation, their tuple membership attribute is preserved. Therefore, $\overset{*}{\cup}$ satisfies closure property.

$\overset{*}{\times}$ satisfies closure property because the tuple membership derivation function F_{TM} used in its definition produces $sn > 0$—by Observation 2.

With $\overset{*}{\sigma}$ and $\overset{*}{\times}$ satisfying the closure property, we also have $\overset{*}{\bowtie}$ satisfying the closure property.

Therefore, Lemma 1 holds. □

LEMMA 2. *$\overset{*}{\sigma}$, $\overset{*}{\pi}$, $\overset{*}{\cup}$, $\overset{*}{\times}$, and $\overset{*}{\bowtie}$, satisfy the boundedness property. In other words, let R and S be any two extended relations.*

$$\forall o \in \left\{\overset{*}{\sigma}, \overset{*}{\pi}\right\}, \left\{t \,|\, t \in o(R) \wedge t.sn > 0\right\} \equiv$$

$$\left\{t \,|\, t \in o\left(R \overset{*}{\cup} \overline{R}\right) \wedge t.sn > 0\right\}$$

$$\forall o \in \left\{\overset{*}{\cup}, \overset{*}{\times}, \overset{*}{\bowtie}\right\}, \left\{t \,|\, t \in o(R,S) \wedge t.sn > 0\right\} \equiv$$

$$\left\{t \,|\, t \in o\left(R \overset{*}{\cup} \overline{R}, S \overset{*}{\cup} \overline{S}\right) \wedge t.sn > 0\right\}$$

PROOF. Since $R \subseteq (R \overset{*}{\cup} \overline{R})$ and $S \subseteq (S \overset{*}{\cup} \overline{S})$, it is clear that

$$\forall o \in \left\{\overset{*}{\sigma}, \overset{*}{\pi}\right\}, \left\{t \,|\, t \in o(R) \wedge t.sn > 0\right\}$$

$$\subseteq \left\{t \,|\, t \in o\left(R \overset{*}{\cup} \overline{R}\right) \wedge t.sn > 0\right\}$$

$$\forall o \in \left\{\overset{*}{\cup}, \overset{*}{\times}, \overset{*}{\bowtie}\right\}, \left\{t \,|\, t \in o(R,S) \wedge t.sn > 0\right\}$$

$$\subseteq \left\{t \,|\, t \in o\left(R \overset{*}{\cup} \overline{R}, S \overset{*}{\cup} \overline{S}\right) \wedge t.sn > 0\right\}$$

Thus, in the following, we focus on showing the inverse is also true.

$\overset{*}{\pi}$ satisfies the boundedness property since $\overset{*}{\pi}$ does not modify the tuple membership values of the original relation. Any tuple in the complement of an input extended relation will appear in the result of the projection operation with $sn = 0$.

We observe that the revised tuple membership values of the $\overset{*}{\cup}$ operation are obtained by the Dempster's rule of combination, i.e., $\oplus$. By Observation 1, for a pair of tuples with membership values $(0, sp_1)$ and $(0, sp_2)$, the combination produces a new tuple with $(0, sp_3)$. Therefore, $\overset{*}{\cup}$ satisfies the boundedness property.

Let R be an extended relation. Every tuple in $\overline{R}$ has $sn = 0$. By Observation 2, the sn remains zero after the concatenation with any other tuple. Thus, $\overset{*}{\times}$ satisfies the boundedness property.

By Observation 2, F_{TM} function in the definition of $\overset{*}{\sigma}$ will not change the sn value of any tuple from $\overline{R}$. Therefore, any tuple from $\overline{R}$ cannot be included into the selection result, and boundness property of $\overset{*}{\sigma}$ is satisfied.

Since $\overset{*}{\times}$ satisfies boundness property,

$$\left\{t \,|\, t \in \left(R \overset{*}{\times} \overline{S}\right) \wedge t.sn > 0\right\} \equiv \left\{t \,|\, t \in \left(\left(R \overset{*}{\cup} \overline{R}\right) \overset{*}{\times} \left(S \overset{*}{\cup} \overline{S}\right)\right) \wedge t.sn > 0\right\}$$

Let $T = \left(R \overset{*}{\cup} \overline{R}\right) \overset{*}{\times} \left(S \overset{*}{\cup} \overline{S}\right)$. Since $\overset{*}{\sigma}\left(T \overset{*}{\cup} \overline{T}\right) = \overset{*}{\sigma}(T), \overset{*}{\bowtie}$ involving the complement relations does not create extra tuples with $sn > 0$, thus satisfying the boundedness property. □

THEOREM 1. *Our extended relational operations satisfy the Closure and Boundedness properties.*

PROOF. This follows from Lemmas 1 and 2. □

3.7 Correctness of Extended Relational Operations

In this subsection, we shows that our proposed extended relational operations are correct. The correctness of the proposed operations is evaluated based on the correct extension of the regular relational operations. We say that a set of operations **correctly extends** a set of relational operations if any query result computed by the relational algebra expression will be equivalent to the one computed by the corresponding extended relational algebra expression. Among the proposed extended operations, $\overset{*}{\cup}$ has been defined specially for the purpose of resolving attribute value conflicts. In contrast to the regular union operation which combines two identical tuples, $\overset{*}{\cup}$ combines nonkey attributes of two tuples with identical key value. Since it is strictly not an extension of the regular union, its correctness with respect to regular relational union will not be discussed.

In Section 2, we define an extended relation to be one that can represent uncertain attribute values and tuple membership. Since an extended relation can also represent definite attribute values and tuple membership, it is possible to represent an ordinary relation as an extended relation. The tuple membership of any tuple in such an extended relation is always (1, 1). On the other hand, given any extended relation, we can induce from it the set of tuples with tuple membership = (1, 1). We call this the **induced extended relation**. From now on, we represent the

induced extended relation of an extended relation R by $I_e(R)$, and the equivalent ordinary relation[6] of $I_e(R)$ by $I_r(R)$.

Let op_e and op_r be an extended relational operation and a relational operation respectively. Let $\mathcal{R}$ be a list of extended relations $(R_1, R_2, ..., R_n)$. Let $I_r(\mathcal{R})$ be the equivalent list of ordinary relations, i.e., $I_r(\mathcal{R}) = (I_r(R_1), ..., I_r(R_n))$. Let $S_e = I_e(op_e(\mathcal{R}))$ and $S_r = op_r(I_r(\mathcal{R}))$. We say that op_e **generalizes** op_r if for any $\mathcal{R}$, $I_r(S_e) = S_r$.

LEMMA 3. *The extended selection operation generalizes the regular selection operation.*

PROOF. Recall that the extended selection is defined as:

$$\overset{*}{\sigma}{}_P^Q R \equiv \left\{ \left(r.\vec{A},\ t_{TM} \right) \middle|\ r \in R \wedge t_{TM} = F_{TM}\left(r.(sn, sp),\ F_{SS}(r, P) \right) \wedge Q(t_{TM}) \right\}$$

Let $S_e = I_e(\overset{*}{\sigma}{}_P^Q R)$ and $S_r = \sigma_P I_r(R)$.[7]

To prove Lemma 3, we need to show that $I_r(S_e) = S_r$.

Suppose $t \in S_e$. By the definition of I_e, $t.(sn, sp) = (1, 1)$. Moreover, $t \in R$ and $F_{TM}(t.(sn, sp), F_{SS}(t, P)) = (1, 1)$. This implies that $F_{SS}(t, P) = (1, 1)$.

By carefully analyzing the definition of F_{SS}, t must fully satisfy the predicate P.

Therefore, t without tuple membership $\in I_r(R)$. Since t fully satisfies P, t without tuple membership can be found in S_r.

Suppose $s \in S_r$. In other words, s with tuple membership $(1, 1) \in R$ and s fully satisfies the predicate P.

Therefore $F_{TM}(s.(sn, sp), F_{SS}(s, P)) = (1, 1)$ and s with tuple membership (1, 1) can be found in S_e.

With the above analysis, we conclude that Lemma 3 holds. □

LEMMA 4. *The extended projection operation generalizes the regular projection operation.*

PROOF. Recall that the extended projection is defined as:

$$\overset{*}{\pi}_{\vec{A}} R \equiv \left\{ r.\vec{A} \middle| r \in R \right\}$$

Let $S_e = I_e\left(\overset{*}{\pi}_{\vec{A}} R \right)$ and $S_r = \pi_{\vec{A}} I_r(R)$. Since $\overset{*}{\pi}$ does not modify the tuple membership at all, a tuple t appears in S_e if and only if it has tuple membership (1, 1) and t without tuple membership should also appear in S_r. Hence, Lemma 4 holds. □

LEMMA 5. *The extended Cartesian product operation generalizes the regular Cartesian product operation.*

PROOF. Recall that the extended Cartesian product is defined as:

$$R \overset{*}{\times} S \equiv \left\{ \left(t, t.(sn, sp) \right) \middle| (\exists r)(\exists s)(r \in r \wedge s \in S \wedge t.\vec{A} = r.\vec{A} \wedge t.\vec{B} = s.\vec{B} \wedge t.(sn, sp) = F_{TM}\left(r.(sn, sp), s.(sn, sp) \right)) \right\}$$

Let $S_e = I_e\left(R \overset{*}{\times} S \right)$ and $S_r = I_r(R) \times I_r(S)$. A tuple t appears in S_e if and only if there exist r and s, from R and S respectively, such that $r.(sn, sp) = s.(sn, sp) = (1, 1)$ (due to the definition of F_{TM}). This can happen if and only if r and s without tuple membership appears in $I_r(R)$ and $I_r(S)$ respectively. Hence, t without tuple membership exists in S_r and Lemma 5 holds. □

With Lemmas 3 and 5, the following corollary holds.

COROLLARY. *The extended join operation generalizes the regular join operation.*

In the above, we have excluded the lemma for extended union since our extended union operation is strictly used for integrating attribute values in contrast with the regular union operation which can only merge tuples with identical attribute values.

THEOREM 2. *Our proposed extended operations correctly extend the relational operations* $\{\sigma, \pi, \times, \bowtie\}$.

PROOF. Due to Lemmas 3, 4, 5, and the above corollary, a query computed by a relational algebra expression consisting of $\{\sigma, \pi, \times, \bowtie\}$ will have a result identical to the one computed by the corresponding extended relational algebra expression cosisting of $\left\{ \overset{*}{\sigma}, \overset{*}{\pi}, \overset{*}{\times}, \overset{*}{\bowtie} \right\}$.

Hence, the theorem holds. □

4 EXTENDED EXAMPLE

In this section, we provide a sample session to illustrate the use of the extended relational model to resolve attribute value conflicts and to process user specified queries.

Consider the integration example in Section 1. Recall that the relations have been preprocessed and common keys between two relations can be used to identify tuples modeling the same real world entities. We are now left with the tuple merging process before arriving at the integrated relations.

4.1 Tuple Merging Process

To merge the tuples from DB_A and DB_B, the extended union operations: $R_A \overset{*}{\cup}_{(rname)} R_B$, $RM_A \overset{*}{\cup}_{(rname,mname)} RM_B$, and $M_A \overset{*}{\cup}_{(mname)} M_B$ are required. Let the three integrated relations be R, RM, and M respectively. Since $R_A \overset{*}{\cup}_{(rname)} R_B$, has been shown in Section 3.2, we will just illustrate the latter two below:

6. That is, the tuple membership attribute is removed.

7. Since the membership threshold condition Q does not exist in the regular selection operation, we prove this lemma under the assumption that Q allows the tuple membership of (1, 1).

TABLE $RM = RM_A \tilde{\cup}_{(rname,mname)} RM_B$

rname	mname	position	†(sn,sp)
garden	hwang	owner	(1,1)
garden	lim	pub-rel	(1,1)
wok	hwang	owner	(1,1)
country	jim	executive	(1,1)
mehfil	jaideep	executive	(0.8,0.8)
olive	shashi	executive	(1,1)

TABLE $M = M_A \tilde{\cup}_{(mname)} M_B$

mname	phone	†(sn,sp)
hwang	624-7807	(1,1)
lim	625-9631	(1,1)
jim	951-1234	(1,1)
jaideep	625-4012	(1,1)
shashi	625-1234	(0.7,0.9)

4.2 Query Example

Suppose we are interested in the manager name and restaurant name for those restaurants which offer mughalai food and are rated as excellent. The algebraic expression of this query is written as:

$$\overset{*}{\pi}_{(RM.rname,mname)}\left(\left(\overset{*}{\sigma}^{(sn>0)}_{(specialty\,is\,\{mu\})\wedge(rating\,is\,\{ex\})} R\right) \overset{*}{\bowtie}_{(rname=rname)} RM\right)$$

The steps taken to evaluate the above query are shown below:

STEP 1 (EXTENDED SELECTION OPERATION). The selection operation on R (see Section 3.2) filters off those tuples which do not have necessary support on either of the two selection predicates. Note that in Table $T_1 = \overset{*}{\sigma}^{(sn>0)}_{(specialty\,is\,\{mu\})\wedge(rating\,is\,\{ex\})} R$, above, the selected tuples have revised tuple membership values.

STEP 2 EXTENDED JOIN OPERATION. We perform extended join as a Cartesian product followed by a selection operation. The restaurant "ashiana" does not have any manager, and is removed during the join. This is illustrated in Table $T_2 = T_1 \overset{*}{\bowtie}_{(rname=rname)} RM$.

STEP 3 EXTENDED PROJECTION OPERATION. The following projection retains the required attributes as well as the orginal tuple membership.

TABLE $T_3 = \tilde{\pi}_{(RM.rname,mname)} T_2$

RM.rname	mname	†(sn,sp)
mehfil	jaideep	(0.66,0.66)

Thus, the result of the example query is contained in relation T_3.

5 COMMENTS ON EXTENDED RELATIONAL MODEL

5.1 Tradeoff Between Scalability and Expressiveness

The extended relational model presented in this paper attempts to represent uncertain information that can arise during data integration. Based on the Dempster-Shafer theory, the model allows an attribute value to be a collection of subsets of the domain such that each subset is assigned a mass value. Since the number of possible subsets of a domain may potentially be large, the computation involving such an attribute value can be time consuming, thus affecting the scalability of this approach. Nevertheless, in a practical situation as suggested by [9], it is possible to place restriction on the size of domain (e.g., restricting the domain size to be two), or the size of the focal elements (e.g., restricting the size of focal elements to be one) to improve the efficiency of manipulating extended relations. Essen-

TABLE $T_1 = \overset{*}{\sigma}^{(sn>0)}_{(speciality\ is\ \{mu\})\wedge(rating\ is\ \{ex\})} R$

rname	street	bldg-no	phone	†speciality	†best-dish	†rating	†(sn,sp)
mehfil	9th-street	820	333-4035	$[mu^1]$	$[d24^{0.069}, d31^{0.931}]$	$[ex^1]$	(0.83,0.83)
ashiana	univ.ave.	353	371-0824	$[mu^{0.9}, \Theta^{0.1}]$	$[d34^{0.8}, d25^{0.2}]$	$[ex^1]$	(0.9,1)

TABLE $T_2 = T_1 \tilde{\bowtie}_{(rname=rname)} RM$

T1.rname	street	bldg-no	phone	†speciality
mehfil	9th-street	820	333-4035	$[mu^1]$

†best-dish	†rating	RM.rname	mname	position	†(sn,sp)
$[d24^{0.069}, d31^{0.931}]$	$[ex^1]$	mehfil	jaideep	executive	(0.66,0.66)

tially, this kind of restriction sacrifices the expressive power of extended relational model in exchange for time efficiency and the appropriate decision must be made with respect to the application domain.

5.2 Relationship Between the Proposed Extended Relational Model and Others

While the extended relational model presented here stems from the Dempster-Shafer theory, it would be interesting to compare it with other extended relational models that are based on probability (e.g., PDM [13]) and fuzzy theory (e.g., [16], [17]). In the probabilistic relational model and also relational model based on Dempster-Shafer theory, the attribute value can assume stochastic values. In the former, a probability is assigned to each possible domain value of an attribute such that the sum of probabilities for all possible domain values = 1 (including the **missing probability** which is the probability assigned to the entire domain due to incompletely specified probability distribution). Fig. 5 depicts an example of a probabilistic relation in the PDM model. In the latter, a mass value is assigned to each possible subset of domain values of an attribute such that the sum of mass values for all possible subsets = 1. By restricting the size of domain subset to one (except the set Θ that represents the entire domain), the latter is reduced to a probabilistic relational model. In other words, our proposed model generalizes the probabilistic relational model. With this generalization, our extended model allows a wide variety of uncertain attribute values that can be represented by some stochastic model.

TABLE *Restaurant*

rname	speciality	best-dish
garden	0.5 [sichuan] 0.25 [hunan] 0.25 [Θ]	0.5 [d31] 0.5 [d32]
country	1 [american]	0.3 [d2] 0.7 [d3]

Fig. 5. Example of PDM relation.

In contrast to our proposed model and the probabilistic model, the fuzzy relational model is based on fuzzy set and possibility theory. Instead of using mass value assignment or probability distribution function, fuzzy relations model their attribute values as fuzzy sets. Each fuzzy attribute value is defined by a possibility distribution function. Moreover, a possibility distribution function μ is defined to map each tuple to a value over [0, 1] to indicate its membership in the relation. Unlike probability or mass value assignment, the sum of all possibility values with respect to an attribute value do not have to be one. An example of a fuzzy relation containing music club member information is shown in Fig. 6. In the relation, the possibility values of Tom having Beethoven, Chopin, and Mozart as his favorite composer are 0.5, 0.7, and 0.2 respectively. Moreover, the possibility value of Tom being a club member is 0.6. Note that the sum of possibility values for Tom's favorite composer is not one. While certain attributes are better modeled as fuzzy sets, there are also attributes which are more appropriately modeled as probabilities or evidence sets. Therefore, an appropriate modeling decision can only be made based on the actual application domain.

TABLE *Music – Club*

name	favorite-composer	μ
tom	0.5 [beethoven] 0.7 [chopin] 0.2 [mozart]	0.6
mark	0.4 [brahms]	1.0

Fig. 6. Example of fuzzy relation.

CONCLUSION

We have presented in this paper an approach, based on the Dempster-Shafer theory of evidence, to resolve attribute value conflict between relations from independently developed databases. We demonstrate that relations modeling both entity and relationship types can be integrated in a uniform manner. An extended relational model has been developed to capture imprecision and uncertainty in information. Our model can capture information about entities whose membership may range from full certainty to totally unknown. An attribute value in general is a collection of subsets of values with some probability assignment. We have also formally defined a set of extended operations that manipulate the extended relations. An extended union operation is given to combine uncertain attribute values using Dempster's rule of combination. A prototype based on our approach has been implemented in Prolog, and its results are reported in [18].

Attribute value conflict resolution is a major task to be dealt with in database integration. In processing a federated database query, attribute value conflict resolution may have to be performed whenever information about real world entities exists in different databases. Our ongoing research is developing mechanisms to do so.

In the following subsections, we list several topics that need to be addressed as future work

6.1 Modeling of Complex Attributes

So far, our extended relation has only considered simple attributes. Each simple attribute has a domain consisting of atomic values. In some cases, the attribute of a class of real world entities can assume some compound value. For example, the attribute Name of a Person entity can made of constituent attributes FIRST_NAME, INITIAL, and LAST_NAME. In other cases, due to interdependency between two attributes, e.g., SALES_RECORD attribute determines BONUS attribute, it is appropriate to treat the two together when the mass values are assigned. Representing and manipulating complex attributes or combinations of

attributes in the extended relational model is a subject of our future research.

6.2 Integration Strategies

In this paper, we introduce Dempster's rule of combination as a formal approach to combine the attribute values. The commutativity and associativity properties of the extended union operator and other algebraic properties of our extended relational operations can allow flexibility in deciding the order of integrating the source relations. For example, to integrate four source relations, each at a different site, we can have two of them integrated by a processor, and another two of them by a different processor. The intermediate relations produced are then further integrated by a third processor. The selection of the minimum cost strategy is an optimization problem.

6.3 Query Language Extension

As suggested by the set of extended relational operations, we propose extensions to existing query languages for posing a query declaratively. We believe that such extensions can be developed based on the SQL language which has become a standard.

6.4 Uncertainty Filtering

It is sometimes useful to derive definite attribute values from the evidence sets for all the relations. For example, we may want to remove the uncertainty from all relations in the integrated database in order to apply the traditional relational operations. We call the operation that performs this a *filter*. Although we may use a series of extended selection operations to construct a filter, care must be taken to prevent the violation of referential integrity in the resulting relation. This situation will arise when a relationship instance is retained while its related entity instance has been filtered. In this case, it is perhaps appropriate to define a filter operation that examines the relationships between entities before the entities are removed.

Acknowledgment

This work was partially supported by the Rome Laboratory of the US Air Force, contract #F30602-91-C-0128.

References

[1] L.G. DeMichiel, "Resolving Database Incompatibility: An Approach to Performing Relational Operations Over Mismatched Domains," *IEEE Trans. Knowledge and Data Eng.*, vol. 1, no. 4, pp. 485-493, 1989.

[2] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, N.J.: Princeton Univ. Press, 1976.

[3] A. Chatterjee and A. Segev, "Data Manipulation in Heterogeneous Databases," *ACM SIGMOD Record*, vol. 20, no. 4, pp. 64-68, Dec. 1991.

[4] R. Elmasri, J. Larson, and S. Navathe, "Schema Integration Algorithms for Federated Databases and Logical Database Design," Technical Report, Honeywell Corporate Systems Development Division, 1986.

[5] J.A. Larson, S.B. Navathe, and R. Elmasri, "A Theory of Attribute Equivalence in Databases With Application to Schema Integration," *IEEE Trans. Software Eng.*, vol. 15, no. 4, pp. 449-463, Apr. 1989.

[6] E.-P. Lim, J. Srivastava, S. Prabhakar, and J. Richardson, "Entity Identification Problem in Database Integration," *Proc. Ninth IEEE Data Eng. Conf.*, pp. 294-301, 1993.

[7] W. Litwin and A. Abdellatif, "Multidatabase Interoperability," *Computer*, vol. 19, no, 12, pp. 10-18, Dec. 1986.

[8] F.S.-C. Tseng, A.L.P. Chen, and W.-P. Yang, "Answering Heterogeneous Database Queries With Degrees of Uncertainty," *Distributed and Parallel Databases*, vol. 1, pp. 281-302, 1993.

[9] S.K. Lee, "Imprecise and Uncertain Information in Databases: An Evidential Approach," *Proc. Eighth IEEE Data Eng. Conf.*, pp. 614-621, 1992.

[10] H.Y. Hau and R.L. Kashyap, "Belief Combination and Propagation in a Lattice-Structured Inference Network," *Trans. Systems, Man, and Cybernetics*, vol. 20, no. 1, pp. 45-57, Feb. 1990.

[11] J.C. Giarratano and G. Riley, *Expert Systems: Principles and Programming*. Boston: PWS-KENT Publishing Co., 1989.

[12] U. Dayal, "Processing Queries Over Generalized Hierarchies in a Multidatabase System," *Proc. VLDB Conf.*, pp. 342-353, 1983.

[13] D. Barbara, H. Garcia-Molina, and D. Porter, "The Management of Probabilistic Data," *IEEE Trans. Knowledge and Data Eng.*, vol. 4, no. 5, pp. 487-502, 1992.

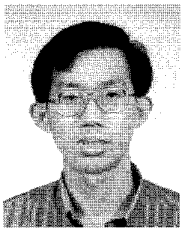
[14] J. Pearl, "Bayesian and Belief-Functions Formalisms for Evidential Reasoning: A Conceptual Analysis," *Readings in Uncertain Reasoning*, G. Shafer and J. Pearl, eds., pp. 540-574. Morgan Kauffmann, 1985.

[15] J.F. Baldwin. "Evidential Support Logic Programming," *Fuzzy Sets and Systems*, vol. 24, pp. 1-26, 1987.

[16] L.A. Zadeh, "Fuzzy Logic," *Computer*, vol. 21, no. 4, pp. 83-93, Apr. 1988.

[17] Y. Takahashi, "Fuzzy Database Query Languages and Their Relational Completeness Theorem," *Trans. Knowledge and Data Eng.*, vol. 5, no. 1, pp. 122-125, 1993.

[18] E-P. Lim, J. Srivastava, and S. Shekhar, "Attribute Value Conflict in Database Integration: An Evidential Reasoning Approach," Technical Report TR93-14, Dept. of Computer Science, Univ. of Minnesota, 1993.



Ee-Peng Lim is a lecturer at the Nanyang Technological University School of Applied Science. His current research interests include database integration, query processing on multidatabase systems, and digital libraries. Dr. Lim received a BS (Honors) degree in information systems and computer science from the National University of Singapore in 1989. He later obtained a PhD in computer science from the University of Minnesota in 1994. He is a member of the IEEE Computer Society and the ACM.

Jaideep Srivastava received the BTech degree in computer science from the Indian Institute of Technology, Kanpur, India, in 1983, and the MS and PhD degrees in computer science from the University of California, Berkeley, in 1985 and 1988, respectively. Since 1988 he has been on the faculty of the Computer Science Department at the University of Minnesota, Minneapolis, where he is currently an associate professor. In 1983, he was a research engineer with Uptron Digital Systems, Lucknow, India.

He has published more than 75 papers in refereed journals and conferences in the areas of databases, parallel processing, artificial intelligence, and multimedia. His current research is in the areas of databases, distributed systems, and multimedia computing. He has given a number of invited talks and participated in panel discussions on these topics.

Dr. Srivastava is a member of the IEEE Computer Society and the ACM. His professional activities have included being on various program committees, and refereeing for journals, conferences, and the National Science Foundation.

Shashi Shekhar received the BTech degree in computer science from the Indian Institute of Technology, Kanpur, India, in 1985, the MS degree in business administration and the PhD degree in computer science from the University of California, Berkeley, in 1989. He is currently an associate professor in the Department of Computer Science at the University of Minnesota, Minneapolis.

His research interests include databases, geographic information systems and intelligent transportation systems. He has published more than 75 papers in journals, books, and conferences, and workshops.

He is currently a editorial board member of the IEEE Computer Society Computer Science and Engineering Practice Board. He is also on the program committees of the IEEE International Conference on Tools with Artificial Intelligence and the ACM Workshop on Advances in GIS. Dr. Shekhar is a member of IEEE Computer Society, ACM, and AAAI.