# A Probabilistic Approach to Query Processing in Heterogeneous Database systems*

Frank Shou-Cheng Tseng
Department of Computer Science and
Information Engineering
National Chiao Tung University
Hsinchu, Taiwan, ROC
*dcp77806@csunix.csie.nctu.edu.tw*

Arbee L. P. Chen
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan, 30043, ROC
*alpchen@cs.nthu.edu.tw*
*Fax: 886-35-723694*

Wei-Pang Yang
Department of Computer Science and
Information Engineering
National Chiao Tung University
Hsinchu, Taiwan, ROC
*wpyang@twnctu01.bitnet*

## Abstract

In heterogeneous database systems, ***partial values*** can be used to resolve the interoperability problems, including domain mismatch, inconsistent data, and missing data. Performing operations on partial values may produce ***maybe tuples*** in the query result which cannot be compared. Thus, users have no way to distinguish which maybe tuple is the most possible answer. In this paper, the concept of partial values is generalized to ***probabilistic partial values.*** We develop a full set of extended relational operators for manipulating relations containing probabilistic partial values. With this approach, the uncertain answer tuples of a query are associated with degrees of uncertainty. That provides users a comparison among maybe tuples and a better understanding on the query results. Besides, extended selection and join are generalized to $\alpha$-selection and $\alpha$-join, respectively, which can be used to filter out maybe tuples with low possibilities — those which have possibilities smaller than $\alpha$.

## 1 Introduction

The advance in communication and database technologies has changed the data processing capabilities tremendously. The proliferation of independent databases implies that for effective information sharing an increasing number of applications are required to access and derive data from various independent databases located in a heterogeneous distributed environment. For independent databases, the data sources are created and developed independently; that is, they are pre-existing in an uncoordinated way without the consideration of future integration with other databases.

There are two approaches to derive data in a heterogeneous database environment. One is to provide a global schema for the independent databases by integrating their schemas. Dayal and Hwang [11] and Motro [20] adopted this approach based on functional model, while Breitbart et al. [5] and Deen et al. [12] were based on relational model. For a comprehensive survey of methodologies developed for schema integration, refer to [2]. The other approach is by providing users a multidatabase query language. Users refer to the schemas and pose their queries against these schemas using the multidatabase query language. Litwin and Abdellatif [18] and Czejdo et al. [10] fell into this category.

To derive data from a heterogeneous database environment, we must first solve the interoperability problems such as data representation conflicts, data scaling conflicts, naming conflicts, missing data, and inconsistent data [4][5]. Among these conflicts, data representation, data scaling and naming conflicts can be considered as the ***domain mismatch*** problems. To resolve the domain mismatch problems in heterogeneous systems, Czejdo et al. [10] used abstract data type to form a ***domain knowledge base. Attribute equivalence*** [16] is another mechanism for resolving these problems. Litwin and Vigier [17] used ***dynamic attributes*** to define one-one and many-one mapping between mismatched domains. This work was later expanded by DeMichiel [13], who used the notions of ***virtual attributes*** and ***partial values*** to provide a general algebraic solution on resolving domain mismatch problems.

The concept of ***partial values*** [14] is a generalization of ***null values*** [9]. Instead of being treated as an atomic value, an attribute value in a table is considered as a nonempty subset of the corresponding domain. A partial value in Grant [14] is represented as an interval such that exactly one of the values in the interval is the "true" value of the partial value. In our work, however, we adopt DeMichiel's definition on partial values [13], that is, a partial value is considered as a finite set of ***possible*** values such that exactly one of the values in that set is the "true" value of the partial value. We discuss the preservation of functional dependency in partial values after a join operation and the manipulation of partial values for a division operation in [21]. Besides, the elimination of redundant partial values is studied in [22].

***Partial values*** can be used to resolve the interoper-

*This work was supported in part by the Republic of China National Science Council under Contract No. NSC 81-0408-E-007-12.

0-8186-2660-7/92 $3.00 © 1992 IEEE

ability problems, including domain mismatch, inconsistent data, and missing data. Performing operations on partial values may produce maybe tuples in the query result which cannot be compared. Thus, users have no way to distinguish which maybe tuple is the most possible answer. In this paper, the concept of partial values is generalized to ***probabilistic partial values***. This work stems from the idea of the probabilistic relational data model proposed by Barbará, Garcia-Molina and Porter [1]. We develop a full set of extended relational operators for manipulating relations containing probabilistic partial values. With this approach, the uncertain answer tuples of a query are associated with degrees of uncertainty. That provides users a comparison among maybe tuples and a better understanding on the query results. Besides, extended selection and join are generalized to $\alpha$-selection and $\alpha$-join, respectively, which can be used to filter out maybe tuples with low possibilities — those which have possibilities smaller than $\alpha$.

In the rest of this paper we will discuss the probabilistic partial values in detail and extend the relational operators to manipulate probabilistic partial values. In the next section, basic concepts and definitions for probabilistic partial values are introduced. Section 3 presents the resolution of the interoperability problems by partial values and an example to show the need for generalizing partial values to probabilistic partial values. Section 4 devotes to the extended relational operators to manipulate relations containing probabilistic partial values. Finally, we conclude and present our future work in Section 5.

## 2 Basic Concepts

*Domain mapping* and *virtual attributes* were used in [13] for the resolution of domain mismatch problems. A domain mapping defines a correspondence (can be one-to-many) between domains of two different attributes. By mapping values in the attributes to ones in the common virtual attribute, relations can be transformed into a union-compatible form suitable for query executions.

When the mapping from one attribute to another is a one-to-many correspondence, a *partial value* can be used to characterize the mapping result. Partial values are formally defined as follows.

Definition 2.1 A *partial value*, denoted $\eta = [a_1, a_2, \ldots, a_n]$, associates with $n$ possible values, $a_1, a_2, \ldots, a_n$, $n \geq 1$, of the same domain, in which exactly one of the values in $\eta$ is the "true" value of $\eta$.

For a partial value $\eta = [a_1, a_2, \ldots, a_n]$, a function $\nu$ is defined by DeMichiel [13], where $\nu$ maps the partial value to its corresponding finite set of *possible* values; that is, $\nu(\eta) = \{a_1, a_2, \ldots, a_n\}$. Notice that an *applicable null value* [9], $\aleph$, can be considered as a partial value with $\nu(\aleph) = D$, where $D$ is the whole domain. In the following, we will use $\eta$ and $\nu(\eta)$ interchangeably when it does not cause confusion. For example, $v \in \eta$ if $v \in \nu(\eta)$.

The *cardinality* of a partial value $\eta$ is defined as $\mid \nu(\eta) \mid$ in [13]. When the cardinality of a partial value equals to 1, i.e., there exists only one ***possible*** value, say $d$, in the partial value, then the partial value $[d]$ actually corresponds to the definite value $d$. On the other hand, a definite value $d$ can be represented as a partial value $[d]$. Besides, a partial value with cardinality greater than 1 is referred as a ***proper partial value*** in [13].

For any two proper partial values, say $\eta_1$ and $\eta_2$, $\eta_1 \neq \eta_2$ even if $\nu(\eta_1) = \nu(\eta_2)$. This is because the ***true*** value of $\eta_1$ may not be the same as the ***true*** value of $\eta_2$.

## 3 Resolving the Interoperability Problems

The rosolution of the interoperability problems are described as follows.

1. *Naming conflicts.* When semantically-related (respectively, semantically-irrelated) data items are named differently (respectively, identically), they are mapped to a canonical virtual attribute (respectively, different virtual attributes).

2. *Data representation conflicts.* This can be resolved by defining a conversion function between the semantically-related attributes and the canonical virtual attribute.

3. *Data scaling conflicts.* This occurs when semantically-related attributes stored in different databases use different units of measure. DeMichiel [13] resolved this problem by partial values when their mapping relationship is one-to-many.

4. *Missing data.* This occurs when relations with different sets of attributes are to be integrated into a global relation. Missing data are all applicable null values which can be represented by the partial values $\aleph$.

5. Inconsistent data. This occurs when semantically-related attributes have different data values in different databases. Partial values can also be used to resolve this problem. That is, these inconsistent data are collected together to form a partial value.

In the following, we give an example to illustrate the resolution of these interoperability problems. This example will also show why we generalize partial values to probabilistic partial values.

Example 3.1
Consider the databases in Figure 1. Site 1 and Site 2 both contain information about computer science researchers in Taiwan, $\sigma_{region=Taiwan}$(**CS-Researchers**) and $\sigma_{specialty=CS}$(**Taiwan-Researchers**), respectively,

which come from different source relations, **CS-Researchers** and **Taiwan-Researchers**, respectively. Suppose there are only three cities in Taiwan, namely Taipei (T), Hsinchu (H), and Kaohisung (K). Besides, Computer Sciences (CS) are assumed to fall into either Artificial Intelligence (AI), Database (DB), or Software Engineering (SE). Then, the attributes *region* and *city* in $\sigma_{region=Taiwan}$(**CS-Researchers**) and $\sigma_{specialty=CS}$(**Taiwan-Researchers**), respectively, are semantically-related but mismatched in their domains. The same situation occurs on the attributes *specialty* and *specialty* in $\sigma_{region=Taiwan}$(**CS-Researchers**) and $\sigma_{specialty=CS}$(**Taiwan-Researchers**), respectively.

To cope with such situations, the relation **CS-2** can be obtained from the relation $\sigma_{region=Taiwan}$(**CS-Researchers**) by applying a domain mapping from the attribute *region* to the virtual attribute **city** with {T, H, K} as its domain.

Analogously, the relation **Taiwan-2** can be derived from $\sigma_{specialty=CS}$(**Taiwan-Researchers**) by mapping the attribute *specialty* to the canonical virtual attribute **specialty** with {AI, DB, SE} as its domain. Figure 2 depicts these two derived relations. Note that since the domain mapping is one-to-many, the mapping result is characterized by partial values.

Now suppose we want to find database researchers in Hsinchu with their ages greater than or equal to 27. We can first "union" **CS-2** and **Taiwan-2** into the relation **Taiwan-CS**. Notice that partial values are used to resolve inconsistent data and missing data. This is shown in Figure 3.

With the derived relation **Taiwan-CS**, our query can be posed as

$$\sigma_{(city=H)\wedge(specialty=DB)\wedge(age\geq 27)}(\textbf{Taiwan-CS})$$

and the tuple depicted in Figure 4 can be obtained as the query result.

Note that the column *status* in Figure 4 does not correspond to an attribute. It shows whether the corresponding tuples are "definite" (with status *true*) or "maybe" (with status *maybe*) [3][8]. For those maybe tuples, we don't know how possible it is a true answer. Therefore, we generalize partial values to probabilistic partial values to provide more informative answer for users. □

We generalize partial values to probabilistic partial values by regarding an attribute as a discrete random variable [1][19]. The probability of an attribute value is therefore a conditional probability depending on the key value of that tuple (key values are assumed definite). To illustrate, consider the following relation, where *name* is the key attribute.

| *name* | *city* | *specialty* | *age* |
|---|---|---|---|
| Jesse | $[T^{0.4}, H^{0.5}, K^{0.1}]$ | SE | 30 |
| Annie | K | $[DB^{0.2}, *^{0.8}]$ | 27 |

This relation describes two entities, "Jesse" and "Annie". The probability that Jesse's city is T (Taipei) is

$$Prob(city = \text{"T"} \mid name = \text{"Jesse"}) = 0.4$$

Note that there is an asterisk '*' with 0.8 probability in the "Annie" tuple. This probability is called a ***missing probability*** and used to specify incomplete probability distributions [1]. In the "Annie" tuple, 0.8 probability has not been assigned to particular specialties. It is assumed that this missing probability is distributed over all ranges in the domain of *specialty*, without any assumptions being made as to how it is distributed. That is,

$$0.2 \leq Prob(specialty = \text{"DB"} \mid name = \text{"Annie"}) \leq 1$$

We define a probabilistic partial value as follows.

Definition 3.1 A ***probabilistic partial value***, denoted $\xi = [a_1^{p_1}, a_2^{p_2}, \ldots, a_n^{p_n}]$, associates with $n$ possible values, $a_1, a_2, \ldots, a_n$, of the same domain $D\cup\{*\}$, where each $a_i$ associates with a probability $p_i \neq 0$ such that $\sum_{i=1}^{n} p_i = 1$.

A relation consisting of tuples with probabilistic partial values is called a ***probabilistic partial relation***. We also use $\nu$ to denote the function that maps a probabilistic partial value to its corresponding finite set of possible values. That is, for a probabilistic partial value $\xi = [a_1^{p_1}, a_2^{p_2}, \ldots, a_n^{p_n}]$, $\nu(\xi) = \{a_1, a_2, \ldots, a_n\}$. Besides, $\nu(\xi)$ and $\xi$ are used interchangeably when it does not cause confusion.

The probabilities of a probabilistic partial value come from the resolution of the interoperability problems. To simplify our discussion, we assume probabilities are ***uniformly distributed*** over all possible values in a partial value. This assumption, however, can be adjusted to fit the specific requirements of different applications. For instance, some applications may assign probabilities according to the timeliness of the possible values. That is, a conflict between a twenty-year-old datum and a recent datum may cause the probability of the old datum to approach to zero and that of the recent one approach to one.

By this approach, the relations in Figure 2 can be modified as Figure 5 depicts. Besides, Figure 3 can be modified as Figure 6 shows. We integrate tuples from different sites by the assumption that they have equal probability on the corresponding conflict attribute values. In Figure 6, for example, the probability of *city* = "T" in the "Andy" tuple is computed as $\frac{1}{3}\times\frac{1}{2}+1\times\frac{1}{2} = \frac{4}{6}$.

Notice that in Figure 6 we resolve the conflicting data *age* in the "Andy" tuple by a missing probability. Therefore, the query result of

$$\sigma_{(city=H)\wedge(specialty=DB)\wedge(age\geq 27)}(\textbf{Taiwan-CS}')$$

can now be specified as Figure 7 depicts. Note that the column *poss* (possibility) in Figure 7 does not correspond to an attribute. This column gives us the possibilities of the answer tuples (the computation of these possibilities will be discussed later) that satisfy the query condition. These quantitative information is used to facilitate qualitative comparisons.

In general, a query result is regarded as a *fuzzy set* [24], where each result tuple is associated with a ***possibility*** that denotes the ***grade*** of its membership in the result set.

# 4 The Extended Relational Operators

In this section, the full set of our extended relational operators will be presented. Our extended relational operators are all marked by a hat ($\widehat{\ }$) over the operator symbols. The possibility of a tuple $t$ is only computed when performing the extended selection and join. The other extended operations all generate the possibility 1 for their result tuples.

If a query $Q$ involves a sequence of extended relational operators, $O_1, O_2, \ldots, O_n$, then the possibility of an answer tuple $t$ of $Q$ is equal to $\prod_{1 \leq i \leq n} O_i(poss(t))$, where $O_i(poss(t))$ is the possibility of $t$ computed from $O_i$.

## 4.1 Selection

An extended selection on a relation $R$ is of the form: $\widehat{\sigma}_P(R)$, where $P$ is a predicate defined by the following grammar specified by the syntax of YACC [23]:

```
$P$ :- $P \vee P$
    | $P \wedge P$
    | $\neg P$
    | Attr $\theta$ Constant /* $\theta \in \{>, <, =, \neq, \leq, \geq\}$ */
    | Attr$_1$ $\theta$ Attr$_2$
    ;
```

Note that Attr, Attr$_1$ and Attr$_2$ are regarded as probabilistic partial values (recall that we can transform a definite value into a probabilistic partial value of cardinality one). The lines of a grammar of YACC syntax are listed in the order of increasing precedence. Given a tuple in a source relation $R$, we define the possibility of an answer tuple of an extended selection to be computed by the following action routines:

```
$P$ :- $P \vee P$ {$\$\$ = max(\$1, \$3);$}
    | $P \wedge P$ {$\$\$ = \$1 \times \$3;$}
    | $\neg P$ {$\$\$ = 1 - \$2;$}
    | Attr $\theta$ Constant {$\$\$ = \sum_{(\forall a \in \$1)(a \theta \$3)} Prob(a);$}
    | Attr$_1$ $\theta$ Attr$_2$
        {$\$\$ = \sum_{(\forall a \in \$1)(\forall b \in \$3)(a \theta b)} [prob(a) \times prob(b)];$}
    ;
```

Note that the "$$" in an action is called a ***pseudo-variable***, which is used to represent a return value of the left-hand-side symbol $P$. Besides, "$\$i$" is used to represent the value associated with the $i$th right-hand-side symbol. For instance, in the first line of this grammar, $P$ :- $P \vee P$, $1 is '$P$', $2 is '$\vee$' and $3 is '$P$'. According to the selection predicate $P$, the answer of $\widehat{\sigma}_P(R)$ is defined as

$$\widehat{\sigma}_P(R) \equiv \{t \mid t \in R \wedge poss_P(t) > 0\},$$

where "$poss_P(t)$" is the possibility of a tuple $t \in R$ computed according to the predicate $P$ by the action routines presented above. The following example illustrates this.

EXAMPLE **4.1** By referring to Figure 6, the answer of the following selection

$$\widehat{\sigma}_{(city=H) \vee (age \geq 27)}(\textbf{Taiwan-CS}')$$

is as Figure 8 shows. Notice that the possibility of the "Andy" tuple is an interval between $\max(\frac{1}{6}, 0) = \frac{1}{6}$ and $\max(\frac{1}{6}, \frac{1}{2}) = \frac{1}{2}$, inclusively.

If the disjunction is changed into conjunction, then the answer is as shown in Figure 9. □

As we can compute the possibility of an answer tuple for an extended selection, a more general selection, call *α-selection*, can be defined as follows.

DEFINITION **4.1** An *α-selection*, denoted $\widehat{\sigma}_P^{\alpha}(R)$, involving a threshold $\alpha$, $0 \leq \alpha \leq 1$, is defined as

$$\widehat{\sigma}_P^{\alpha}(R) \equiv \{t \mid t \in R \wedge poss_P(t) \geq \alpha\},$$

where "$poss_P(t)$" is the possibility of a tuple $t$.

If the query in Example 4.1 is changed into the $\alpha$-selection

$$\widehat{\sigma}^{\frac{1}{3}}_{(city=H) \vee (age \geq 27)}(\textbf{Taiwan-CS}')$$

involving a threshold $\frac{1}{3}$, then the answer is as shown in Figure 10.

The $\alpha$-selection subsumes both the extended selection and the conventional selection. When $\alpha = 1$, $\widehat{\sigma}_P^{\alpha}(R)$ works as the conventional selection $\sigma_P(R)$. When $0 < \alpha \leq min_{t \in \widehat{\sigma}_P(R)}[poss_P(t)]$, it works as the extended selection $\widehat{\sigma}_P(R)$. Notice that when $\alpha = 0$, $\widehat{\sigma}_P(R) = R$. Besides, for any $\alpha_1 \leq \alpha_2$, we have $\widehat{\sigma}_P^{\alpha_2}(R) \subseteq \widehat{\sigma}_P^{\alpha_1}(R)$.

The concept of $\alpha$-selection is similar to the concept of *α-cut* in a fuzzy set [15].

## 4.2 Projection

The projection for a probabilistic partial relation is the same as the conventional projection. Let the source relation be $A(X)$ and $T \subseteq X$. Then $\widehat{\pi}_T(A) \equiv \{t \mid (\exists u)(u \in A \wedge t = u.T)\}$.

## 4.3 Union and Intersection

The extended union is defined on two union-compatible relations $A$ and $B$. Let the source relations be $A(K, N)$ and $B(K, N)$, where $K$ is the key and $N$ the set of non-key attributes. Then $A\widehat{\cup}B$ is defined as follows.

$$\begin{aligned} A\widehat{\cup}B \equiv\ & \{t \mid t \in A \wedge (\nexists u)(u \in B \wedge u.K = t.K)\} \\ & \bigcup \{t \mid t \in B \wedge (\nexists u)(u \in A \wedge u.K = t.K)\} \\ & \bigcup \{t \mid (\exists u)(\exists v)(u \in A \wedge v \in B \\ & \qquad\qquad \wedge t.K = u.K = v.K) \\ & \wedge(\forall C)(C \in N \wedge t.C = u.C \cup v.C \\ & \wedge(\forall e)(e \in t.C \wedge \\ & prob_{t.C}(e) = \tfrac{1}{2}(prob_{u.C}(e) + prob_{v.C}(e)))))\}, \end{aligned}$$

where $prob_x(e)$ is the probability of $e$ in $x$.

Tuples in $A$ and $B$ are collected first. Then, for tuples $t_1 \in A$ and $t_2 \in B$, if the key of $t_1$ is the same as that of $t_2$ then they are unified into a single tuple with the probabilities of their corresponding attribute values being redistributed. Example 4.2 illustrates this.

EXAMPLE 4.2 Consider the following probabilistic partial relations $A$ and $B$:

**A**

| *key* | *A1* | *A2* |
|---|---|---|
| k1 | b | $[x^{0.4}, y^{0.6}]$ |
| k2 | $[a^{0.6}, c^{0.4}]$ | $[w^{0.8}, x^{0.2}]$ |
| k3 | $[b^{0.2}, c^{0.8}]$ | $[x^{0.4}, *^{0.6}]$ |

**B**

| *key* | *A1* | *A2* |
|---|---|---|
| k1 | $[b^{0.4}, c^{0.6}]$ | $[x^{0.1}, y^{0.8}, z^{0.1}]$ |
| k3 | $[a^{0.1}, d^{0.9}]$ | $[x^{0.2}, z^{0.8}]$ |

The result of $A\widehat{\cup}B$ is

$A\widehat{\cup}B$

| *key* | *A1* | *A2* |
|---|---|---|
| k1 | $[b^{\frac{1.0+0.4}{2}}, c^{\frac{0.6}{2}}]$ | $[x^{\frac{0.4+0.1}{2}}, y^{\frac{0.6+0.8}{2}}, z^{\frac{0.1}{2}}]$ |
| k2 | $[a^{0.6}, c^{0.4}]$ | $[w^{0.8}, x^{0.2}]$ |
| k3 | $[a^{\frac{0.1}{2}}, b^{\frac{0.2}{2}}, c^{\frac{0.8}{2}}, d^{\frac{0.9}{2}}]$ | $[x^{\frac{0.4+0.2}{2}}, z^{\frac{0.8}{2}}, *^{\frac{0.6}{2}}]$ |

The extended intersection is defined as follows.

$$\begin{aligned} A\widehat{\cap}B \equiv\ & \{t \mid (\exists u)(\exists v)(u \in A \wedge v \in B \\ & \qquad\qquad \wedge t.K = u.K = v.K) \\ & \wedge(\forall C)(C \in N \wedge t.C = u.C \cup v.C \\ & \quad \wedge(\forall e)(e \in t.C \wedge \\ & \quad prob_{t.C}(e) = \tfrac{1}{2}(prob_{u.C}(e) + prob_{v.C}(e)))))\}, \end{aligned}$$

where $prob_x(e)$ is the probability of $e$ in $x$.

For tuples $t_1 \in A$ and $t_2 \in B$, only when the key of $t_1$ is the same as that of $t_2$ are they unified into a single tuple with the probabilities of their corresponding attribute values being redistributed. This tuple is regarded as an answer tuple of the extended intersection. Therefore, the result of $A\widehat{\cap}B$ contains the "k1" and "k3" tuples of $A\widehat{\cup}B$.

## 4.4 Set Difference

The extended set difference of two probabilistic partial relations $A$ and $B$, $A\widehat{-}B$, is defined as follows. Let the source relations be $A(K, N)$ and $B(K, N)$, where $K$ is the key and $N$ the set of non-key attributes. Then

$$A\widehat{-}B \equiv \{t \mid t \in A \wedge (\nexists u)(u \in B \wedge (t.K = u.K))\}.$$

That is, for any tuple $t \in A$, if the *key* of $t$ is not identical to that of all tuples in $B$, then $t \in A\widehat{-}B$. For the example relations in Example 4.2, $A\widehat{-}B$ contains the "k2" tuple only.

## 4.5 Cartesian Product

The extended cartesian product works as in a conventional system. Let the operand relations be $A(X)$ and $B(Y)$. Then

$$A\widehat{\times}B \equiv \{t \mid (\exists u)(\exists v)(u \in A \wedge v \in B \wedge t.X = u \wedge t.Y = v)\}.$$

## 4.6 Join

The extended join, denoted $A\widehat{\bowtie}_{A_i\theta B_j}B$, where $A_i\theta B_j$ is the join predicate and $A_i$ and $B_j$ are attributes of $A$ and $B$, respectively, is defined by

$$A\widehat{\bowtie}_{A_i\theta B_j}B \equiv \widehat{\sigma}_{A_i\theta B_j}(A\widehat{\times}B) \equiv$$

$$\{t \mid t \in A\widehat{\times}B \wedge poss_{A_i\theta B_j}(t) > 0\},$$

which is consistent with the conventional definition. Therefore, the possibility of an answer tuple of an extended join can be computed by the action routines in the last line discussed in Section 4.1. The following example illustrates this.

EXAMPLE 4.3 Consider the following two relations $A$ and $B$.

**A**

| *key_A* | *A1* |
|---|---|
| *KA1* | $[a^{0.2}, b^{0.3}, c^{0.5}]$ |
| *KA2* | $[b^{0.2}, c^{0.8}]$ |

**B**

| *key_B* | *B1* |
|---|---|
| *KB1* | $[a^{0.3}, c^{0.7}]$ |

Then the result of $A\widehat{\bowtie}_{A1=B1}B$ is

$A\widehat{\bowtie}_{A1=B1}B$

| *key_A* | *A1* | *key_B* | *B1* | poss |
|---|---|---|---|---|
| *KA1* | $[a^{0.2}, b^{0.3}, c^{0.5}]$ | *KB1* | $[a^{0.3}, c^{0.7}]$ | 0.41 |
| *KA2* | $[b^{0.2}, c^{0.8}]$ | *KB1* | $[a^{0.3}, c^{0.7}]$ | 0.56 |

The possibilities of "$KA1$" and "$KA2$" are computed as $0.2 \times 0.3 + 0.5 \times 0.7 = 0.41$ and $0.8 \times 0.7 = 0.56$, respectively. □

Moreover, similar to the $\alpha$-selection, we can define a more general join, called *$\alpha$-join*, as follows.

Definition 4.2 An *α-join*, denoted $A\widehat{\bowtie}^{\alpha}_{P}B$, involving a threshold $\alpha$, $0 \le \alpha \le 1$, is defined as

$$A\widehat{\bowtie}^{\alpha}_{P}B \equiv \widehat{\sigma}^{\alpha}_{P}(A\widehat{\times}B) \equiv \{t \mid t \in (A\widehat{\times}B) \wedge poss_P(t) \ge \alpha\},$$

where "$poss_P(t)$" is the possibility of a tuple $t \in (A\widehat{\times}B)$.

In Example 4.3, if we change $A\widehat{\bowtie}_{A1=B1}B$ into $A\widehat{\bowtie}^{\alpha}_{A1=B1}B$, where $\alpha = 0.5$, then the answer is

$A\widehat{\bowtie}^{0.5}_{A1=B1}B$

| *key_A* | *A1* | *key_B* | *B1* | poss |
|---|---|---|---|---|
| *KA2* | $[b^{0.2}, c^{0.8}]$ | *KB1* | $[a^{0.3}, c^{0.7}]$ | 0.56 |

Analogous to the α-selection, the α-join subsumes both the extended join and the conventional join. When $\alpha = 1$, $A\widehat{\bowtie}^{\alpha}_{P}B$ works as the conventional join $A \bowtie_P B$. When $0 < \alpha \le min_{t \in A\widehat{\bowtie}_P B}[poss_P(t)]$, $A\widehat{\bowtie}^{\alpha}_{P}B$ works as the extended join $A\widehat{\bowtie}_{P}B$. Similarly, when $\alpha = 0$, $A\widehat{\bowtie}^{\alpha}_{P}B \equiv \widehat{\sigma}^{\alpha}_{P}(A\widehat{\times}B) = A\widehat{\times}B$. Besides, for any $\alpha_1 \le \alpha_2$, we have $A\widehat{\bowtie}^{\alpha_2}_{P}B \subseteq A\widehat{\bowtie}^{\alpha_1}_{P}B$.

# 5 Conclusion and Future Work

In this paper, we propose a probabilistic approach to query processing in heterogeneous database systems. To compare our work with that of DeMichiel [13], we believe that a query result with quantitative "possibilities" is more informative than that with just "maybe" status. Moreover, our resolution of the interoperability problems by partial values is more complete.

The query optimization techniques used in a conventional database system need to be reconsidered in a heterogeneous database system. In conventional systems, time-consuming operations (e.g., union and join) are usually performed as late as possible. However, in a heterogeneous system, different relations from different sites may need to be "unioned" and their conflicts resolved before other operations can be performed. We have studied the optimization techniques in heterogeneous database systems in [6][7]. Further work on the optimization techniques based on the probabilistic approach will be pursued.

# References

[1] D. Barbará, H. Garcia-Molina, and D. Porter, A Probabilistic Relational Data Model, *Lecture Notes in Computer Science: Advances in Database Technology — EDBT'90* (Springer-Verlag, NY, 1990) 60-74.

[2] C. Batini, M. Lenzerini, and S.B. Navathe, A Comparative Analysis of Methodologies for Database Schema Intergration, *ACM Computing Surveys* 18 (4) (1986) 323-364.

[3] J. Biskup, A Fundation of Codd's Relational Maybe Operations, *ACM Trans. Database Systems* 8 (4) (1983) 608-636.

[4] Y. Breitbart, P.L. Olson, and G.R. Thompson, Database Intergration in a Distributed Heterogeneous Database System, *Proc. IEEE Int. Conf. Data Eng.* (1986) 301-310.

[5] Y. Breitbart, Multidatabase Interoperability, *SIGMOD RECORD* 19 (3) (1990) 53-60.

[6] A.L.P. Chen, Outerjoin Optimization in Multidatabase Systems, *Proc. IEEE Int. Symposium on Databases in Parallel and Distributed Systems (DPDS)* (1990) 211-218.

[7] A.L.P. Chen, A Localized Approach to Distributed Query Processing, *Lecture Notes in Computer Science: Advances in Database Technology - EDBT'90* (416, Springer-Verlag, Berlin, 1990) 188-202.

[8] E.F. Codd, Extending the Database Relational Model to Capture More Meaning, *ACM Trans. Database Sys.* 4 (4) (1979) 397-434.

[9] E.F. Codd, Missing Information (Applicable and Inapplicable) in Relational Databases, *SIGMOD RECORD* 15 (4) (1986) 53-78.

[10] B. Czejdo, M. Rusinkiewicz, and D.W. Embley, An Approach to Schema Integration and Query Formulation in Federated Database Systems, *Proc. IEEE Int. Conf. Data Eng.* (1987) 477-484.

[11] U. Dayal and H.Y. Hwang, View Definition and Generalization for Database Intergration in a Multidatabase System, *IEEE Trans. Software Eng.* 10 (6) (1984) 628-644.

[12] S.M. Deen, R.R. Amin, and M.C. Taylor, Data Integration in Distributed Databases, *IEEE Trans. Software Eng.* 13 (7) (1987) 860-864.

[13] L.G. DeMichiel, Resolving Database Incompatibility: An Approach to Performing Relational Operations over Mismatched Domains, *IEEE Trans. Knowledge and Data Eng.* 1 (4) (1989) 485-493.

[14] J. Grant, Partial Values in a Tabular Database Model, *Information Processing Letters*, 9 (2) (1979) 97-99.

[15] G.J. Klir and T.A. Folger, *Fuzzy Sets, Uncertainty, and Information* (Prentice-Hall, NJ, 1988).

[16] 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.* 15 (4) (1989) 449-463.

[17] W. Litwin and Ph. Vigier, Dynamic Attributes in the Multidatabase System MRDSM, *Proc. IEEE Int. Conf. Data Eng.* (1986) 103-110.

[18] W. Litwin and A. Abdellatif, An Overview of the Multi-Database Manipulation Language MDSL, *Proc. of the IEEE* 75 (5) (1987) 621-632.

[19] P.L. Meyer, *Introductory Probability and Statistical Applications* (Addison-Wesley, 2nd Ed., 1970).

[20] A. Motro, Superviews: Virtual Intergration of Multiple Databases, *IEEE Trans. Software Eng.* 13 (7) (1987) 785-798.

[21] F.S.C. Tseng, A.L.P. Chen, and W.P. Yang, Deriving Maybe Results from Mismatched Domains in Heterogeneous Distributed Databases, submitted for publication, 1991.

[22] F.S.C. Tseng, A.L.P. Chen, and W.P. Yang, Searching a Minimal Semantically-Equivalent Subset of a Set of Partial Values, submitted for publication, 1991.

[23] *UNIX™ Time-Sharing System: Unix Programmer's Manual* (Bell Lab., 7th Ed., Vol.2B, 1979)

[24] L.A. Zadeh, Fuzzy Sets as a Basis for a Theory of Possibility, *Fuzzy Sets and Sys.*, 1 (1) (1978) 3-28.

$\sigma_{region=Taiwan}$(**CS_Researchers**)

| *name* | *region* | *specialty* | *age* | *degree* |
|---|---|---|---|---|
| Andy | Taiwan | AI | $\aleph$ | MS |
| Frank | Taiwan | DB | 26 | PhD |
| Jesse | Taiwan | SE | 30 | MS |

Site 1

$\sigma_{specialty=CS}$(**Taiwan_Researchers**)

| *name* | *city* | *specialty* | *age* | *affiliation* |
|---|---|---|---|---|
| Andy | T | CS | 25 | NTU |
| Frank | H | CS | 28 | NCTU |
| Annie | K | CS | 27 | NCKU |

Site 2

Figure 1: Example Databases for Example 2.1.

**CS-2**

| *name* | **city** | *specialty* | *age* | *degree* |
|---|---|---|---|---|
| Andy | [T, H, K] | AI | $\aleph$ | MS |
| Frank | [T, H, K] | DB | 26 | PhD |
| Jesse | [T, H, K] | SE | 30 | MS |

Site 1

**Taiwan-2**

| *name* | *city* | **specialty** | *age* | *affiliation* |
|---|---|---|---|---|
| Andy | T | [AI, DB, SE] | 25 | NTU |
| Frank | H | [AI, DB, SE] | 28 | NCTU |
| Annie | K | [AI, DB, SE] | 27 | NCKU |

Site 2

Figure 2: The Derived Relations **CS-2** and **Taiwan-2**.

**Taiwan-CS**

| *name* | *city* | *specialty* | *age* | *degree* | *affiliation* |
|---|---|---|---|---|---|
| Andy | [T, H, K] | [AI, DB, SE] | $\aleph$ | MS | NTU |
| Frank | [T, H, K] | [AI, DB, SE] | [26, 28] | PhD | NCTU |
| Jesse | [T, H, K] | SE | 30 | MS | $\aleph$ |
| Annie | K | [AI, DB, SE] | 27 | $\aleph$ | NCKU |

Figure 3: The Relation **Taiwan-CS** Obtained from "unioning" **CS-2** and **Taiwan-2**.

$\sigma_{(city=H)\wedge(specialty=DB)\wedge(age>27)}$(**Taiwan-CS**)

| *name* | *city* | *specialty* | *age* | *degree* | *affiliation* | status |
|---|---|---|---|---|---|---|
| Andy | [T, H, K] | [AI, DB, SE] | $\aleph$ | MS | NTU | maybe |
| Frank | [T, H, K] | [AI, DB, SE] | [26, 28] | PhD | NCTU | maybe |

Figure 4: The Relation $\sigma_{(city=H)\wedge(specialty=DB)\wedge(age\geq27)}$(**Taiwan-CS**).

**CS-2′**

| *name* | **city** | *specialty* | *age* | *degree* |
|---|---|---|---|---|
| Andy | $[T^{\frac{1}{3}}, H^{\frac{1}{3}}, K^{\frac{1}{3}}]$ | AI | $[*^1]$ | MS |
| Frank | $[T^{\frac{1}{3}}, H^{\frac{1}{3}}, K^{\frac{1}{3}}]$ | DB | 26 | PhD |
| Jesse | $[T^{\frac{1}{3}}, H^{\frac{1}{3}}, K^{\frac{1}{3}}]$ | SE | 30 | MS |

Site 1

**Taiwan-2′**

| *name* | *city* | **specialty** | *age* | *affiliation* |
|---|---|---|---|---|
| Andy | T | $[AI^{\frac{1}{3}}, DB^{\frac{1}{3}}, SE^{\frac{1}{3}}]$ | 25 | NTU |
| Frank | H | $[AI^{\frac{1}{3}}, DB^{\frac{1}{3}}, SE^{\frac{1}{3}}]$ | 28 | NCTU |
| Annie | K | $[AI^{\frac{1}{3}}, DB^{\frac{1}{3}}, SE^{\frac{1}{3}}]$ | 27 | NCKU |

Site 2

Figure 5: The Derived Relations **CS-2′** and **Taiwan-2′** with probabilities.

**Taiwan-CS′**

| *name* | *city* | *specialty* | *age* | *degree* | *affiliation* |
|---|---|---|---|---|---|
| Andy | $[T^{\frac{4}{6}}, H^{\frac{1}{6}}, K^{\frac{1}{6}}]$ | $[AI^{\frac{4}{6}}, DB^{\frac{1}{6}}, SE^{\frac{1}{6}}]$ | $[25^{\frac{1}{2}}, *^{\frac{1}{2}}]$ | MS | NTU |
| Frank | $[T^{\frac{1}{6}}, H^{\frac{4}{6}}, K^{\frac{1}{6}}]$ | $[AI^{\frac{1}{6}}, DB^{\frac{4}{6}}, SE^{\frac{1}{6}}]$ | $[26^{\frac{1}{2}}, 28^{\frac{1}{2}}]$ | PhD | NCTU |
| Jesse | $[T^{\frac{1}{3}}, H^{\frac{1}{3}}, K^{\frac{1}{3}}]$ | SE | 30 | MS | $[*^1]$ |
| Annie | K | $[AI^{\frac{1}{3}}, DB^{\frac{1}{3}}, SE^{\frac{1}{3}}]$ | 27 | $[*^1]$ | NCKU |

Figure 6: The Relation **Taiwan-CS′** Derived from **CS-2′** and **Taiwan-2′**.

$\sigma_{(city=H)\wedge(specialty=DB)\wedge(age>27)}$(**Taiwan-CS′**)

| *name* | *city* | *specialty* | *age* | *degree* | *affiliation* | poss |
|---|---|---|---|---|---|---|
| Andy | $[T^{\frac{4}{6}}, H^{\frac{1}{6}}, K^{\frac{1}{6}}]$ | $[AI^{\frac{4}{6}}, DB^{\frac{1}{6}}, SE^{\frac{1}{6}}]$ | $[25^{\frac{1}{2}}, *^{\frac{1}{2}}]$ | MS | NTU | $0 \leq p \leq \frac{1}{6} \times \frac{1}{6} \times \frac{1}{2}$ |
| Frank | $[T^{\frac{1}{6}}, H^{\frac{4}{6}}, K^{\frac{1}{6}}]$ | $[AI^{\frac{1}{6}}, DB^{\frac{4}{6}}, SE^{\frac{1}{6}}]$ | $[26^{\frac{1}{2}}, 28^{\frac{1}{2}}]$ | PhD | NCTU | $\frac{4}{6} \times \frac{4}{6} \times \frac{1}{2} = \frac{2}{9}$ |

Figure 7: The Relation $\sigma_{(city=H)\wedge(specialty=DB)\wedge(age\geq27)}$(**Taiwan-CS′**).

$\widehat{\sigma}_{(city=H)\vee(age>27)}$(**Taiwan-CS′**)

| *name* | *city* | *specialty* | *age* | *degree* | *affiliation* | poss |
|---|---|---|---|---|---|---|
| Andy | $[T^{\frac{4}{6}}, H^{\frac{1}{6}}, K^{\frac{1}{6}}]$ | $[AI^{\frac{4}{6}}, DB^{\frac{1}{6}}, SE^{\frac{1}{6}}]$ | $[25^{\frac{1}{2}}, *^{\frac{1}{2}}]$ | MS | NTU | $\max(\frac{1}{6},0) \leq p \leq \max(\frac{1}{6},\frac{1}{2})$ |
| Frank | $[T^{\frac{1}{6}}, H^{\frac{4}{6}}, K^{\frac{1}{6}}]$ | $[AI^{\frac{1}{6}}, DB^{\frac{4}{6}}, SE^{\frac{1}{6}}]$ | $[26^{\frac{1}{2}}, 28^{\frac{1}{2}}]$ | PhD | NCTU | $\max(\frac{4}{6},\frac{1}{2}) = \frac{2}{3}$ |
| Jesse | $[T^{\frac{1}{3}}, H^{\frac{1}{3}}, K^{\frac{1}{3}}]$ | SE | 30 | MS | $[*^1]$ | $\max(\frac{1}{3},1) = 1$ |
| Annie | K | $[AI^{\frac{1}{3}}, DB^{\frac{1}{3}}, SE^{\frac{1}{3}}]$ | 27 | $[*^1]$ | NCKU | $\max(0,1) = 1$ |

Figure 8: The Relation $\widehat{\sigma}_{(city=H)\vee(age\geq27)}$(**Taiwan-CS′**)

$\widehat{\sigma}_{(city=H)\wedge(age>27)}$(**Taiwan-CS′**)

| *name* | *city* | *specialty* | *age* | *degree* | *affiliation* | poss |
|---|---|---|---|---|---|---|
| Andy | $[T^{\frac{4}{6}}, H^{\frac{1}{6}}, K^{\frac{1}{6}}]$ | $[AI^{\frac{4}{6}}, DB^{\frac{1}{6}}, SE^{\frac{1}{6}}]$ | $[25^{\frac{1}{2}}, *^{\frac{1}{2}}]$ | MS | NTU | $\frac{1}{6} \times 0 \leq p \leq \frac{1}{6} \times \frac{1}{2}$ |
| Frank | $[T^{\frac{1}{6}}, H^{\frac{4}{6}}, K^{\frac{1}{6}}]$ | $[AI^{\frac{1}{6}}, DB^{\frac{4}{6}}, SE^{\frac{1}{6}}]$ | $[26^{\frac{1}{2}}, 28^{\frac{1}{2}}]$ | PhD | NCTU | $\frac{4}{6} \times \frac{1}{2} = \frac{1}{3}$ |
| Jesse | $[T^{\frac{1}{3}}, H^{\frac{1}{3}}, K^{\frac{1}{3}}]$ | SE | 30 | MS | $[*^1]$ | $\frac{1}{3} \times 1 = \frac{1}{3}$ |

Figure 9: The Relation $\widehat{\sigma}_{(city=H)\wedge(age\geq27)}$(**Taiwan-CS′**)

$\widehat{\sigma}^{\frac{1}{3}}_{(city=H)\wedge(age>27)}$(**Taiwan-CS′**)

| *name* | *city* | *specialty* | *age* | *degree* | *affiliation* | poss |
|---|---|---|---|---|---|---|
| Frank | $[T^{\frac{1}{6}}, H^{\frac{4}{6}}, K^{\frac{1}{6}}]$ | $[AI^{\frac{1}{6}}, DB^{\frac{4}{6}}, SE^{\frac{1}{6}}]$ | $[26^{\frac{1}{2}}, 28^{\frac{1}{2}}]$ | PhD | NCTU | $\frac{4}{6} \times \frac{1}{2} = \frac{1}{3}$ |
| Jesse | $[T^{\frac{1}{3}}, H^{\frac{1}{3}}, K^{\frac{1}{3}}]$ | SE | 30 | MS | $[*^1]$ | $\frac{1}{3} \times 1 = \frac{1}{3}$ |

Figure 10: The Relation $\widehat{\sigma}^{\frac{1}{3}}_{(city=H)\wedge(age\geq27)}$(**Taiwan-CS′**)