Semantics of an Extended Relational Model for Managing Uncertain Information

V. S. Alagar *
Concordia University
Montreal, Canada
alagar@cs.concordia.ca

F. Sadri †
University of North Carolina at Greensboro
Greensboro, NC
sadri@cs.uncg.edu

J. N. Said ‡
Concordia University
Montreal, Canada
said@cs.concordia.ca

Abstract

The Information Source Tracking (IST) method has been recently developed for the modeling and manipulation of uncertain and inaccurate data. In this paper we extend the IST method to model uncertain and inaccurate information at the finest granularity of data in the relational model, namely at the attribute value level. An extended relational model is proposed in which each attribute value in a tuple in an extended relation is associated with an information source vector showing the sources (observers) that contribute to that attribute value, and the nature of their contribution. We will discuss how the relational algebra operations can be extended and implemented to trace the information sources that correspond to each attribute/tuple in the answer to a query. We also present a semantic interpretation of our extended relational model, and prove that the extended relations are "correct" with respect to the semantic model.

1 Introduction

The modeling and management of uncertain and inaccurate information has attracted considerable interest in recent years [1, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. The "Information Source Tracking Method", or IST, is one of the techniques proposed recently for this purpose [17]–[21]. Information in a database system could come from different sources, where every source has a certain reliability associated with it. In the IST approach the reliability of the information is based on the reliability of the contributing sources of information. Each tuple in an extended relation is associated with an information source vector indicating the sources that have contributed to the tuple and their nature of contribution. Query processing is based on extended relational algebra operations that propagate the source information to the answers, identifying the sources contributing to the answer as well as their nature of contribution. Reliability calculation algorithms [17, 18] are used to calculate the reliability of the answers to a query as a function of the reliability of contributing sources.

*supported by grants from NSERC (Canada) and FCAR (Quebec).
†supported by grants from NSERC (Canada), FCAR (Quebec), and the University of North Carolina at Greensboro
‡supported by grants from NSERC (Canada) and FCAR (Quebec).

In this paper we present an important extension to the IST method to enable the modeling and management of uncertainty at the attribute level. The IST method can be used in the modeling and implementation of many areas of strategic importance such as defense and medicine. In such systems, observers might not contribute to a whole tuple stored in an extended relation; however, an observer might contribute to some attribute values that constitute part of that tuple. In the IST approach the observer is forced to contribute to the whole tuple provided to the uncertain database instead of contributing to the individual attribute values that constitute that tuple. For this reason, we are in need for a new model that gives the observers the flexibility to contribute to any attribute value of any tuple in the extended relations.

An extended relational model is proposed in this paper in which each attribute in the extended relation is associated with an information source vector showing the sources (observers) that contribute to that attribute. Moreover, each tuple in an extended relation is associated with an information source vector showing the contributing sources and indicating the condition under which each tuple exist in the extended relation. We will discuss how the relational algebra operations can be extended and implemented to trace the information sources that correspond to each attribute/tuple in the answer to a query. We also present a semantic interpretation of our extended relational model, and prove that the extended relations are "correct" with respect to the semantic model. will not discuss how to allocate reliabilities to the contributing information sources. This issue is of significance and is addressed elsewhere [4, 5]. We assume that information sources have predefined reliabilities so that we can calculate the reliability of answers to user's queries.

The main contribution of this work is to extend the IST method to model uncertain and inaccurate information at the finest granularity of data in the relational model, namely at the attribute value level. This extension is a huge step towards realizing intelligent database systems capable of handling uncertain and inaccurate information in complicated application domains. This paper is organized as follows. the extended model is presented in Section 2, and the extended relational algebra operations are discussed in Section 3. In Section 4 we present the semantic model, and prove the correctness of the extended operations. Finally, conclusions and future directions are presented in Section 5.

2 The Model

A relation scheme in the classical relational model is simply a set of attributes. In our extended model, we associate an "information source attribute" I_i with each attribute A_i in a relation to record the sources contributing to the value of A_i. In addition, we use a special information source attribute I_0 to record the conditions under which a tuple is valid [1]. Formally, an extended relation scheme is

$R = \{I_0, (A_1, I_1), \ldots, (A_n, I_n)\}$

where A_i, $i = 1, \ldots, n$, are regular attributes, and I_i, $i = 0, \ldots, n$, are information source attributes. I_0 is the information source attribute for the tuple, while I_i is the information source attribute corresponding to the regular attribute A_i, for $i = 1, \ldots, n$.

An extended relation instance r is a finite subset of $D_{I_0} \times D_{A_1} \times D_{I_1} \times \ldots \times D_{A_n} \times D_{I_n}$, where $D_{A_1}, \ldots, D_{A_n}$ are the domains of the attributes $A_1, \ldots, A_n$respectively, and D_{I_0}, $\ldots, D_{I_n}$ are the domains of information source attributes $I_0, \ldots, I_n$, respectively (to be discussed shortly).

Information Source Vectors

The domains of information source attributes consists of vectors of length k, where k is a fixed integer representing the number of information sources. The elements of these vectors are taken from the set $\{0, -1, +1, \top\}$, and are similar in spirit to the four-valued logic of Belnap [2]. Their interpretation is as follows. Given a tuple $t@U$, the component u_i of U records the information sources contributing to the value of the attribute A_i in t as follows. Let $u_i = (a_1, a_2, \ldots, a_k)$, the value of a_j specifies the manner of the contribution (the *role*) of the information source s_j to the value of the attribute A_i of t as follows:

$a_j = 0$ means that the source s_j neither confirms nor denies the value,
$a_j = -1$ means that the source s_j denies the value,
$a_j = +1$ means that the source s_j confirms the value,
$a_j = \top$ means that the source s_j is inconsistent with respect to the value.

The constants 0 and $\top$ denote under-specification and over-specification, respectively. The case of $a_j = \top$ normally does not happen for base (stored) relations. It can happen for views or answers to a query, and indicates a situation where an information source s_j appears to confirm and at the same time deny an attribute value. We will see how this can happen in the next section.

To make the interpretation of source vectors precise, we define the *expression* associated with a source vector $u = (a_1, \ldots, a_k)$ as follows. We associate a boolean variable f_i with each information source $s_i, i = 1, \ldots, k$. The set of information sources that are contributing positively to u is $S^+ = \{s_i \mid a_i = +1 \text{ or } a_i = \top\}$. Similarly, the set of information sources that are contributing negatively to u is $S^- = \{s_i \mid a_i = -1 \text{ or } a_i = \top\}$. The expression of u is:

$e(u) = \bigwedge_{s_i \in S^+} f_i \bigwedge_{s_i \in S^-} \neg f_i$

The value of the attribute A_i of a tuple $t@U$ is valid if and only if $e(u_i) = true$, where u_i is the i-th component of U (which corresponds to A_i).

We extend the definition of the expression for a source vector to a set $x = \{v_1, \ldots, v_m\}$ of source vectors as: $e(x) = \bigvee_{v \in x} e(v)$.

[1] We have introduced this "tuple source attribute" with an eye towards extending our model to object-oriented databases. In a relational environment, a combination of source attributes associated with the key of a relation can play the role of I_0.

Source Vector Operations

For query processing in our extended model, we use the source vector operations *sv-disjunction*, *sv-conjunction*, and *sv-negation* to implement extended relational algebra operations.

The *sv-disjunction* ($\overset{s}{\vee}$) of two sets of source vectors x and y, written $x \overset{s}{\vee} y$, is their union $x \cup y$.

Given two source vectors $v = (a_1, \ldots, a_k)$ and $w = (b_1, \ldots, b_k)$, their *sv-conjunction* $u = v \overset{s}{\wedge} w$ is a source vector $u = (c_1, \ldots, c_k)$, where c_i is calculated based on the table below:

a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i	a_i	b_i	c_i
0	0	0	+1	0	+1	−1	0	−1	$\top$	0	$\top$
0	−1	−1	+1	−1	$\top$	−1	−1	−1	$\top$	−1	$\top$
0	+1	+1	+1	+1	+1	−1	+1	$\top$	$\top$	+1	$\top$
0	$\top$	$\top$	+1	$\top$	$\top$	−1	$\top$	$\top$	$\top$	$\top$	$\top$

The *sv-conjunction* of two sets of source vectors x and y is $x \overset{s}{\wedge} y = \{v \overset{s}{\wedge} w \mid v \in x \text{ and } w \in y\}$

The *sv-negation* ($\overset{s}{\neg}$) of a source vector $v = (a_1, \ldots, a_k)$, written $\overset{s}{\neg}(v)$, is defined as follows: let u_i denote the source vector $(b_1, b_2, \ldots, b_k)$, where $b_i = +1$ and $b_j = 0$, for $j \neq i$, and similarly let w_i denote the source vector $(b_1, b_2, \ldots, b_k)$, where $b_i = -1$ and $b_j = 0$ for $j \neq i$, then

$\overset{s}{\neg}(v) = \{u_i \mid a_i = -1 \text{ or } a_i = \top\} \cup \{w_i \mid a_i = +1$ or $a_i = \top\}$. The sv-negation of a set $x = \{v_1, \ldots, v_m\}$ of source vectors is calculated as follows:

$\overset{s}{\neg}(x) = \overset{s}{\neg}(v_1) \overset{s}{\wedge} \overset{s}{\neg}(v_2) \overset{s}{\wedge} \ldots \overset{s}{\wedge} \overset{s}{\neg}(v_m)$

Note that the definition of source vectors in this paper is more general than that in [17] [18]. However, their results can be extended to our case. The following theorems were proven in [17], where x, y, and z are sets of source vectors and $e(x), e(y)$, and $e(z)$ are their corresponding expressions.

Theorem 1 *Let $z = x \overset{s}{\vee} y$. Then $e(z) = e(x) \vee e(y)$.*

Theorem 2 *Let $z = x \overset{s}{\wedge} y$. Then $e(z) = e(x) \wedge e(y)$.*

Theorem 3 *Let $z = \overset{s}{\neg}(x)$. Then $e(z) = \neg(e(x))$.*

Notation

We will be using the following notation to represent extended tuples in the remainder of the paper. Let τ be a tuple on the extended scheme $R = \{I_0, (A_1, I_1), \ldots, (A_n, I_n)\}$. Then τ can be written as $(u_0, (a_1@u_1), \ldots, (a_n@u_n))$, where $a_1, \ldots, a_n$ are the attribute values for $A_1, \ldots, A_n$, and $u_0, \ldots, u_n$ are the source vectors corresponding to the source attributes $I_0, \ldots, I_n$.

If the details of values are not important, we write $\tau = t@U$, where t, called the *pure* component of τ, is the restriction of τ to regular attributes, namely, $t = (a_1, \ldots, a_n)$, and U is the tuple of source vectors $(u_0, \ldots, u_n)$. In general, a pure component t may appear more than once in an extended relation, associated with different values for source attributes. We will denote this situation by writing $t@\mathcal{U} \in r$, where $\mathcal{U} = \{U_1, \ldots, U_p\}$ is the associated set of source vectors.

Well-Formed Tuples

Our model permits the modeling of the uncertainty of a tuple independent of the modeling of the uncertainties of individual attribute values of the tuple. An extended tuple is *well-formed* if attribute values cannot be valid if the tuple as a unit is not. More precisely, $\tau = (u_0, (a_1@u_1), \ldots, (a_n@u_n))$ is well-formed if and only if $e(u_j) \rightarrow e(u_0)$ for all $j = 1, \ldots, n$. An extended relation r is *well-formed* if all of its tuples are.

Given an extended relation r (not necessarily well-formed) we can derive a well-formed relation r' by replacing the attribute source vectors in each tuple by their sv-conjunction with the corresponding tuple source vector. that is

$$r' = \{(u_0, (a_1@(u_1 \overset{s}{\wedge} u_0)), \ldots, (a_n@(u_n \overset{s}{\wedge} u_0))) \mid (u_0, (a_1@u_1), \ldots, (a_n@u_n)) \in r\}$$

Lemma 1 *The extended relation r' obtained as above is well-formed.*

Proof: *Follows from Theorem 2. If $e(u_i \overset{s}{\wedge} u_o)$ is* true*, then $e(u_i) \wedge e(u_0)$ is* true*. Consequently $e(u_0)$ is* true.

We will assume that all database relations are well-formed. Our extended relational algebra operations, to be discussed in the next section, preserve well-formedness. That is, the result of an extended relational operation on operands that are well-formed relations will be a well-formed relation.

3 Extended Relational Algebra Operations

Query processing in our extended model is accomplished using extended relational algebra operators discussed in this section. A query can be specified in any database language, such as SQL, which is converted into a relational algebra expression. Like regular relational database system, the query processor accepts the query and uses extended relations as operands, and computes an extended relation as the answer to the query. Calculating the reliabilities is the last step of query processing and is based on the algorithms given by Sadri [17, 18].

3.1 Extended Selection (σ')

Let r and r' be the extended relations defined by $r' = \sigma'_F(r)$, where F is a formula (selection condition) formed in the usual way from simple selection formulas using logical connectives $\wedge$ (for AND), $\vee$ (for OR), and $\neg$ (for NOT). A simple selection formula is formed by operands (constants or regular attributes), and arithmetic comparison operators $<, \leq, >, \geq, =, \neq$. Note that special source attributes are invisible to the users and cannot be used in queries [2].

We assume that the formula F (selection condition) is converted to disjunctive form, $F = F_1 \vee \ldots \vee F_q$, where each F_i is a conjunction of simple formulas. If a tuple $t@U \in r$ satisfies F, it satisfies at least one F_i. Let $F_{t_1}, \ldots, F_{t_p}$ be the conjuncts satisfied by t. One tuple $t@U \in r$ gives rise to p tuples $t@U'_1, \ldots, t@U'_p \in r'$, where p is the number of conjuncts satisfied by t. The source vectors U'_j corresponding to the conjunct F_{t_j} can be computed as follows.

Let $U = u_0, \ldots, u_n$ and $U'_j = (u'_{j_0} \cdots u'_{j_n})$, i.e. the components of U corresponding to $I_0, \ldots, I_n$ are $u_0, \ldots, u_n$, respectively, and similarly fot U'_j. Define $\mathcal{A}_j = \{A \mid$ attribute A appears in F_{t_j} $\}$, and

$$\mathcal{C}_j = (\bigwedge_{A_l \in \mathcal{A}_j} u_l)$$

Then, $u'_{j_0} = u_0 \wedge \mathcal{C}_j$, and $u'_{j_i} = u_i \wedge u'_{j_0}$, for $i = 1, \ldots, n$.

Example 1 Consider the selection $r'_1 = \sigma'_F r_1$, where F is the formula ($Address =$ "St. Paul") OR ($DOB >$ "1970" AND $DOB <$ "1976").

Relation r_1						
I_0	*Name*	I_1	*Address*	I_2	*DOB*	I_3
10000	John	10000	St. Paul	11000	1956	10010
11000	Mark	11000	St. John	11010	1978	11010
00011	Peter	00011	St. Paul	00111	1975	01011

The resulting extended relation is represented as follows:

Relation r'_1						
I_0	*Name*	I_1	*Address*	I_2	*DOB*	I_3
11000	John	11000	St. Paul	11000	1956	11010
00111	Peter	00111	St. Paul	00111	1975	01111
01011	Peter	01011	St. Paul	01111	1975	01011

The first and the third tuples in r_1 satisfy the selection condition. Note that the third tuple has generated two tuple in r'_1 with different source vectors since it satisfies two conjuncts of the selection condition.

3.2 Extended Projection ($\prod'$)

The projection operator is very similar to the classical operator. The instruction specifies the set of regular attributes for the projection, while the system actually carries out the projection on the specified attributes, together with their corresponding source attributes and the tuple source attribute. Let $r' = \prod'_X(r)$, where $X \subseteq \{A_1, \ldots, A_n\}$. Define $I_X = \{I_j \mid A_j \in X\}$ to be the set of source attributes associated with the regular attributes in X. The extended relational scheme of r' is $R' = \{I_0\} \cup X \cup I_X$. Then $r' = \{t' \mid t \in r,$ and t' is the restriction of t to $R'\}$.

3.3 Extended Union ($\cup'$)

Extended union is exactly the same as the classical case. Let r_1 and r_2 be extended relations on compatible schemes. Then $r_1 \cup' r_2 = \{t@U \mid t@U \in r_1$ or $t@U \in r_2\}$.

3.4 Extended Cartesian Product ($\times'$)

We will first define the product of two extended tuples. Let $\tau_1 = (u_0, (a_1@u_1), \ldots, (a_n@u_n))$ and $\tau_2 = (v_0, (b_1@v_1), \ldots, (b_n@v_m))$ be two extended tuples. Then their product, written $\tau_1 \times \tau_2$, is a tuple $\tau = (w_0, (c_1@w_1), \ldots, (c_n@w_{n+m}))$ where

$$\begin{array}{rcl} w_0 & = & u_0 \overset{s}{\wedge} v_0 \\ w_i & = & u_i \overset{s}{\wedge} w_0,\ i = 1, \ldots, n \\ w_i & = & v_{i-n} \overset{s}{\wedge} w_0,\ i = n+1, \ldots, n+m \\ c_i & = & a_i,\ i = 1, \ldots, n \\ c_i & = & b_{i-n},\ i = n+1, \ldots, n+m \end{array} \quad (1)$$

The computation of w_0 indicates that the product of two tuples exists ($e(w_0) = true$) if and only if both tuples taking part in the product exist ($e(u_0) = true$ and $e(v_0) = true$). The formulas for w_i enforce the well-formed condition (Section 2) for the product. Finally, the values of regular attributes of the product are simply the values from the two operands.

[2] We are investigating the extension of the method to allow queries to refer to sources and reliabilities as well. The extension should utilize a higher level of abstraction than source attribute values, since different coding methods can be applied to source vectors. This and other related issues are beyond the extent of this paper.

The Cartesian product of two extended relations can now be defined as the product of their tuples:

$$r_1 \times' r_2 = \{\tau_1 \times \tau_2 \mid \tau_1 \in r_1, \tau_2 \in r_2\}$$

3.5 Extended Join ($\bowtie'$)

We will first define the join of two extended tuples. Consider two extended relations r_1 and r_2 on the schemes R_1 and R_2, respectively. Let the set of common regular attributes of R_1 and R_2 be X. Also, let X_1 be the set of regular attributes in R_1 but not in R_2, and similarly, X_2 be the set of regular attributes in R_2 but not in R_1, Two tuples $t_1@U \in r_1$ and $t_2@V \in r_2$ can join if $t_1[X] = t_2[X]$. For an attribute $A_i \in X$, let $\mathcal{C}_i = u_i \overset{s}{\wedge} v_i$, where source vector u_i is associated with $t_1(A_i)$, and source vector v_i is associated with $t_2(A_i)$. The vector $\mathcal{C}_\rangle$ specifies the condition for both $t_1(A_i)$ and $t_2(A_i)$ to be valid. Define $\mathcal{C}(X) = \bigwedge_{A_i \in X} \mathcal{C}_i$. If $t_1@U \in r_1$ and $t_2@V \in r_2$ can join, then the resulting tuple $t@W = t_1@U \bowtie' t_2@V$ is obtained as follows.

$$\begin{aligned}
t[X] &= t_1[X] = t_2[X] \\
t[X_1] &= t_1[X_1] \\
t[X_2] &= t_2[X_2] \\
w_0 &= u_0 \overset{s}{\wedge} v_0 \overset{s}{\wedge} \mathcal{C}(X) \qquad (2)\\
w_i &= w_0, \text{ for source attributes } I_i,\ A_i \in X \\
w_i &= u_i \overset{s}{\wedge} w_0, \text{ for source attributes } I_i,\ A_i \in X_1 \\
w_i &= v_i \overset{s}{\wedge} w_0, \text{ for source attributes } I_i,\ A_i \in X_2
\end{aligned}$$

The extended join of two relations is defined as the extended join of their tuples:

$$r_1 \bowtie' r_2 = \{\tau_1 \bowtie' \tau_2 \mid \tau_1 \in r_1, \tau_2 \in r_2 \text{ and } \tau_1 \text{ and } \tau_2 \text{ can join}\}$$

The classical result that natural join can be implemented by projection, selection, and Cartesian product is also valide for extended relational algebra as the following theorem demonstrates.

Theorem 4 *Let X, X_1 and X_2 be as defined above. Then*

$$r_1 \bowtie' r_2 = {\prod_Y}' (\sigma'_F(r_1 \times' r_2))$$

where $Y = X_1 \cup X \cup X_2$ is the set of attributes for the resulting relations, and $F = \wedge_{A \in X}(r_1.A = r_2.A)$.

Proof: The argument for regular attribute values is the same as the regular relations. Here we will concenterate on the values for information source attributes, i.e. source vectors. A tuple $t@W$ is in $r_1 \bowtie' r_2$ if and only if there are tuples $t_1@U \in r_1$ and $t_2@V \in r_2$ which can join, and $t@W$ is obtained by Equation 2. But then $r_1 \times' r_2$ will contain a tuple $t'@W'$ obtained by Equation 1, namely, $w_0 = u_0 \overset{s}{\wedge} v_0$ and $w_i = u_i \overset{s}{\wedge} w_0$, if $A_i \in X_1 \cup X$ or $w_i = v_i \overset{s}{\wedge} w_0$, if $A_i \in X_2 \cup X$. The condition F of the extended selection consists of a single conjunct $F = \wedge_{A \in X}(r_1.A = r_2.A)$. Since t_1 and t_2 can join then t' satisfies the selection condition. Hence, by the extended selection rule, $\sigma'_F(r_1 \times' r_2)$ contains a tuple $t''@W''$ with $w''_0 = w'_0 \overset{s}{\wedge} \mathcal{C}$, and $w''_i = w'_i \overset{s}{\wedge} w''_0$, where $\mathcal{C} = \overset{s}{\wedge}_{A_i \in X}(u_i \overset{s}{\wedge} v_i) = \mathcal{C}(X)$. Finally, the projection operation simply "throws away" duplicated attribute values and source vectors of the attributes in X. The resulting tuple is $t@W$.

3.6 Extended Intersection ($\cap'$)

Let r_1 and r_2 be extended relations on compatible schemes, then $r = r_1 \cap' r_2$ is

$$\begin{aligned}
r = \{&(w_0, (a_1@w_1), \ldots, (a_n@w_n)) \mid \\
&(u_0, (a_1@u_1), \ldots, (a_n@u_n)) \in r_1, \text{ and}, \\
&(v_0, (a_1@v_1), \ldots, (a_n@v_n)) \in r_2\}
\end{aligned}$$

where $w_i = u_0 \overset{s}{\wedge} \cdots \overset{s}{\wedge} u_n \overset{s}{\wedge} v_0 \overset{s}{\wedge} \cdots \overset{s}{\wedge} v_n$, $i = 1, \ldots, n$. We will see in the next section that this definition is compatible with the semantics interpretation that considers a null value to be different from non-null values as well as other null values.

3.7 Extended Set Difference ($-'$)

The extended set difference is probably the most complicated operation in our extended relational algebra. Let r_1 and r_2 be two extended relations with compatible schemes, and $r = r_1 -' r_2$. The relation r can be obtained as the union of three sets of tuples as follows:

(1) If a pure tuple t appears in r_1 but does not appear at all in r_2, then r contains all tuples with pure component t. That is: $r \supset \{t@U \mid t@U \in r_1, \text{ and } t \text{ does not appear in } r_2\}$.

(2) If a pure tuple t appears in both r_1 and r_2, we need to consider all extended tuples with the pure component t in the computation of r. Let $t@U_1, \ldots, t@U_p \in r_1$ and Let $t@V_1, \ldots, t@V_q \in r_2$ be all tuples with pure component t in r_1 and r_2, respectively. Let $U_i = (u_{i_0}, \ldots, u_{i_n})$, $i = 1, \ldots, p$, and similarly $V_i = (v_{i_0}, \ldots, v_{i_n})$, $i = 1, \ldots, q$. Define $F_i = u_{i_0} \overset{s}{\wedge} \cdots \overset{s}{\wedge} u_{i_n}$, $i = 1, \ldots, p$, and similarly $G_i = v_{i_0} \overset{s}{\wedge} \cdots \overset{s}{\wedge} v_{i_n}$, $i = 1, \ldots, q$. The source vector formula

$$F = (\bigvee_{i=1}^{p} F_i) \bigwedge \neg(\bigvee_{i=1}^{q} G_i)$$

Defines a set of source vectors (recall that source vector disjunction was defined as the union of sets of source vectors). Let $x_1, \ldots, x_l \in F$. Then r also contains the tuples $t@W1, \ldots, t@Wl$ where $W_i = (w_{i_0}, \ldots, w_{i_n})$, where $w_{i_j} = x_i$, $j = 1, \ldots, n$.

The justification for including these tuples in r follows from the examination of the formula F. The first part of F specifies the condition under which the pure tuple t is a valid tuple in r_1. Similarly, the second part of F, without the negation, specifies the condition under which the pure tuple t is a valid tuple in r_2. Hence F specifies the condition under which t is a valid tuple in r_1, and is not a valid tuple in r_2, in which case it should be a valid tuple in $r_1 -' r_2$.

(3) Finally, there is one more set of tuples that should be included in r. If $t@U \in r_1$ and t appears in r_2, then the tuples $t@W$ should be included in r where $W = (w_0, \ldots, w_n)$, where $w_0 = u_0 \overset{s}{\wedge} u'_1 \overset{s}{\wedge} \cdots \overset{s}{\wedge} u'_n$, $u'_j = u_j$ or $u'_j = \overset{s}{\neg} u_j$, and at least one $u'_j = \overset{s}{\neg} u_j$, and $w_i = w_0 \overset{s}{\wedge} u_i$, $i = 1, \ldots, n$.

This set of tuples account for the case where a tuple t in r_1 has a null value for one or more attributes. In this case, t should be included in r since no tuple in r_2 can be equal to t. This is a consequence of the semantic interpretation that treats nulls as different from non-null as well as other nulls.

Example 2 Let r_1 and r_2 be the following relations

Relation r_1						
I_0	*Name*	I_1	*Address*	I_2	*DOB*	I_3
001	Sally	001	St. John	101	1954	011
100	Mary	100	St. John	101	1973	101
100	John	100	St. Paul	100	1956	110

Relation r_2						
I_0	*Name*	I_1	*Address*	I_2	*DOB*	I_3
010	John	010	St. Paul	011	1956	110
100	Mary	100	St. Paul	101	1973	101
001	Bob	001	St. John	011	1975	001
001	Peter	001	St. Paul	001	1975	101

The relation $r = r_1 -' r_2$ is shown below. The first two tuples of r_1 do not appear in r_2, so they are copied directly to r. (Note that Mary's Address is different in the two relations). The third tuple of r_1 appears in r_2, and hence the second and third rules for constructing tuples in r should be applied. In this case, each rule generates one tuple in r. This example also demonstrates how the -1 and $\top$ elements can be generated in the source vectors. The sv-negation operator used in the implementation of extended set difference can generate -1 entries. The $\top$ entries can then be generated as $-1 \overset{s}{\wedge} 1$ or $1 \overset{s}{\wedge} -1$. Such a situation indicates that the attribute value corresponding to the source vector (with a $\top$ entry) is not a valid value.

Relation $r = r_1 -' r_2$						
I_0	*Name*	I_1	*Address*	I_2	*DOB*	I_3
001	Sally	001	St. John	101	1954	011
100	Mary	100	St. John	101	1973	101
11-1	John	11-1	St. Paul	11-1	1956	11-1
1-10	John	1-10	St. Paul	1-10	1956	$1\top0$

4 Semantics and Correctness of Operations

In this section we will formalize the semantics of our model. We will extend the approach of [18] to prove the correctness of the extended relational algebra operations. An extended relation represents a set of regular relations. We can process a query on the extended relations by processing it against the (regular) relations represented by the extended relations. Of course this approach is not computationally feasible in general. What we need is a query processing methodology that is applied to extended relations directly, is efficient, and generates the same answers that would be obtained by processing the query against represented (regular) relations. We call a query processing methodology that satisfies the latter condition to be "precise." In what follows, we will formalize the concepts of representation and precision.

4.1 Representation

Given an extended relation r on the (extended) scheme R, $r = \{ t_1@\mathcal{U}_1, t_2@\mathcal{U}_2, \ldots, t_p@\mathcal{U}_p \}$; we define r^* as a function from the set of subsets of information sources $S = \{ s_1, s_2, ..., s_k \}$ to the set of (regular) relations Rel on the scheme $R - I$, where I is the set of source attributes, that is, $r^* : 2^S \rightarrow Rel$

Let $Q \subseteq S$ be a set of information sources. Assign truth values "$true$" to sources in Q, and "$false$" to other sources. (We will denote this assignment by the $truth(Q)$). Let $t = (a_1, a_2, \ldots, a_n)$ be a pure tuple such that $t@U \in r$ and $e(u_0) = true$ under truth(Q). Then $r^*(Q)$ will contain a (regular) tuple $t' = \mathcal{COR}(t, U, Q)$, where $t' = (b_1, b_2, \ldots, b_n)$ such that

$b_i = a_i$ if $e(u_i) = true$ under truth(Q)

$b_i =$ NULL if $e(u_i) = false$ under truth(Q).

and there exists at least one ($b_i \neq NULL$). Then,

$r^*(Q) = \{ t' \mid t' = \mathcal{COR}(t, U, Q), t@U \in r,$ and $e(u_0) = true$ under truth(Q). $\}$

Definition: The function r^* *represents* the extended relation r, written as $rep(r) = r^*$. The set of (regular) relations $r^*(Q), (\forall Q)(Q \subseteq S)$, is called the *alternate world* of the extended relation r.

Example 3 Consider the extended relation r_1 of Example 2 (Section 3.7) that represents the information gathered by sources s_1, s_2, and s_3.

We would like to find the alternate world representation of r_1. We shall consider all subsets (Q) of $\{ s_1, s_2, s_3 \}$ and find their corresponding regular relations. Given $2^S = \{ \{\}, \{ s_1\}, \{s_2\}, \{s_3\}, \{s_1, s_2\}, \{s_1, s_3\}, \{s_2, s_3\}, \{s_1, s_2, s_3\} \}$, then the alternate world representation of r_1 consists of the empty regular relation (corresponding to the empty set of sources as well as $\{s_2\}$) and the following regular relations:

$r_1^*(\{s_1\})$		
Name	*Address*	*DOB*
Mary	NULL	NULL
John	St. Paul	NULL

$r_1^*(\{s_3\})$		
Name	*Address*	*DOB*
Sally	NULL	NULL

$r_1^*(\{s_1, s_2\})$		
Name	*Address*	*DOB*
Mary	NULL	NULL
John	St. Paul	1956

$r_1^*(\{s_1, s_3\})$		
Name	*Address*	*DOB*
Sally	St. John	NULL
Mary	St. John	1973
John	St. Paul	NULL

$r_1^*(\{s_2, s_3\})$		
Name	*Address*	*DOB*
Sally	NULL	1954

$r_1^*(\{s_1, s_2, s_3\})$		
Name	*Address*	*DOB*
Sally	St. John	1954
Mary	St. John	1973
John	St. Paul	1956

4.2 Precision

Definition: Let r be an extended relation, and Θ be the unary relational algebra operator. An extended operator Θ' is *precise* if

$$rep(\Theta'(r)) = \Theta(rep(r))$$

for all extended relations r, where $rep(r) = r^*$ is the function representing r, and $\Theta(rep(r))$ represents a function f such that $f(Q) = \Theta(r^*(Q)) \; \forall Q \subseteq S$.

Definition: Let r_1 and r_2 be extended relations, and let Θ be a binary relational algebra operator. An extended operator Θ' is precise if

$$rep(r_1 \Theta' r_2) = rep(r_1) \Theta rep(r_2)$$

for all extended relations r_1 and r_2, where $rep(r_1)\Theta rep(r_2)$ represents a function f such that $f(Q) = {r_1}^*(Q)\Theta {r_2}^*(Q)$ $\forall Q \subseteq S$.

4.3 Extended Relational Algebra Operations are Precise.

In this section we discuss the "correctness" of the extended relational algebra operations of Section 3. We will show that these operations are *precise* with respect to the alternate worlds semantic model presented in this section. This provides a strong justification for the correctness of the operations. Because of space limitations we will not provide all the proofs in this paper. Interested readers are referred to the full paper for complete proofs and additional discussions and examples.

Theorem 5 *The extended relational algebra operations selection, projection, Cartesian product, join, intersection, and set difference are precise.*

Proof: We will only give the proof for extended selection, and argue the case for extended set difference using an example. Complete proofs are presented in the full paper.

Extended selection: Let r be an extended relation whose scheme is defined as $R = \{I_0, (A_1, I_1), \ldots, (A_n, I_n)\}$. Let $r' = \sigma'_F(r)$, where F is in disjunctive form $F = F_1 \vee \ldots \vee F_q$ where each F_i is a conjunction of simple conditions. Let S be the set of information sources. We will show that for any $Q \subseteq S$

$$r'^*(Q) = \sigma_F r^*(Q)$$

and hence extended selection is precise.

(1) If $t' \in r'^*(Q)$, then there exist t and U such that $t@U \in r'$, $U = (u_0, \ldots, u_n)$, $e(u_0) = true$ under truth(Q), and $t' = \mathcal{COR}(t, U, Q)$. Let $\mathcal{A} = \{A \mid t'(A) \neq NULL\}$ and $\mathcal{B} = \{A \mid t'(A) = NULL\}$. Now, since $r' = \sigma'_F(r)$, then there should exist t and $\mathcal{V}$, where $t@\mathcal{V} \in r_1$ and t satisfies F. Since t satisfies F, then t satisfies at least one F_i. Let $F_{t_1}, \ldots, F_{t_p}$ be the conjuncts satisfied by t. It follows that there should exist $V = (v_0, v_1, \ldots, v_n)$ such that $t@V \in r_1$, pure tuple t satisfy F_{t_j} for some $j \in \{1, \ldots, p\}$, and

- $u_0 = v_0 \overset{s}{\wedge} \mathcal{C}_j$ and $e(u_0) = true$ under truth(Q). It follows by theorem 2 that $e(v_0) = true$ under truth(Q).
- $\forall A_i \in \mathcal{A}$, $\exists u_i$ associated with $t(A_i)$ in r' such that $u_i = u_0 \overset{s}{\wedge} v_i$ where $e(u_i) = true$ under truth(Q). It follows by theorem 2 that $e(v_i) = true$ under truth(Q).
- $\forall A_j \in \mathcal{B}$, $\exists u_j$ associated with $t(A_j)$ in r' such that $u_j = u_0 \overset{s}{\wedge} v_j$ where $e(u_j) = false$ under truth(Q) and $e(u_0) = true$. It follows by theorem 2 that $e(v_j) = false$ under truth(Q).

Now, pure tuple t satisfies the selection condition F_{t_j} and $e(v_0) = true$ under truth(Q). Moreover, since $\forall A_i \in \mathcal{A}$, $e(v_i) = true$ under truth(Q) and $\forall A_j \in \mathcal{B}$, $e(v_j) = false$ under truth(Q), we deduce that $t' \in r^*(Q)$, where $t' = \mathcal{COR}(t, V, Q)$ and t' satisfies F_{t_j}. Therefore, $rep(\sigma'_F(r)) \subseteq \sigma_F(rep(r))$.

(2) If $t' \in r^*(Q)$ and t' satisfies F, then there exist t and V such that $t@V \in r$, $V = (v_0, \ldots, v_n)$, $e(v_0) = true$ under truth(Q), and $t' = \mathcal{COR}(t, V, Q)$. Let $\mathcal{A} = \{A \mid t'(A) \neq NULL\}$ and $\mathcal{B} = \{A \mid t'(A) = NULL\}$. Now, since $r' = \sigma'_F(r)$, where $t@V \in r$, and since t satisfies $F = F_1 \vee \ldots \vee F_q$, then t satisfies at least one F_j. It follows that there exist some $U \in \mathcal{U}$ such that $t@U \in r'$, where $U = (u_0, \ldots, u_n)$, and

- $u_0 = v_0 \overset{s}{\wedge} \mathcal{C}_j$ and $e(v_0) = true$ under truth(Q), where v_0 is the source vector associated with pure tuple t in r. Moreover, $e(\mathcal{C}_j) = true$ under truth(Q) since t' satisfies F. It follows by theorem 2 that $e(u_0) = true$ under truth(Q).
- $\forall A_i \in \mathcal{A}$, $\exists v_i$ associated with $t(A_i)$ in r such that $e(v_i) = true$ under truth(Q), it follows that, since $u_i = u_0 \overset{s}{\wedge} v_i$ and $e(u_0) = true$ under truth(Q), then $e(u_i) = true$ under truth(Q).
- $\forall A_j \in \mathcal{B}$, $\exists v_j$ associated with $t(A_j)$ in r such that $e(v_j) = false$ under truth(Q), it follows that, since $u_j = u_0 \overset{s}{\wedge} v_j$, then $e(u_j) = false$ under truth(Q).

Now, pure tuple t satisfies the selection condition F and $e(u_0) = true$ under truth(Q); moreover, since $\forall A_i \in \mathcal{A}$, $e(u_i) = true$ under truth(Q) and $\forall A_j \in \mathcal{B}$, $e(u_j) = false$ under truth(Q), we deduce that $t' \in r'^*(Q)$, where $t' = \mathcal{COR}(t, U, Q)$. Therefore, $\sigma_F(rep(r)) \subseteq rep(\sigma'_F(r))$.

Precision of Extended Set Difference

The proof of precision for extended set difference is involved and is given in the full paper. Here we will demonstrate its precision for the Example 2 of Section 3.7. The alternate world relations of r_1 was obtained in Example 3 of Section 4.1. Here we show the alternate world relations of r_2 and $r = r_1 -' r_2$. It can easily be verified that $r^*(Q) = r_1^*(Q) - r_2^*(Q)$, for all $Q \subseteq S$. We should note here that the semantics used for NULL values is that a NULL is different from all non-null values, as well as all other NULLs. This might seem counter intuitive. For example, for $Q = \{s_1\}$, both $r_1^*(Q)$ and $r_2^*(Q)$ contain the tuple (Mary, NULL, NULL), and $r^*(Q) = r_1^*(Q) - r_2^*(Q)$ also contains this tuple. The reason is that the two tuples are considered to be *different* because NULLs are different. It is possible to define an extended set difference operation with a different semantics for NULLs. These issues have been discussed in the full paper.

$r_2^*(\{s_1\})$

Name	*Address*	*DOB*
Mary	NULL	NULL

$r^*(\{s_1\})$

Name	*Address*	*DOB*
Mary	NULL	NULL
John	St. Paul	NULL

$r_2^*(\{s_3\})$

Name	*Address*	*DOB*
Bob	NULL	1975
Peter	St. Paul	NULL

$r^*(\{s_3\})$

Name	*Address*	*DOB*
Sally	NULL	NULL

$r_2^*(\{s_1, s_2\})$

Name	*Address*	*DOB*
John	NULL	1956
Mary	NULL	NULL

$r^*(\{s_1, s_2\})$

Name	*Address*	*DOB*
Mary	NULL	NULL
John	St. Paul	1956

$r_2^*(\{s_1, s_3\})$

Name	*Address*	*DOB*
Mary	St. Paul	1973
Bob	NULL	1975
Peter	St. Paul	1975

$r^*(\{s_1, s_3\})$

Name	*Address*	*DOB*
Sally	St. John	NULL
Mary	St. John	1973
John	St. Paul	NULL

$r_2^*(\{s_2, s_3\})$

Name	*Address*	*DOB*
John	St. Paul	NULL
Bob	St. John	1975
Peter	St. Paul	NULL

$r^*(\{s_2, s_3\})$

Name	*Address*	*DOB*
Sally	NULL	1954

$r_2^*(\{s_1, s_2, s_3\})$

Name	*Address*	*DOB*
John	St. Paul	1956
Mary	St. Paul	1973
Bob	St. John	1975
Peter	St. Paul	1975

$r^*(\{s_1, s_2, s_3\})$

Name	*Address*	*DOB*
Sally	St. John	1954
Mary	St. John	1973

5 Conclusions and Future Work

An extended relational model was presented that enables and facilitates the modeling and manipulation of uncertain and inaccurate data at the level of attribute values. This is an extension of the Information Source Tracking (IST) method, which only supports the modeling and management of uncertainty at the level of tuples in relational databases. We presented extended relational algebra operations, and proved their correctness with respect to a semantic model for extended relations.

There are several interesting directions that require further investigation. We would like to provide facilities to refer to sources and reliabilities in the queries. For example we might be interested only in the answers that are not dependent on particular information sources. These types of queries are currently possible by producing all answers and then selecting the desired ones, but a direct approach can result in more efficient query processing. A suitable syntax and language for this type of queries is also an interesting research topic. The appropriate functions for combining numeric measures of uncertainty is also an important research direction. Various combination rules have been used, and many others can be envisioned. No rule seems appropriate for all cases, and a characterization of rules and their domain of applicability is highly desirable.

References

[1] D. Barbara, H. Garcia-Molina, and D. Porter, "The Management of Probabilistic Data." IEEE Transactions on Knowledge and Data Engineering, Vol. 4, No. 5, October 1992, pp 487-502.

[2] N. D., Jr. Belnap, "A Useful Four-Valued Logic." In *Modern Uses of Multiple-Valued Logic*, J. Michael Dunn and G. Epstein, eds., Reidel, 1977, pp 8-37.

[3] U. Güntzer, W. Kießling, and H. Thöne, "New Directions for Uncertainty Reasoning in Deductive Databases." Proceedings of the 1991 ACM SIGMOD International Conference on Management of Data, pp 178-187.

[4] H. Jamil, and F. Sadri, "Trusting an Information Agent." In *Rough Sets, Fuzzy Sets, and Knowledge Discovery*, W. P. Ziarko, ed. Springer-Verlag Workshops in Computing Series, 1994. (Proceedings of the workshop on Rough Sets and Knowledge Discovery, RSKD'93, Banff, Canada, October 1993.)

[5] H. M. Jamil, and F. Sadri, "Recognizing Credible Experts in Inaccurate Databases." Proceedings of the 1994 International Symposium on Methodologies for intelligent Systems, ISMIS'94, Charlotte, NC, October 1994, pp 46-55. LNAI No. 869, Springer-Verlag.

[6] C. S. Ku, H. D. Kim, and L. J. Henschen, "An Efficient Indefiniteness Inference Scheme in Indefinite Deductive Databases." IEEE Transactions on Knowledge and Data Engineering, Vol. 6, No. 5, October 1994, pp 713-722.

[7] V. S. Lakshmanan, and F. Sadri, "Modeling Uncertainty in Deductive Databases." International Conference on Database and Expert Systems Applications, DEXA'94, Athens, Greece, September 1994, pp 724-733. LNCS No. 856, Springer-Verlag.

[8] V. S. Lakshmanan, and F. Sadri, "Probabilistic Deductive Databases." Proceedings of the 1994 International Logic Programming Symposium, ILPS'94, Ithaca, NY, November 1994, pp 254-268.

[9] Suk Kyoon Lee, "Imprecise and Uncertain Information in Databases: An Evidential Approach." Proceedings of the 1992 IEEE International Conference on Data Engineering, pp 614-621.

[10] S. K. Lee, "An Extended Relational Database Model for Uncertain and Imprecise Information." Proceedings of the 1992 International Conference on Very Large Databases, pp 211-220.

[11] K-C Liu, and R. Sunderraaman, "Indefinite and Maybe Information in Relational Data Bases.", *ACM Transactions on Database Systems*, Vol. 15, No. 1, March 1990, pp 1-39.

[12] K-C. Liu, and R. Sunderraaman, "A Generalized Model for Indefinite and Maybe Information.", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 3, No. 1, March 1991, pp 65-77.

[13] R. T. Ng, "Semantics and Consistency of Empirical Databases." Proceedings of 10th International Conference on Logic Programming,

[14] R. Ng, and V. S. Subrahmanian, "Probabilistic Logic Programming." Information and Computation, 1992, pp 150-201.

[15] A. Ola, and G. Ozsoyoglu, "Incomplete Relational Database Models Based on Intervals." IEEE Transactions on Knowledge and Data Engineering, Vol. 5, No. 2, April 1993, pp 293-308.

[16] M. Pittarelli, "An Algebra for Probabilistic Databases." IEEE Transactions on Knowledge and Data Engineering, Vol. 6, No. 2, April 1994, pp 293-303.

[17] F. Sadri "Reliability of Answers to queries in Relational Database.", *IEEE Transaction on Knowledge and data Engineering*, Vol. 3, No. 2, June 91, pp 245-251.

[18] F. Sadri "Modeling Uncertainty in Databases.", *Proceeding of the 1991 IEEE International Conference on Data Engineering*, pp 122-131.

[19] F. Sadri, "Aggregate Operations in Information Source Tracking Method." Theoretical Computer Science 133 (1994) pp 421-442.

[20] F. Sadri, "Integrity Constraints in the Information Source Tracking Method." IEEE Transactions on Knowledge and Data Engineering, Vol. 7, No. 1, February 1995, pp 106-119.

[21] F. Sadri, "Information Source Tracking Method: Efficiency Issues." To appear in IEEE Transactions on Knowledge and Data Engineering.