Turk J Elec Engin, VOL.7, NO.1-3 1999, © TÜBİTAK

Exclusive Disjunctions in Indefinite and Maybe Information in Relational Databases*

Taflan İ. Gündem
Computer Engineering Dept., Boğaziçi University,
80815, Bebek, Istanbul-TURKEY

Abstract

Incorporating indefinite information into databases has been studied extensively. In this paper, we propose a structure called an E-table to represent maybe information and inclusive and exclusive disjunctions. We define the type of redundancies in E-tables and show how to eliminate them. Also we present an extended relational algebra to operate on E-tables. In this paper we expand the concepts and operations defined by Lin and Sunderraman [1] in order to accommodate exclusive disjunctions in relational databases.

1. Introduction

Although the relational model [2] did not originally allow incomplete information, the real world modeling requirements stimulated a considerable amount of research on extending the relational model to include incomplete information. Various kinds of incomplete information have been considered [3-6]. There has been a thorough treatment of indefinite and maybe kinds of incomplete information in relational databases by Liu and Sunderraman [1].

A data structure called an I-table is defined [1]. An I-table has three components: definite, indefinite and maybe. For example the I-table in Figure 1 represents the following facts:

a)

TEACHES

Teacher	Course	
John	CS 101	definite part
Mac	CS 102	indefinite part
Mac	CS 104	
Robin	CS 500	maybe part

b)

TEACHES

Teacher	Course	
John	CS 101	definite part
Mac	CS 102	inclusive part
Mac	CS 104	
Lisa	CS 200	exclusive part
Lisa	CS 400	
Robin	CS 500	maybe part

Figure 1.a) An I-table, **b)** An E-table.

(1) John teaches CS 101.

(2) Mac teaches CS 102 or CS 104.

(3) Robin maybe teaches CS 500.

*This study was supported by Boğaziçi University Research Fund under grant number 92A0117.

The indefinite part of an I-table consists of a set, S, of set of tuples (i.e. S=$\{w_1, w_2, \ldots w_j, \ldots w_k\}$, where $w_i = \{t_1, t_2, \ldots t_i, \ldots t_j\}$ and where t_i is a tuple). Each set w_i can be viewed as a statement $R(t_1)$ or $R(t_2) \ldots$ or $R(t_j)$. In Figure 1. a), the I-table contains in its idenfinite part just one set with two tuples in it.

In this paper we extend the I-table so that it includes exclusive disjunctions as well as inclusive disjunctions. In other words, the indefinite part of a table is divided into two parts: inclusive and exclusive. We call such an extended I-table an E-table and explain it formally in the next section. The exclusive part of an E- table is also a set of tuples. However here a set $w_v = \{t_1, t_2, \ldots t_z\}$ is viewed as a statement R(t_1) xor R(t_2) ... xor R(t_z). In Figure 1.b), in the E-table we represent not only all the information in the I-table in Figure 1.a) but also the information that Lisa teaches either CS 200 or CS 400 but not both. This example may suggest exclusive disjunctions are needed for representing an important class of incomplete information and are often present in real life applications.

In the next section, we redefine the relational algebra operations for the E-table and also define an operation called ELIMRID to reduce the redundancies in an E-table.

2. E-tables, Redundancies and Relational Algebra Operations

The E-table is an extension of the table representing a relation in the relational model. The table allows the existence of disjunctive facts as proposed by the definition of I-tables [1]. The difference between an I-table and an E-table is that the latter is capable of representing not only inclusive disjunctions, but exclusive disjunctions as well.

An E-table scheme is an ordered list of attribute names R=$< A_1, \ldots, A_n >$. Associated with each attribute name, A_i, is a domain D_i. Then, $T =< T_D, T_I, T_E, T_M >$ is an E-table over the scheme R where $T_D \subseteq D_1 X \ldots X D_n$

$$T_I \subseteq 2^{D1X \ldots XDn} - \{\{\theta\} U \{\{t\} | t \varepsilon D_1 X \ldots X D_n\}\}$$

$$T_E \subseteq 2^{D1X \ldots XDn} - \{\{\theta\} U \{\{t\} | t \varepsilon D_1 X \ldots X D_n\}\}$$

$$T_M \subseteq 2^{D1X \ldots XDn} - \{\theta\}$$

T_D is the set of tuples named definite tuples. T_I is the set of sets of tuples called inclusive indefinite tuple sets. T_E is the set of sets of tuples called exclusive indefinite sets. T_M is the set of sets of tuples called maybe tuple sets. The symbols T, $T_1, \ldots$ are used for E-tables, $t, t_1, \ldots$, for tuples, $w_1, w_2, \ldots, q_1, q_2, \ldots$ for tuple sets, $r, r_1, \ldots$ for relations. We use 'or' for the logical or operator and 'xor' for the logical exclusive or operator.

Suppose that we have an E-table for storing information about a predicate P. The way to interpret the content of an E-table can be explained as follows:

i) tϵT_D is interpreted as P(t).

ii) wϵT_I and w=$\{t_1, \ldots, t_n\}$ is interpreted as $P(t_1)$ or ... or $P(t_n)$

iii) wϵT_E and w=$\{t_1, \ldots, t_n\}$ is interpreted as $P(t_1)$ xor... xor P(t_n)

iv) wϵT_M and w={t} is interpreted as P(t) xor $\neg$P(t) wϵT_M and w=$\{t_1, \ldots, t_n\}$ is interpreted as $(P(t_1)$ xor $\ldots$ xor $P(t_n))$ xor $(\neg P(t_1)$ and $\ldots$ and $\neg P(t_n))$

In Lipski's [7] words, T_M provides us "the set of objects for which we cannot rule out the possibility that they belong to an external interpretation of a query". The need to store the elements of T_M arises from i) the user's desire to store such information, ii) the system's generation of such tuples as a result of the application of the algebra operators on T_I, T_E and T_M.

Introduction of redundant information may be incurred through the application of the extended relational algebra operators to E-tables or after an insert/modify operation. Possible sources of redundancy in E-tables and the way to eliminate these redundancies can be summarized as follows:

1) tϵT_D, wϵT_I and t ϵw. Remedy: Delete w from T_I. Insert in T_m each of the tuples in w-{t}.

2) w_1 $\epsilon T_I, w_2 \epsilon T_I$ and $w_2 \supset w_1$.

Remedy: Delete w_2 from T_I. Insert in T_m each of the tuples in $w_2 - w_1$.

3) w ϵT_M, tϵT_D and tϵw. Remedy: Delete w from T_m.

4) w_1 ϵT_M, $|w_1|$=1 and $w_2 \epsilon T_I$ and $w_2 \supset w_1$. Remedy: Delete t from T_m.

5) t ϵT_D, w ϵT_E and tϵw. Remedy: Delete w from T_E.

6) w_1 ϵT_E, $w_2 \epsilon T_E$ and $w_2 \supset w_1$. Remedy: Delete w_2 from T_E.

7) w_1 ϵT_E, w_2 ϵT_M and $w_1 \supset w_2$. Remedy: Delete w_2 from T_m.

8) w_1 ϵT_E, w_2 ϵT_M and $w_2 \supset w_1$. Remedy: Delete w_1 from T_E.

9) w_1 ϵT_E, w_2 ϵT_I and $w_2 \supset w_1$.

Remedy: Delete w_2 from T_I. Insert each element of $w_2 - w_1$ to T_M.

10) w_1 ϵT_E, w_2 ϵT_I and $w_1 \supset w_2$.

Remedy: Delete w_1 from T_I. Delete w_2 from T_E. Insert w_1 to T_E.

In the light of the different types of redundancies listed above, we can define an operator, ELIMRED, that takes in as input an E-table and returns the E-table with all the redundancies eliminated.

Definition 2.1 ELIMRED : $\Psi_R -> \Psi_R$ is a mapping such that ELIMRED (T) $=T^0$, where T^0 is defined as follows:

$$
\begin{aligned}
T_D^0 &= \{\text{t} \mid \text{t } \epsilon T_D\} \\
T_I^0 &= \{\text{w} \mid \text{w } \epsilon T_I \wedge \neg(\exists t)(t \epsilon T_D \wedge t \epsilon w) \wedge \neg(\exists w_1)(w_1 \epsilon T_I \wedge w \supset w_1) \\
&\quad \wedge \neg(\exists w_2)(w_2 \epsilon T_E \wedge w \supset w_2) \wedge \neg(\exists w_3)(w_3 \epsilon T_E \wedge w_3 \supset w)\} \\
T_E^0 &= \{\text{w} \mid \text{w } \epsilon S_1 \wedge \neg(\exists t_1)(t_1 \epsilon T_D \wedge t_1 \epsilon w) \wedge \neg(\exists w_1)(w_1 \epsilon T_E \wedge w \supset w_1) \\
&\quad \wedge \neg(\exists w_2)(w_2 \epsilon T_M \wedge w_2 \supset w) \wedge \neg(\exists w_3)(w_3 \epsilon T_I \wedge w \supset w_3)\} \\
&\quad \text{where } S_1 = \{\text{w} \mid \text{w } \epsilon T_E \text{V} (\exists w_1)(\exists w_2)(w_1 \epsilon T_E \wedge w_2 \epsilon T_I \wedge w_1 \supset w_2 \wedge w = w_1)\} \\
T_M^0 &= \{\text{w} \mid \text{w } \epsilon S_2 \wedge (\forall t)(t \epsilon w -> \neg(t \epsilon T_D^o)) \wedge \neg(\exists w_1)(w_1 \epsilon T_I^0 \wedge \mid w \mid = 1 \wedge w_1 \supset w) \\
&\quad \wedge \neg(\exists w_2)(w_2 \epsilon T_E^0 \wedge w \supset w_2)\} \\
&\quad \text{where } S_2 = \{\text{w} \mid \text{w } \epsilon T_M V(\exists w_1)(\exists w_2)(w_1 \epsilon T_I \wedge w_2 \epsilon T_I \wedge w_2 \supset w_1 \wedge (\forall t)(t \epsilon w_2 - w_1 -> \\
&\quad \text{w=\{t\}) V } (\exists w_3)(\exists w_4)(w_3 \epsilon T_E \wedge w_4 \epsilon T_I \wedge w_4 \supset w_3 \wedge (\forall t)(t \epsilon w_4 - w_3 -> w = \{\text{t}\}) \\
&\quad V(\exists t_1)(\exists w_5)(t_1 \epsilon T_D \wedge w_5 \epsilon T_I \wedge t_1 \epsilon w_5 \wedge (\forall t)(t \epsilon w_5 - \{t_1\} -> w = \{t\}))\}
\end{aligned}
$$

Now, we define our extended algebra operators. The definitions of T_D and T_I are the same as those in [1] for the union, selection, cartesian product, projection and intersection operators. Note that Ψ_R:{T | T: E-Table over R}

Definition 2.2 Union on Ψ_R is a mapping U: $\Psi_R \times \Psi_R -> \Psi_R$.

Let T_1 and T_2 be two domain compatible E-tables. Then, $T_1 U T_2$= ELIMRED(T), where T is defined as follows:

$$\begin{array}{ll} T_D & =\{\text{t}|\text{t } \epsilon T_D^1 V t\epsilon T_D^2\} \\ T_I & =\{\text{w}|\text{w } \epsilon T_I^1 V w\epsilon T_I^2\} \\ T_E & =\{\text{w}|\text{w } \epsilon T_E^1 V w\epsilon T_E^2\} \\ T_M & =\{\text{w}|\text{w } \epsilon T_M^1 V w\epsilon T_M^2\} \end{array}$$

An example of the onion operation is given in Figure 2.

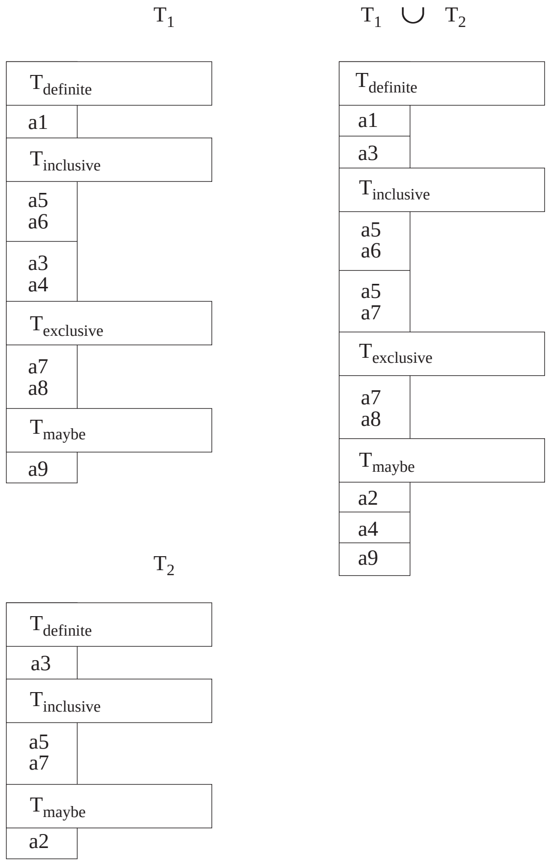

Figure 2. An example of the extended union operation.

Definition 2.3 Selection on elements of Ψ_R is a mapping $\sigma : \Psi_R - > \Psi_R$.

Let T_1 be an E-table and F be a formula involving operands that are constants or attribute numbers and arithmetic comparison operators: $< . =, >$, and logical operators V, $\wedge, \neg$. Then, $\sigma_F(T_1)$=ELIMRED(T), where T is defined as follows:

$$\begin{array}{lll} T_D & = & \{\text{t} \mid \text{t } \epsilon T_D^1 \text{ V F(t)}\} \\ T_I & = & \{\text{w} \mid \text{w } \epsilon T_I^1 \wedge (\forall t)(t\epsilon w- > \text{F(t)})\} \\ T_E & = & \{\text{w} \mid \text{w } \epsilon T_E^1 \wedge (\forall t)(t\epsilon w- > F(t))\} \\ T_M & = & \{\text{w} \mid (\text{w } \epsilon T_M^1 \wedge (\forall t)(t\epsilon w- > \text{F(t)})) \\ & & \text{V } (\exists w_1)(w_1 \epsilon T_I^1 \wedge t\epsilon w_1 \wedge F(t) \wedge w = t) \\ & & \text{V } (\exists w_2)(w_2 \epsilon T_E^1 \wedge w_2 \supset w \wedge (t\epsilon w- > F(t)))\} \end{array}$$

An example of the selection operation is given in Figure 3.

T

$T_{definite}$	
a3	b3
a2	b2
$T_{inclusive}$	
a3	b1
a3	b2
a4	b2
a3	b4
$T_{exclusive}$	
a4	b1
a4	b4
a3	b5
a4	b6
a3	b8
T_{maybe}	
a3	b7
a3	b6

$\sigma_{1=\text{"a3"}}(T)$

$T_{definite}$	
a3	b3
$T_{inclusive}$	
a3	b1
a3	b2
T_{maybe}	
a3	b4
a3	b5
a3	b8
a3	b7
a3	b6

Figure 3. An example of the extended selection operation.

Definition 2.4 Difference on Ψ_R is a mapping -: $\Psi_R \times \Psi_R -> \Psi_R$.

Let T_1 and T_2 be two domain-compatible E-tables. Then, T_1-T_2=ELIMRED(T), where T is defined as follows:

$$
\begin{array}{lll}
T_D & = & \{t|(t\epsilon T_D^1) \wedge \neg(t\epsilon T_D^2) \wedge \neg(\exists w)((w\epsilon T_I^2 \wedge t\epsilon w)V(w\epsilon T_E^2 \wedge t\epsilon w)V(w\epsilon T_M^2 \wedge t\epsilon w))\} \\
T_I & = & \{w|(w\epsilon T_I^1) \wedge \neg(\exists t)(t\epsilon T_D^2 \wedge t\epsilon w \wedge \neg(\exists w_1)((w_1\epsilon T_I^2 \wedge \neg(w \cap w_1 = \phi)) \\
 & & V(w_1\epsilon T_E^2 \wedge \neg(w \cap w_1 = \phi))V(w_1\epsilon T_M^2 \wedge \neg(w \cap w_1 = \phi))\} \\
T_E & = & \{w \mid (w\epsilon T_E^1) \wedge \neg(\exists t)(t\epsilon T_D^2 \wedge t\epsilon w) \\
 & & \wedge\neg(\exists w_1)((w_1\epsilon T_I^2 \wedge \neg(w \cap w_1 = \phi)) \\
 & & V(w_1\epsilon T_E^2 \wedge \neg(w \cap w_1 = \phi)) \\
 & & V(w_1\epsilon T_M^2 \wedge \neg(w \cap w_1 = \phi))\} \\
T_M & = & \{w \mid (w\epsilon T_M^1) \\
 & & \text{V } (\exists w_1)(\exists t)(t\epsilon T_D^1 \wedge w_1\epsilon T_I^2 \wedge t\epsilon w_1 \wedge w = \{t\}) \\
 & & \text{V } (\exists w_1)(\exists t)(t\epsilon T_D^1 \wedge w_1\epsilon T_E^2 \wedge t\epsilon w_1 \wedge w = \{t\}) \\
 & & \text{V } (\exists w_1)(\exists t)(t\epsilon T_D^1 \wedge w_1\epsilon T_M^2 \wedge t\epsilon w_1 \wedge w = \{t\}) \\
 & & \text{V } (\exists w_1)(\exists t_1)(w_1\epsilon T_I^1 \wedge t_1\epsilon T_D^2 \wedge t_1\epsilon w_1 \wedge (\forall t_2)(t_2\epsilon w_1 - \{t_1\} -> w = \{t_2\})) \\
 & & \text{V } (\exists w_1)(\exists w_2)(w_1\epsilon T_I^1 \wedge w_2\epsilon T_I^2 \wedge \neg(w_1 \cap w_2 = \phi) \wedge (\forall t_1)(t_1\epsilon w_1 -> w = \{t_1\})) \\
 & & \text{V } (\exists w_1)(\exists w_2)(w_1\epsilon T_I^1 \wedge w_2\epsilon T_M^2 \wedge \neg(w_1 \cap w_2 = \phi) \wedge (\forall t_1)(t_1\epsilon w_1 -> w = \{t_1\})) \\
 & & \text{V } (\exists w_1)(\exists w_2)(w_1\epsilon T_I^1 \wedge w_2\epsilon T_E^2 \wedge \neg(w_1 \cap w_2 = \phi) \wedge (\forall t_1)(t_1\epsilon w_1 -> w = \{t_1\})) \\
 & & \text{V } (\exists w_1)(\exists w_2)(w_1\epsilon T_E^1 \wedge w_2\epsilon T_I^2 \wedge \neg(w_1 \cap w_2 = \phi) \wedge (\forall t_1)(t\epsilon w_1 -> w = \{t_1\}))\}
\end{array}
$$

An example of the difference operation is given in Figure 4.

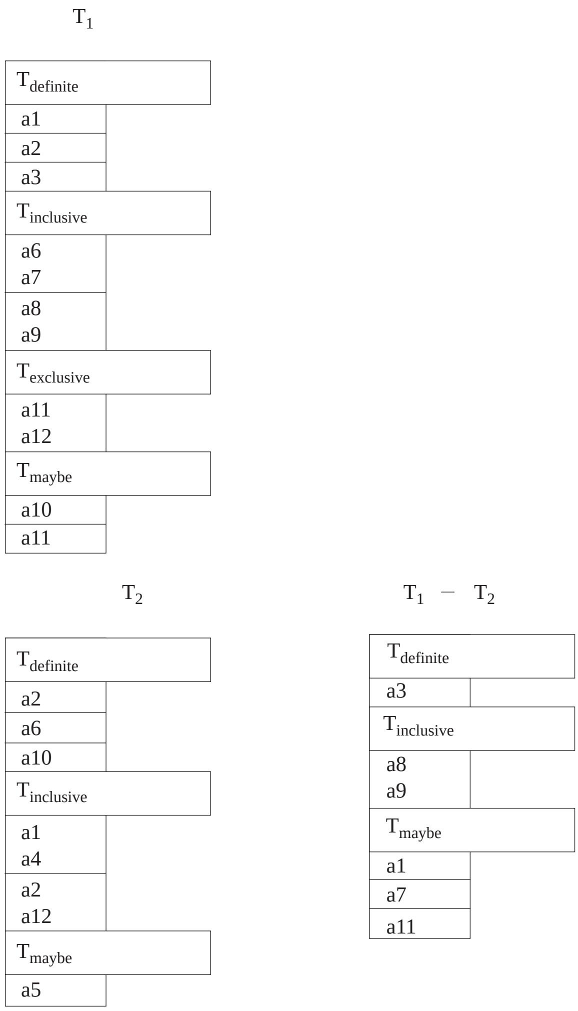

Figure 4. An example of the extended difference operation.

Definition 2.5 Projection on elements of Ψ_R is a mapping $\prod : \Psi_R - > \Psi_R$

Let T_1 be an E-table and A be a list of attribute numbers. Then, $\prod_A(T_1)$=ELIMRED(T), where T is defined as follows:

$$\begin{aligned} T_D &= \{t \mid (\exists t_1)(t_1 \epsilon T_D^1 \wedge t[A] = t_1[A]) V (\exists w_1)(w_1 \epsilon T_I^1 \wedge (\forall t_1)(t_1 \epsilon w_1 - > t[A] = t_1[A])) \\ &\quad V(\exists w_2)(w_2 \epsilon T_I^1 \wedge (\forall t_2)(t_2 \epsilon w_2 - > t[A] = t_2[A]))\} \\ T_I &= \{w \mid (\exists w_1)(w_1 \epsilon T_I^1 \wedge w = \textstyle\prod_A(w_1) \wedge \mid w \mid > 1\} \\ T_E &= \{w \mid (\exists w_1)(w_1 \epsilon T_E^1 \wedge w = \textstyle\prod_A(w_2) \wedge \mid w \mid > 1\} \\ T_M &= \{w \mid (\exists w_1)(w_1 \epsilon T_M^1 \wedge w = \textstyle\prod_A(w_1)\} \end{aligned}$$

An example of the difference operation is given in Figure 5.

Definition 2.6 Cartesian product of elements of Ψ_{R1} and Ψ_{R2} is a mapping x : $\Psi_{R1} \times \Psi_{R2} - > \Psi_{R1.R2}$

Let T_1 and T_2 be I-tables such that $T_I^1 = \{w_1^1, \ldots w_m^1\}$ and $T_I^2 = \{w_1^2, \ldots w_n^2\}$.

Let $E = \{\{t_1, \ldots, t_m\} \mid (\forall i)(1 \le i \le m - > t_i \epsilon w_i^1)\}$ and F=$\{\{t_1, \ldots, t_n\} \mid (\forall i)(1 \le i \le n - > t_i \epsilon w_i^2)\}$.

Let the elements of E be $E_1, \ldots, E_e$ and those of F be $F_1, \ldots, F_f$. Let

$$A_{ij} = \{t \mid (\exists t_1)(\exists t_2)(t_1 \epsilon T_D^1 \wedge t_2 \epsilon F_I \wedge t = t_1.t_2)$$
$$V(\exists t_1)(\exists t_2)(t_1 \epsilon E_k \wedge t_2 \epsilon T_D^2 \wedge t = t_1.t_2)$$
$$V(\exists t_1)(\exists t_2)(t_1 \epsilon E_k \wedge t_2 \epsilon F_I \wedge t = t_1.t_2)\}$$

T

$T_{definite}$	
a3	b3
a2	b2
$T_{inclusive}$	
a3	b1
a3	b2
a4	b2
a3	b4
$T_{exclusive}$	
a4	b1
a4	b4
a3	b5
a4	b6
a3	b8
T_{maybe}	
a3	b7
a3	b6

Π_1 (T)

$T_{definite}$
a2
a3
a4
T_{maybe}
a5

Figure 5. An example of the extended projection operation.

where $1 \leq k \leq e, 1 \leq l \leq f$; i=k if ¬(e=0) and i=0 otherwise;

j=l if ¬(f=0) and j=0 otherwise.

Let $A_1, \ldots, A_g$ be the distinct $A_{ij}s$.

Then, $T_1 \times T_2$=ELIMRED(T), where T is defined as follows

$$
\begin{array}{lcl}
T_D & = & \{t \mid (\exists t_1)(\exists t_2)(t\epsilon T_D^1 \wedge t_2\epsilon T_D^2 \wedge t = t_1.t_2)\},\\
T_I & = & \{w \mid (\exists t_1)\ldots(\exists t_g)(t_1\epsilon A_1 \wedge \ldots \wedge t_1\epsilon A_g \wedge w = \{t_1, \ldots, t_g\})\},\\
T_E & = & \{w \mid (\exists t_1)(\exists w_1)(t_1\epsilon T_D^1 \wedge w_1 = \{t_{11}, t_{12}, \ldots, t_{1k}\}\epsilon T_E^2 \wedge w = \{t_1.t_{11}, t_1.t_{12}, \ldots, t_1.t_{1k}\})\\
& & V(\exists w_1)(\exists w_2)(w_1 = \{t_{11}, t_{12}, \ldots t_{1k}\}\epsilon T_E^1 \wedge w_2 = \{t_{21}, t_{22}, \ldots t_{2m}\}\epsilon T_E^2\\
& & \wedge w = \{t_{11}.t_{21}, \ldots, t_{1k}.t_{21}, t_{11}.t_{22}, \ldots, t_{1k}.t_{22}, \ldots, t_{1k}.t_{2m}\})\}\\
T_M & = & \{w \mid (\exists t_1)(\exists w_1)(t_1\epsilon T_D^1 \wedge w_1 = \{t_{11}, t_{12}, \ldots, t_{1k}\}\epsilon T_M^2 \wedge w = \{t_1.t_{11}, t_1.t_{12}, \ldots, t_1.t_{1k}\})\\
& & V(\exists t_1)(\exists w_1)(t_1\epsilon T_D^2 \wedge w_1 = \{t_{11}, t_{12}, \ldots t_{1k}\}\epsilon T_M^1 \wedge w = \{t_{11}.t_1, t_{22}.t_1 \ldots t_{1k}.t_1\})\\
& & V(\exists w_1)(\exists w_2)(w_1 = \{t_{11}, t_{12}, \ldots t_{1k}\}\epsilon T_I^1 \wedge w_2 = \{t_{21}, t_{22}, \ldots t_{2m}\}\epsilon T_M^2\\
& & \wedge(w = \{t_{11}.t_{21}, t_{11}.t_{22}, \ldots, t_{11}.t_{2m}\}\\
& & Vw = \{t_{12}.t_{21}, t_{12}.t_{22}, \ldots, t_{12}.t_{2m}\}\\
& & \ldots\ldots\\
& & \ldots\ldots\\
& & Vw = \{t_{1k}.t_{21}, t_{1k}.t_{22}, \ldots, t_{1k}.t_{2m}\}))\\
& & V(\exists w_1)(\exists w_2)(w_1 = \{t_{11}, t_{12}, \ldots t_{1k}\}\epsilon T_I^2 \wedge w_2 = \{t_{21}, t_{22}, \ldots t_{2m}\}\epsilon T_M^1\\
& & \wedge(w = \{t_{21}.t_{11}, t_{22}.t_{11}, \ldots, t_{2m}.t_{11}\}\\
& & Vw = \{t_{21}.t_{12}, t_{22}.t_{12}, \ldots, t_{2m}.t_{12}\}\\
& & \ldots\ldots\\
& & \ldots\ldots\\
& & Vw = \{t_{21}.t_{1k}, t_{22}.t_{1k}, \ldots, t_{2m}.t_{1k}\}))\\
& & V(\exists w_1)(\exists w_2)(w_1 = \{t_{11}, t_{12}, \ldots t_{1k}\}\epsilon T_E^1 \wedge w_2 = \{t_{21}, t_{22}, \ldots t_{2m}\}\epsilon T_M^2\\
& & \wedge w = \{t_{11}.t_{21}, \ldots, t_{1k}.t_{21}, t_{11}.t_{22}, \ldots, t_{1k}.t_{22}, \ldots, t_{1k}.t_{2m}\})
\end{array}
$$

$$V(\exists w_1)(\exists w_2)(w_1 = \{t_{11}, t_{12}, \ldots t_{1k}\}\epsilon T_M^1 \wedge w_2 = \{t_{21}, t_{22}, \ldots t_{2m}\}\epsilon T_E^2$$
$$\wedge w = \{t_{11}.t_{21}, \ldots, t_{1k}.t_{21}, t_{11}.t_{22}, \ldots, t_{1k}.t_{22}, \ldots, t_{1k}.t_{2m}\})$$
$$V(\exists w_1)(\exists w_2)(w_1 = \{t_{11}, t_{12}, \ldots t_{1k}\}\epsilon T_M^1 \wedge w_2 = \{t_{21}, t_{22}, \ldots t_{2m}\}\epsilon T_M^2$$
$$\wedge w = \{t_{11}.t_{21}, \ldots, t_{1k}.t_{21}, t_{11}.t_{22}, \ldots, t_{1k}.t_{22}, \ldots, t_{1k}.t_{2m}\})\}$$

An example of the cartesian product operation is given in Figure 6.

T_1

$T_{definite}$
a1
$T_{inclusive}$
a2 a3
$T_{exclusive}$
a5 a6
T_{maybe}
a4

T_2

$T_{definite}$
b1
b2
T_{maybe}
b3

$T_1 \times T_2$

$T_{definite}$	
a1	b1
a1	b2
$T_{inclusive}$	
a2 a3	b2 b2
a2 a3	b1 b1
$T_{exclusive}$	
a5 a6	b1 b1
a5 a6	b2 b2
T_{maybe}	
a4	b1
a4	b2
a1	b3
a2	b3
a3	b3
a5 a6	b3 b3

Figure 6. An example of the extended cartesian product operation.

Since $T_1 \bigcap T_2 \neq T_1 - (T_1 - T_2)$ in our extended algebra, we also define the intersection operator.

Definition 2.7 Intersection on Ψ_R is a mapping $\bigcap : \Psi_R \times \Psi_R -> \Psi_R$

Let T_1 and T_2 be two domain-compatible E-tables. Then, $T_1 \bigcap T_2$=ELIMRED(T), where T is defined as follows:

$$\begin{aligned}
T_D &= \{t | t\epsilon T_D^1 \wedge t\epsilon T_D^2\} \\
T_I &= \{w | (w\epsilon T_I^1 \wedge T_D^2 \supset w) \wedge (w\epsilon T_I^2 \wedge T_D^1 \supset w)\} \\
T_E &= \{w | (w\epsilon T_E^1 \wedge T_D^2 \supset w) \wedge (w\epsilon T_E^2 \wedge T_D^1 \supset w)\} \\
T_M &= \{w | (\exists w_1)(\exists t)(w_1 \epsilon T_M^1 \wedge t\epsilon T_D^2 \wedge t\epsilon w_1 \wedge w = \{t\}) \\
&\quad \text{V } (\exists w_1)(\exists t)(w_1 \epsilon T_M^2 \wedge t\epsilon T_D^1 \wedge t\epsilon w_1 \wedge w = \{t\}) \\
&\quad \text{V } (\exists w_1)(\exists t)(w_1 \epsilon T_E^1 \wedge t\epsilon T_D^2 \wedge t\epsilon w_1 \wedge w = \{t\}) \\
&\quad \text{V } (\exists w_1)(\exists t)(w_1 \epsilon T_E^2 \wedge t\epsilon T_D^1 \wedge t\epsilon w_1 \wedge w = \{t\}) \\
&\quad \text{V } (\exists w_1)(\exists w_2)(w_1 \epsilon T_I^1 w_2 \epsilon T_I^2 \wedge t\epsilon w_1 \bigcap w_2 \wedge w = \{t\}) \\
&\quad \text{V } (\exists w_1)(\exists w_2)(\exists t)(w_1 \epsilon T_I^1 \wedge t\epsilon T_D^2 \wedge w = \{t\}) \\
&\quad \text{V } (w_1 \epsilon T_E^1 \wedge w_2 \epsilon T_I^2 \wedge w = w_1 \bigcap w_2) V (w_1 \epsilon T_E^2 \wedge w_2 \epsilon T_I^1 \wedge w = w_1 \bigcap w_2) \\
&\quad \text{V } (w_1 \epsilon T_E^1 \wedge w_2 \epsilon T_M^2 \wedge w = w_1 \bigcap w_2) V (w_1 \epsilon T_E^2 \wedge w_2 \epsilon T_M^1 \wedge w = w_1 \bigcap w_2) \\
&\quad \text{V } (w_1 \epsilon T_I^1 \wedge w_2 \epsilon T_M^2 \wedge w = w_1 \bigcap w_2) V (w_1 \epsilon T_I^2 \wedge w_2 \epsilon T_M^1 \wedge w = w_1 \bigcap w_2) \\
&\quad \text{V } (w_1 \epsilon T_E^1 \wedge w_2 \epsilon T_E^2 \wedge w = w_1 \bigcap w_2) V (w_1 \epsilon T_M^2 \wedge w_2 \epsilon T_M^1 \wedge w = w_1 \bigcap w_2)\}
\end{aligned}$$

An example of the cartesian product operation is given in Figure 7.

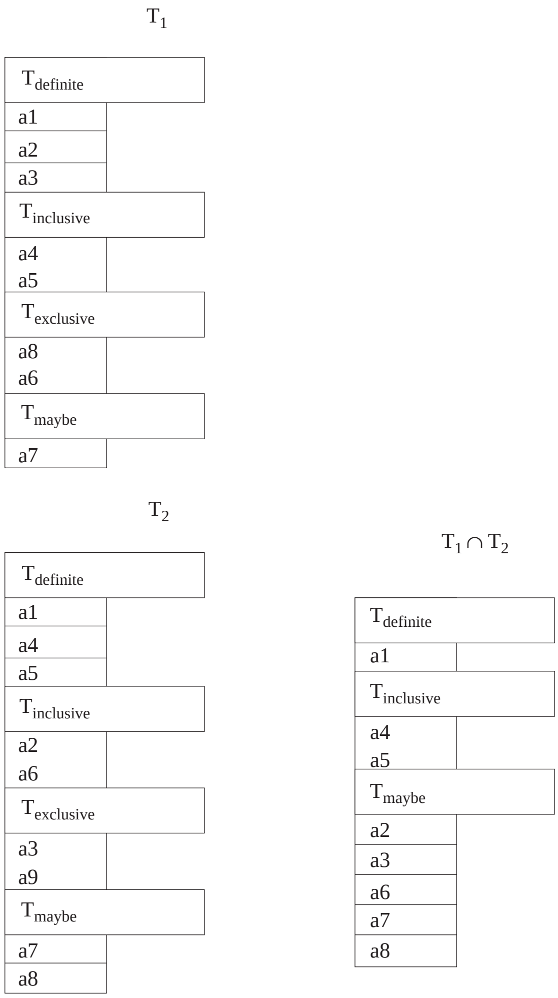

Figure 7. An example of the extended intersection operation.

The use of the new relational algebra in our model can be demonstrated with an example.

Example 2.1: Consider the following query about Mary's teachers:

$\prod_3(\sigma_{2=4}((\sigma_{1='mary'} takes) \times teaches))$.

Assume that we have the following stored E-tables for the relations *takes (student, course)* and *teaches (teacher, course)*.

takes$_D$ ={<john, CS421>, <mary, CS321>, <mary, CS335>} (John takes CS421. Mary takes CS 321. Mary takes CS 335.)

takes$_E$ ={{<john, CS415>, <john, CS495>}} (John takes either CS 421 or CS 495 but not both.)

takes$_I$ ={{<mary, CS 364>, <mary, MATH310>}} (Mary takes CS 364 or MATH 310 or both.)

takes$_M$ ={{<mary, ART201>, <mary, ART301>}} (Mary may take only one of ART 201 and ART 301 or neither.)

teaches$_D$ ={<ann, CS 321>, <jim, CS 364>} (Ann teaches CS 321. Jim teaches CS 364.)

teaches$_E$ ={{<dick, CS 335>, <ed, CS 335>}} (Either Dick or Ed teaches CS335, but not both.)

teaches$_I$ ={ }

teaches$_M$ = {{<tom,ART201>, <tom,ART301>}} (Tom may teach only one of ART201 or ART301 or neither.)

The result of the query about Mary's teachers is an E-table Q as:

Q_D ={<ann>} (Ann is definitely Mary's teacher.)

Q_E={{<dick>, <ed>}} (Either Dick or Ed is Mary's teacher, but not both.)

Q_I={ }

Q_M={{<tom>, <jim>}} (Tom may be Mary's teacher. Jim may be Mary's teacher.)

3. Conclusions

In this paper we presented a structure called an E-table which makes it possible to store exclusive and inclusive disjunctions and maybe information in relational databases. We also redefined the relational algebra operations to include exclusive disjunctions and defined an operator to get rid of redundancies in E-tables. The main contribution of this work is to make possible the storing of exclusive disjunctions, an often needed type of incomplete information. The concepts presented in this paper have been applied to LOGOB[8], a deductive data model with predicates representing object sets.

References

[1] Liu, K. and Sunderraman, R., Indefinite and Maybe Information in Relational Databases, *ACM Trans. on Database Systems,* vol. 15, no. 1, March 1990.

[2] Codd, E. F., A Relational Model for Large Shared Data Banks, *Commun. ACM,* vol. 13, no. 6 (June 1970).

[3] Biskup, J., Extending the Relational Algebra with Maybe Tuples and Existential and Universal Null Values, *Fundamenta Informaticae VII,* 1 (1984), 129-150.

[4] Imielinski, T. and Lipski, W, On Representing Incomplete Information in a Relational Database, *In Proceedings of the 7th International Conference on Very Large Data Bases* (Cannes, Sept. 1981), IEEE, New York, 1981, pp 389-397.

[5] Liu, K. and Sunderraman, R., A Generalized Relational Model for Indefinite and Maybe Information, *IEEE Trans. on Knowledge and Data Engineering,* vol. 3, no. 1, March 1991.

[6] Zaniolo, C., Database Relations with Null Values, *J. Comput. Syst.* Sci., 28, 1 (Feb. 1984), 142-166.

[7] Lipski, W., Jr., On Databases with Incomplete Information, *Journal of the ACM,* Vol. 28, no. 1, Jan. 1981.

[8] Savas, E. and Gundem, T., Handling Incomplete Information in LOGOB, In *Proceedings of the 7th International Symposium on Computer and Information Sciences,* (Antalya, Nov. 1992).