Correspondence

A Note on "Incomplete Relational Database Models Based on Intervals"

Jui-Shang Chiu and Arbee L. P. Chen

Abstract—In [5], a family of relational database models (M-1 to M-5) were proposed to represent unknown values by intervals. Relational operators were extended for evaluating queries on these models. In this note, we stultify the theorems claiming that query evaluation in models M-2, M-3, and M-5 is sound.

Index Terms—Incomplete information, disjunctive information, null values, extended relational model, extended relational algebra.

1 Introduction

In [5], relations were extended to represent unknown values by intervals. Each unknown value (or tuple) may have a unique identifier. Four tuple types were defined to specify *existence* and *uniqueness* relationships among tuples of the same *table* (i.e., extended relation). In addition, tuples of different tables are distinguished between the cases where incompleteness is introduced at the *relation level, tuple level,* or *attribute value level.* Based on these relationships among tuples in different tables, Ola and Ozsoyoglu presented a family of incomplete relational database models (M-1 to M-5).

For each of the models (M-1 to M-4), Ola and Ozsoyoglu showed that query evaluation is *sound* (i.e., no invalid results are derivable) but is not *complete* (i.e., all valid results are derivable) in the Imielinski-Lipski sense [1] when only fully defined and definite (i.e., *total*) tuples are concerned. They also claimed that query evaluation in model M-5 is sound and complete. In this note, we point out several flaws in [5]:

- The definition of the correctness conditions is not correct in the sense that it does not match the statements "no invalid total results are derivable" and "all valid total results are derivable."
- The theorems claiming that query evaluation in models M-2 and M-3 are sound, and in model M-5 is sound and complete, are incorrect.
- The statement about maintaining the COUNT (i.e., the bounds of the number of tuples in the unknown relation) to avoid information loss has no warrant.

2 The Models

In the following, we briefly review the characteristics of the models in [5], in particular, models M-2, M-3, and M-5.

The partial knowledge about an unknown value is specified as possible values in a set of intervals. In addition, four tuple types are allowed in a table.

1) A *type 0* (*total*) tuple is the usual relation tuple without unknown values.
2) A *type 1* partial tuple represents an unknown tuple which *exists,* and is *uniquely* represented.
3) A *type 2* partial tuple represents an unknown tuple which *exists,* but may already be represented by other tuples.
4) A *type 3* partial tuple represents an unknown tuple which *may or may not exist.*

2.1 Assumptions for Model M-2

Each unknown value has a unique identifier. Different occurrences of the same unknown value in the database have the same identifier and the same range value. If two identifiers for two unknown value τ_1 and τ_2 are the same then $\tau_1 = \tau_2$; otherwise τ_1 may or may not be equal to τ_2.

Also note that the identifier for a known value is the value itself.

2.2 Assumptions for Model M-3

Two identifiers for two unknown values τ_1 and τ_2 are the same iff $\tau_1 = \tau_2$.

2.3 Assumptions for Model M-5

Model M-5 is a restricted model of M-2, with additional assumptions and restricted operators described as follows. Every table has key attributes which uniquely determine tuples in the table. Unknown values are not allowed on key attributes. Moreover, the projection is restricted such that the projection attributes always contain the key attributes, and the join is restricted such that the key attributes of each join table are always contained in the join attributes. Also note that type 2 tuples will not be introduced in model M-5 since every tuple is uniquely determined by its key attributes. As an example, table U shown in Table 1 is in model M-5 where the key attribute is A_1, and the unknown value represented by interval [3, 4] has an identifier i. Both r_1 and r_2, also shown in Table 1, are possible relations for the unknown relation represented by U.

TABLE 1
A Table in Model M-5 and Its Possible Relations

U

A_1	A_2	TYPE
1	3	0
2	$[3, 4]_i$	1

r_1

A_1	A_2
1	3
2	3

r_2

A_1	A_2
1	3
2	4

3 The Correctness Notion

The *information content* of a table is defined by the mapping *rep* which maps a table U to a set of possible relations for the unknown relation represented by U [1]. In defining the *rep,* a generalized version of Reiter's *closed world assumption* was adopted [3], [6]: a tuple t is not in the unknown relation represented by U if there is no relation r in $rep(U)$ such that t is in r.

It is well-accepted that the correctness condition for query evaluation on the extended model is defined as follows:

Definition 2.1. [1], [2] Let $U_1, U_2, \ldots, U_k$ be tables. Let f be a relational algebra (RA) expression and f^* be the extended version of f. Query evaluation is sound and complete if

$$rep(f^*(U_1, U_2, \ldots, U_k)) = f(rep(U_1), rep(U_2), \ldots, rep(U_k)).$$

$$f(rep(U_1), rep(U_2), \ldots, rep(U_k))$$

is understood to be

$$\{f(r_1, r_2, \ldots, r_k) \mid r_j \in rep(U_j), j = 1, 2, \ldots, k\}.$$

• *The authors are with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan 30043, Republic of China.*
E-mail: alpchen@cs.nthu.edu.tw.

Manuscript received Sept. 7, 1994.
For information on obtaining reprints of this article, please send e-mail to: transactions@computer.org, and reference IEEECS Log Number K95075.

1041-4347/96$05.00 ©1996 IEEE

However, this ideal condition is hard to achieve. Practically, it is desirable to define the extended relational operators in a semantically meaningful way in which $rep(f^*(U_1, U_2, \ldots, U_k))$ approximates the information given by $f(rep(U_1), rep(U_2), \ldots, rep(U_k))$. Therefore, the *f-information* concept [1] was adopted in [5].

DEFINITION 2.2. Let a database instance D be a sequence of relations $<r_1, r_2, \ldots, r_k>$ with scheme $<R_1, R_2, \ldots, R_k>$, and f be an RA expression involving the relations in D. Let X be a set of database instances D. The *f-information* in X, denoted by X^f, is defined as

$$X^f = \bigcap_{D \in X} f(D)$$

where $f(D)$ denotes the relation obtained by substituting r_j for every occurrence of R_j in f, $j = 1, 2, \ldots, k$. That is, X^f is the largest relation s such that $s \subseteq f(D)$, for all D in X.

In essence, the f-information of an expression f consists of fully defined and definite tuples (i.e., total tuples) that can be derived when f is evaluated against the database. It is desirable, for any RA expression f, that $\cap rep(f^*(X)) = \cap f(rep(X))$. Hence, the soundness (i.e., no incorrect total tuples are derivable) and completeness (i.e., all valid total tuples are derivable) conditions can be defined as follows.

DEFINITION 2.3. Query evaluation is *sound* if $\cap rep(f^*(X)) \subseteq \cap f(rep(X))$. It is *complete* if $\cap rep(f^*(X)) \supseteq \cap f(rep(X))$.

In [5], the soundness and completeness conditions were given as follows.

DEFINITION 2.4. [5] Let U_1 and U_2 be two tables. Let α and θ be unary and binary operators, respectively, i.e., $\alpha \in \{\pi, \sigma\}$ and $\theta \in \{-, \cup, \bowtie\}$. Let α^* and θ^* be the extended versions of α and θ, respectively.

1) An extended model is *sound* if, for every relation r in $(rep(U_1)\ \theta\ rep(U_2))$ or $\alpha(rep(U_1))$, there is a relation s in $rep(U_1\ \theta^*\ U_2)$ or $(rep(\alpha^*(U_1))$, respectively, such that $s = r$. That is, $rep(\alpha^*(U_1)) \supseteq \alpha(rep(U_1))$ and $rep(U_1\ \theta^*\ U_2) \supseteq rep(U_1)\ \theta\ rep(U_2)$.
2) An extended model is *complete* if, for every relation s in $rep(U_1\ \theta^*\ U_2)$ or $rep(\alpha^*(U_1))$, there is a relation r in $(rep(U_1)\ \theta\ rep(U_2))$ or $\alpha(rep(U_1))$, respectively, such that $r \subseteq s$.

Note that Definition 2.4 is not correct in the sense that it does not match the statements "no invalid total tuples are derivable" and "all valid total tuples are derivable." In fact, the two conditions together in Definition 2.4 do not conclude that $\cap rep(f^*(X)) = \cap f(rep(X))$.

In the next section, we shall show that models M-2, M-3 do not satisfy the soundness conditions in Definition 2.4.

4 THE CORRECTNESS THEOREM

A theorem in [5] claimed that query evaluation of RA expression f^* in model M-5 is sound and complete where f^* consists of extended operators in $\{\pi^*, \sigma^*, \cup^*, \bowtie^*, -^*\}$. We give a counter-example in the following to stultify the theorem. Consider table U_1 shown in Table 2 and query $\pi^*_{A_1}\left(\sigma^*_{A_2=2}(U_1)\cup^*\sigma^*_{A_2=3}(U_1)\right)$ where the key attribute is A_1. It is easy to see that any valid result contains exactly one tuple. However, according to the extended algebra defined in [5], both $\sigma^*_{A_2=2}(U_1)$ and $\sigma^*_{A_2=3}(U_1)$ result in a type 3 tuple. So do the union of two type 3 tuples and the projection of a type 3 tuple, which may then produce an empty result. Unfortunately, the empty result violates the "completeness" condition of both definitions.

Recall from Section 3 we pointed out that the correctness conditions in [5], i.e., Definition 2.4 are too restricted. Consider query $\sigma^*_{A_2=2}(U_1)\cup^*\sigma^*_{A_2=3}(U_1)$. The query results in no total tuples. When only total tuples are concerned, the empty result of this query should be regarded as sound and complete. It satisfies Definition 2.3. However, it does not satisfy Definition 2.4.

TABLE 2
TABLES INVOLVED IN COUNTER-EXAMPLES

U_1

A_1	A_2	TYPE
1	$[2,3]_i$	1

U_2

A_3	A_2	TYPE
4	2	0
5	3	0

Now we show that both models M-2 and M-3 are not sound by another counter-example. Consider tables U_1 and U_2 shown in Table 2 and query $U_1 \bowtie^* U_2$. Again, any valid result contains exactly one tuple. However, in both models M-2 and M-3, two tuples are *joinable* only if all of the identifiers in join attributes are identical. The query will, therefore, yield an empty table. Since $r_1 \bowtie r_2 \neq s = \emptyset$ for any $r_1 \in rep(U_1)$, $r_2 \in rep(U_2)$ and $s \in rep(U_1 \bowtie^* U_2)$, it violates the "soundness" condition in Definition 2.4.

We also note that

1) The example in Fig. 11 in [5] is incorrect, which showed that there are no relations $r_1 \in rep(U)$ and $r_2 \in rep(V)$ such that $r_1 \bowtie r_2 \subseteq s$. In that example, the reader can easily find r_1 and r_2 such that $r_1 \bowtie r_2 \subseteq s$. A correct example is given in Table 3.
2) The example in Fig. 15 in [5] does not follow the assumption in model M-5 that unknown values are not allowed in key attributes.

TABLE 3
A COUNTER-EXAMPLE FOR THE "COMPLETENESS" OF JOIN IN M-1

U

A_1	A_2	TYPE
3	2	0
2	2	0

V

A_2	A_3	TYPE
2	[2,4]	1
2	6	0

$U \bowtie^* V$

A_1	A_2	A_3	TYPE
3	2	[2,4]	1
2	2	[2,4]	1
3	2	6	0
2	2	6	0

s

A_1	A_2	A_3
3	2	2
2	2	3
3	2	6
2	2	6

In the following, comments on tuple types and identifiers of unknown values are given. One of the emphases in [5] is the introduction of tuple types for a tuple's existence and uniqueness relationships. However, under the assumption that every tuple has a unique key, as in model M-5, the indication of the uniqueness is actually unnecessary. Moreover, the existence-and-uniqueness (type 1) relationship may often be lost (and the migration of tuples is then required) in the query evaluation involving projection, selection, union, join, or difference. Furthermore, the tuple type is not enough to prevent information loss. For example, in the selection operator, there is no way in [5] to distinguish which part of an interval is satisfiable and which part is not. Hence, the keeping of the type 1 information may often be useless, which also complicates the query processing.

On the other hand, identifiers are introduced to indicate different occurrences of the same unknown value in the database. The *definition* of join in [5] considering the identifiers is, however, inadequate. For example, it is possible to join two intervals $[1, 2]_{i_1}$ and $[2, 3]_{i_2}$, even though their identifiers are different, as long as both intervals turn out to represent the same value, i.e., 2.

5 The COUNT Information

A statement in Section VIII of [5] claimed that the models can be extended to avoid information loss caused by the union and difference operations. They suggested to maintain explicit bounds on the COUNT of the number of tuples in the unknown relation represented by a table. An extension with the COUNT range attached to tables was given in [4]. In the extension, they showed that query evaluation is sound and complete (with respect to Definition 2.1) for expressions consisting of only selection, difference, projection and Cartesian product.

In the following, we give an example to show that the COUNT is not enough to prevent information loss in the difference operation, which stultifies the aforementioned statement. Consider the tables U_1 and U_2 shown in Table 4 and query $U_1 \overset{*}{-} U_2$. The number of tuples in the resulting relation represented by U_3 based on [5] is between 0 and 2. However, the COUNT is not enough to prevent relation $r = \{2, 3\}$ from being in $rep(U_3)$. Moreover, the COUNT information is derived from the results evaluated on all the possible relations represented by a table. Since the number of possible relations can be extremely huge, it is simply impractical.

TABLE 4
An Example of Difference

$U_1^{(2,2)}$

A_1	TYPE
[1,2]	1
[1,3]	1

$U_2^{(2,2)}$

A_1	TYPE
[2,3]	1
[2,4]	1

$U_3^{(0,2)}$

A_1	TYPE
[1,2]	3
[1,3]	3

Acknowledgment

This research was partially supported by the Republic of China National Science Council under Contract No. NSC 84-2213-E-007-007.

References

[1] T. Imielinski and W. Lipski, "Incomplete information in relational databases," *J. ACM*, vol. 31, no. 4, pp. 761-791, 1984.

[2] D. Maier, *The Theory of Relational Databases*. Rockville, Md.: Computer Science Press, 1983.

[3] J. Minker, "On indefinite databases and the closed world assumption," *Lecture Notes in Computer Science, N138*. New York: Springer-Verlag, pp. 292-308, 1982.

[4] A. Ola, "Relational databases with exclusive disjunctions," *Proc. IEEE Data Engineering*, pp. 328-336, 1992.

[5] A. Ola and G. Ozsoyoglu, "Incomplete relational database models based on intervals," *IEEE Trans. Knowledge and Data Engineering*, vol. 5, no. 2, pp. 293-308, 1993.

[6] R. Reiter, "Towards a logical reconstruction of relational database theory," *On Conceptual Modeling*. New York: Springer-Verlag, pp. 191-233, 1984.