JOURNAL OF COMPUTER AND SYSTEM SCIENCES 21, 368–379 (1980)

Adequacy of Decompositions of Relational Databases*,†

DAVID MAIER

State University of New York, Stony Brook, New York 11794

ALBERTO O. MENDELZON

IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598

FEREIDOON SADRI

Princeton University, Princeton, New Jersey 08540

AND

JEFFREY D. ULLMAN

Stanford University, Stanford, California 94305

Received August 6, 1979; revised July 16, 1980

We consider conditions that have appeared in the literature with the purpose of defining a "good" decomposition of a relation scheme. We show that these notions are equivalent in the case that all constraints in the database are functional dependencies. This result solves an open problem of Rissanen. However, for arbitrary constraints the notions are shown to differ.

I. BASIC DEFINITIONS

We assume the reader is familiar with the relational model of data as expounded by Codd [11], in which data are represented by tables called *relations*. Rows of the tables are *tuples*, and the columns are named by *attributes*. The notation used in this paper is that found in Ullman [23].

A frequent viewpoint is that the "real world" is modeled by a *universal* set of attributes R and constraints on the set of relations that can be the "current" relation for scheme R (see Beeri *et al.* [4], Aho *et al.* [1], and Beeri *et al.* [7]). The *database scheme* representing the real world is a collection of (nondisjoint) subsets of the

* Work partially supported by NSF Grant MCS-76-15255. D. Maier and A. O. Mendelzon were also supported by IBM fellowships while at Princeton University, and F. Sadri by a scholarship from Isfahan University of Technology.

† Authors' current addresses: D. Maier, SUNY, Dept. of CS, Stony Brook, N. Y. 11794; A. O. Mendelzon, University of Toronto, Department of Computer Science, Toronto, Canada M5S 1A7; F. Sadri and J. D. Ullman, Stanford University Dept. of CS, Stanford, Calif. 94305.

0022-0000/80/060368-12$02.00/0

univesal relation scheme R; each of these subsets is called a *relation scheme*. The *database* is an assignment of *relations* (sets of tuples) to the relation schemes. The *database design problem* is to pick a database scheme $\rho = (R_1, \ldots, R_k)$ with $\bigcup_{i=1}^{k} R_i = R$, such that, informally:

(1) The "current" relation for R, which does not really exist in the database, can be discovered from the "current" values of the R_i's, which do exist in the database.

(2) The constraints on the legal relations for scheme R can be enforced by constraints on the relations for the R_i's.

(3) The R_i's have certain desirable properties, usually connected with the constraints, such as lack of redundancy.

While the above notions are informal, precise definitions have been given; Beeri *et al.* [4] survey these ideas. Briefly, (1) is formalized as the lossless join property of Aho *et al.* [1], which we shall define. Item (2) is usually taken to mean embedability of dependencies (also to be defined) as in Bernstein [10], while (3) is taken to refer to certain "normal forms," as defined, for example, by Codd [12] and Fagin [14, 15].

Projections and Joins

The key assumption on which our theory of database design rests is the formalization of "representation" by the algebraic operation of projection (see Rissanen [19], e.g.). If r is a relation over set of attributes R, and $S \subseteq R$, then $\pi_S(r)$ is the set of tuples of r with components in columns for attributes in $R - S$ removed. For example, if $R = \{A, B, C\}$, and r is

A	B	C
0	1	2
2	1	3
0	3	2

then $\pi_{AC}(r)$ is

A	C
0	2
2	3

If $\rho = (R_1, \ldots, R_k)$ is a database scheme for universal relation scheme R, and r is a relation for R, then we use $\pi_\rho(r)$ to stand for the list $(\pi_{R_1}(r), \ldots, \pi_{R_k}(r))$.

If t is a tuple, $t[S]$ is the components of t for the attributes in set S. The (*natural*) *join* of relations $r_1, \ldots, r_k$, whose schemes are $R_1, \ldots, R_k$, respectively, is the relation r over scheme $R = \bigcup_{i=1}^{k} R_i$ such that tuple t is in r if and only if for each i, $1 \leqslant i \leqslant k$, $t[R_i]$ is in r_i. We use symbol $\times$ for the join operation. Thus, if r_1 and r_2 are

A	B
0	1
1	2
2	1

,

B	C
3	1
1	2
2	0

then $r_1 \times r_2$ is

A	B	C
0	1	2
1	2	0
2	1	2

.

Constraints and Dependencies

Often, the real world is not modeled by an arbitrary relation over the universal scheme R, but only by a subset of the possible relations chosen to represent some "physical constraints." The most common sorts of constraints are functional (Codd [11, 12], Armstrong [2]) and multivalued (Fagin [14], Beeri *et al.* [5], Zaniolo [25], Delobel [13]) dependencies. If X and Y are subsets of universal scheme R, we say that *functional dependency* $X \to Y$ holds if every relation r that is allowable as a "current" relation for scheme R has the property that if t_1 and t_2 are tuples in r, and $t_1[X] = t_2[X]$, then $t_1[Y] = t_2[Y]$. The *multivalued dependency* $X \to\to Y$ holds if for each two tuples t_1 and t_2 in r, where $t_1[X] = t_2[X]$, there is a tuple t_3 in r such that $t_3[X] = t_1[X] = t_2[X]$, $t_3[Y] = t_1[Y]$, and $t_3[R - X - Y] = t_2[R - X - Y]$.

For example, if r is

A	B	C	D
0	1	2	3
0	2	3	2
0	2	2	2
0	1	3	3

then the functional dependencies $D \to B$, $BC \to D$, and $C \to A$ are seen to hold, as do the multivalued dependencies $A \to\to C$ and $B \to\to CD$. We show $A \to\to C$, e.g., by observing that for each A-value (0 is the only one) the possible C-values (2 and 3) are paired with the possible BD-values (1 3 and 2 2) in all possible ways in r.

In this paper, we shall assume that for each universal relation scheme R there is an associated set L of legal relations for R. If we are given a set of functional and/or multivalued dependencies D, then we let SAT(D) be the set of relations that satisfy all the dependencies in D, and we take $L = \text{SAT}(D)$. We use $\pi_S(L)$ for $\{\pi_S(r) \mid r \text{ is in } L\}$ and we use $\pi_\rho(L)$, where ρ is the database scheme $(R_1, \ldots, R_k)$, for $\{\pi_\rho(r) \mid r \text{ is in } L\}$.

Lossless Joins

Often, the condition that a database scheme $\rho = (R_1, ..., R_k)$ represent the universal relation, i.e., that any legal universal relation be "represented" by its projections, is equated with the *lossless join condition*

$$\text{for all } r \text{ in } L, \qquad \mathop{\times}_{i=1}^{k} \pi_{R_i}(r) = r.$$

A test for this condition, when $L = \text{SAT}(D)$ for a set of functional and multivalued dependencies D, was given by Aho *et al.* [1].

Preservation of Constraints under Projection

The second desired property of database schemes is that if r is a relation over universal scheme R, and $\rho = (R_1, ..., R_k)$ is a database scheme, then whenever the list of relations $\pi_\rho(r)$ satisfies "projected constraints," it follows that r satisfies the constraints themselves. The notion of "projected constraint" is well defined for functional and multivalued dependencies. If D is a set of such dependencies, let D^+ be the *closure* of D, that is, the set of dependencies that follow logically from D in the sense that in every relation r for which D holds, D^+ also holds. The computation of D^+ was explained by Beeri *et al.* [5]. If $S \subseteq R$, then $\pi_S(D)$, the dependencies that hold in $\pi_S(R)$ for every relation r that satisfies D, consists of those $X \to Y$ and $X \to\to Y$ such that

1. there is some $X \to Z$ or $X \to\to Z$ in D^+,
2. $X \subseteq S$, and
3. $Y = Z \cap S$

(Aho *et al.* [1]). For functional dependencies, we can always assume $Y = Z$.

For functional dependencies only, the dependencies in $\pi_{R_i}(D)$ may be taken to hold in R, rather than R_i, and the question of logical implication can be decided by the techniques of Bernstein [10], for example. However, the projection of a multivalued dependency onto S becomes an "embedded dependency," which, while it holds in S, is not the same as having the same dependency holds in R. Properties of embedded dependencies are not well understood, and it is not clear how to test for logical implications.

In the most general case, the formalization of the condition that the projected constraints logically imply the original constraints is as follows. Let L be the set of legal relations for universal scheme R, and let $\rho = (R_1, ..., R_k)$ be a database scheme. Then we say ρ ***preserves constraints*** if whenever for all i, $\pi_{R_i}(r)$ is in $\pi_{R_i}(L)$, it follows that r is in L.

II. Notions of Adequate Representation

We now turn to several conditions that appear to combine the notions of representability and constraint preservation.

Rissanen's Approach

The first is the condition called *independent components* (IC) by Rissanen [19]. Let L be the set of legal relations for universal scheme R and let $\rho = (R_1, \ldots, R_k)$ be a database scheme. Let P be the set of databases $\sigma = (r_1, \ldots, r_k)$ such that

1. For some relation r (not necessarily in L), $\sigma = \pi_\rho(r)$.
2. For $1 \leqslant i \leqslant k$, r_i is in $\pi_{R_i}(L)$.

Obviously $\pi_\rho(L) \subseteq P$, but the inclusion could be proper. Figure 1 shows the relationship between these sets.

We say ρ is a *decomposition into independent components* if the following hold.

(IC1) If σ is in P, then there is at most one r in L such that $\pi_\rho(r) = \sigma$.

(IC2) If σ is in P, then there is at least one r in L such that $\pi_\rho(r) = \sigma$.

Obviously (IC1) and (IC2) could be combined into the single statement "π_ρ is a bijection from L onto P," but we choose to write them this way for comparison with other definitions to be made later. The motivation behind the IC conditions is that a unique universal relation in L is represented by every member of P. In turn, P is the set of databases that might arise in practice if we assume the database is always the projection of a universal instance and check constraints in the relations individually, as updates to the relations are made. That is to say, the databases that we expect to occur are in one-to-one correspondence with the universal instances we expect to occur, so the former can be fairly said to represent the latter.

Arora and Carlson's Approach

A second point of view regarding adequate representation was expressed by Arora and Carlson [3]. Their approach (suitably generalized to arbitrary constraints) is to

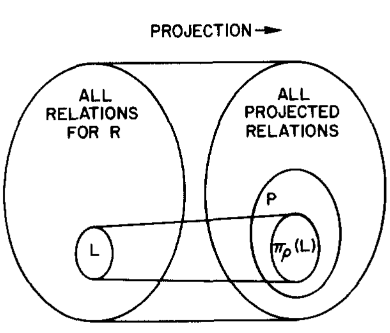

FIG. 1. Diagram of relevant sets of relations and projections.

take the lossless join and constraint preservation conditions together as the definition of an adequate decomposition. We shall relate the Arora – Carlson conditions to independent components by showing equivalence of the above to the following two conditions.

(AC1) = (IC1).

(AC2) If σ is in P, and $\pi_\rho(r) = \sigma$, then r is in L.

THEOREM 1. *Conditions* (AC1) *and* (AC2) *hold if and only if the lossless join and constraint preservation properties hold.*

Proof. (If) Suppose (AC1), which is the same as (IC1), does not hold. Then there are r_1 and r_2 in L such that for some σ in P, $\pi_\rho(r_1) = \pi_\rho(r_2) = \sigma$. Let $\sigma = (r_1, ..., r_k)$, and $\bowtie_{i=1}^k r_i = r$. Then r must be both r_1 and r_2 by the lossless join condition. Thus (AC1) holds. Now suppose (AC2) does not hold, that is, there is a relation r, not in L, such that $\pi_\rho(r) = \sigma$ is in P. We then directly contradict the assumption that ρ preserves constraints.

(Only if) Suppose the lossless join condition is violated. Then there is a relation r in L such that $r' = \bowtie_{i=1}^k \pi_{R_i}(r) \neq r$. It is easy to show that $\pi_{R_i}(r') = \pi_{R_i}(r)$. Thus, $\pi_\rho(r') = \pi_\rho(r)$. Since r is in L, $\pi_\rho(r)$ must be in P. If r' is not in L, we violate (AC2), while if r' is in L, we violate (AC1). Lastly, if constraints are not preserved, then there is r not in L such that $\pi_\rho(r)$ is in P, immediately contradicting (AC2). ∎

COROLLARY. *The lossless join and constraint preservation conditions imply the* IC *conditions.*

Proof. (AC1) is (IC1). Suppose (AC2) holds, while (IC2) does not. For each σ in P there must be some r such that $\pi_\rho(r) = \sigma$. If r is in L, then (IC2) holds; if r is not in L, then (AC2) is violated. ∎

The equivalence of IC and AC was shown by Rissanen [19] for the case that ρ consists of two relation schemes, and $L = \mathrm{SAT}(D)$ for a set of functional dependencies D. We shall generalize this result to arbitrary numbers of relation schemes.

An Intermediate Condition

Let us now introduce a condition that lies between the AC and IC conditions. While the IC conditions guarantee that every database in P represents a unique universal relation in L, we might not be able to find that relation conveniently, given σ in P. Probably the most natural way to go from $\sigma = (r_1, ..., r_k)$ in P to the instance that it represents is to take the join of the r_i's. However, it is conceivable that $r = \bowtie_{i=1}^k r_i$ is not in L, yet there is still some unique r' in L such that $\pi_\rho(r') = \sigma$. We therefore propose that a more realistic definition of adequacy for decompositions is the following. (The same proposal was made independently by Beeri and Rissanen [8].)

(J1) = (IC1) = (AC1).

(J2) If $\sigma = (r_1, ..., r_k)$ is in P, then $\bigtimes_{i=1}^{k} r_i$ is in L.

The *join condition*, above, has the following factors in its favor.

1. We shall show that in some cases it is more general than AC.

2. The universal instance represented by a database σ in P is effectively constructible from σ, while for IC, this may not be the case.

We shall now explore some of the elementary properties of condition J.

THEOREM 2. (AC) *implies* (J) *implies* (IC).

Proof. Suppose (AC2) holds, but (J2) does not. Then there is $\sigma = (r_1, ..., r_k)$ in P such that $r = \bigtimes_{i=1}^{k} r_i$ is not in L. But $\pi_\rho(r) = \sigma$, so (AC2) is violated. Now suppose (J2) holds. Then $r = \bigtimes_{i=1}^{k} r_i$ is in L, and $\pi_\rho(r) = \sigma$, so (IC2) holds. ∎

THEOREM 3. *Suppose D is a set of functional dependencies, and* $L = \text{SAT}(D)$. *Then* (IC), (AC) *and* (J) *are equivalent.*

Proof. By Theorems 1 and 2 it suffices to show that if either the lossless join or constraint preservation properties fail to hold, then (IC) fails. Suppose first that the set of functional dependencies D is not preserved by the decomposition $\rho = (R_1, ..., R_k)$. This means that there exists some relation r not in SAT(D) such that $\pi_{R_i}(r)$ is in $\pi_{R_i}(\text{SAT}(D))$ for all i, and hence r satisfies all the dependencies $\pi_{R_i}(D)$. It follows that $E = (\bigcup_{i=1}^{K} \pi_{R_i}(D))^+$ does not contain D^+. In particular, let $X \to A$ be a dependency in D^+ that is not in E. Let Y be the set of attributes B such that $X \to B$ is in E. Surely $X \subseteq Y$ and A is not in Y.

Let r_0 be a universal relation with exactly two tuples. These tuples agree on all attributes of Y and disagree on all other attributes. Then $X \to A$ does not hold in r_0, so r_0 is not in $L = \text{SAT}(D)$. However, $\sigma = \pi_\rho(r_0)$ is in P, since r_0 is easily seen to satisfy all the dependencies in E. We now claim that σ cannot be $\pi_\rho(r)$ for any r in L, thus violating (IC2). For any such r must have only one symbol appearing in the column of any attribute in Y, and it must also have two different symbols for those attributes not in Y. It follows that r must violate $X \to A$. This shows that (IC) implies constraint preservation.

Now suppose the lossless join condition fails. Let r be a relation in L such that $\sigma = \pi_\rho(R)$ and $\bigtimes \sigma = r' \neq r$. If r' is in L, IC1 is violated. If r' is not in L, constraint preservation is violated, since σ satisfies all the projected dependencies $\pi_\rho(D)$. As we just showed, this implies that IC fails. ∎

COROLLARY. *There is a polynomial time algorithm to test if condition* (IC) *is satisfied, if L is determined by functional dependencies.*

Proof. By Theorems 1 and 3, we have only to test the lossless join and constraint preservation conditions. The lossless join condition can be tested in polynomial time by the algorithm of Aho *et al.* [1]. Preservation of functional dependencies is testable in polynomial time by the algorithm of Beeri and Honeyman [6]. ∎

We have mentioned that the proof of Theorem 3 does not generalize to multivalued dependencies. In fact, we can say more: the theorem is false in this case.

Consider $R = \{A, B, C\}$, and let D consist of the one multivalued dependency $A \rightarrow\rightarrow B$ (which implies $A \rightarrow\rightarrow C$ by the complementation rule of Beeri *et al.* [5]). If $\rho = (AB, AC)$, then ρ has the losless join property (Fagin [14]), and hence, as in the proof of Theorem 1, ρ satisfies (AC1). However, the projected dependencies $A \rightarrow\rightarrow B$ in AB and $A \rightarrow\rightarrow C$ in AC are trivial embedded dependencies and do not imply $A \rightarrow\rightarrow C$ in ABC. Put another way, if $L = \mathrm{SAT}(D)$, then any relation whatsoever is in $\pi_{AB}(L)$ and $\pi_{AC}(L)$. Thus membership of $\pi_{AB}(r)$ and $\pi_{AC}(r)$ in P surely does not imply that r is in L. We therefore have an example where $\rho = (AB, AC)$ does not satisfy (AC). However, the join $\pi_{AB}(r) \times \pi_{AC}(r)$ musst satisfy $A \rightarrow\rightarrow B$ for any relation r over ABC, so ρ satisfies (J).

Weak Constraint Preservation

The example above suggests that the lossless join and constraint preservation conditions may be too strong to characterize all desirable decompositions. In fact, if we assume that the instance represented by a database is the join of the relations in the database, then we should take into account the fact that there are other constraints that must necessarily hold in the join, besides those implied by the projected constraints.

Given a decomposition $\rho = (R_1, \ldots, R_k)$, let FIXPT($\rho$) be the set of relations r such that $\times_{i=1}^{k} \pi_{R_i}(r) = r$. We relax the constraint preservation condition as follows.[1]

Given the set of legal relations L for universal scheme R, we say that ρ *weakly preserves constraints* if whenever r is in FIXPT(ρ) and for all i, $\pi_{R_i}(r)$ is in $\pi_{R_i}(L)$, it follows that r is in L. We can now show that, for arbitrary L, the (J) condition is equivalent to lossless join and weak preservation.

THEOREM 4. *Conditions* (J1) *and* (J2) *hold if and only if the lossless join and weak constraint preservation properties hold.*

Proof. (If) We have seen in the proof of Theorem 1 that the lossless join property implies (J1). Now suppose that $\pi_\rho(r)$ is in P. Let $r' = \times_i \pi_{R_i}(r)$. Since r' is in FIXPT(ρ), and $\pi_\rho(r') = \pi_\rho(r)$ is in P, weak constraint preservation implies r' is in L, proving (J2).

(Only if) Suppose ρ satisfies (J), and let r be in L. By (J2), $r' = \times \pi_\rho(R)$ is in L. Since $\pi_\rho(r) = \pi_\rho(r')$, (J1) implies $r = r'$, proving the lossless join property. Now let r be in FIXPT(ρ) and let $\pi_\rho(r)$ be in P. By (J2), $\times \pi_\rho(r)$ is in L. Since r is in FIXPT(ρ), we have $r = \pi_\rho(r)$, proving weak preservation. ∎

[1] The same proposal was made independently by Beeri and Rissanen [8], who also proved results similar to Theorems 3 and 4.

Adequacy of 4NF *Decompositions*

The literature shows that certain undesirable anomalies in the updating of the database can be avoided when the database scheme is in a *normal form* with respect to the given dependencies. Progressively more general formulations of the normal form concept are third normal form [12], fourth normal form (4NF) [14, 15] and project-join normal form [15]. We shall define 4NF and show that the J and IC conditions are equivalent for 4NF decompositions. We shall also show how to test for the J or IC conditions on a 4NF decomposition.

Suppose the set of legal relations L is defined by a set of fd's D and mvd's M. Let $D_i = \pi_{R_i}(D)$, $M_i = \pi_{R_i}(M)$. A relation scheme R_i is in 4NF if every mvd $m \in M_i$ is implied (in the context of R_i) by an fd of the form $X \to R_i \in D_i$. A database scheme ρ is in 4NF if every $R_i \in \rho$ is in 4NF.

We shall need some preliminary results for the proof of our theorem. The next theorem is due to Sagiv and Fagin [21].

THEOREM 5. *Let D be a set of fd's, M a set of mvd's, and d a functional or multivalued dependency. If* $\mathrm{SAT}(D \cup M) \not\subseteq \mathrm{SAT}(d)$, *then there exists a relation r with only two tuples in it such that* $r \in \mathrm{SAT}(D \cup M) - \mathrm{SAT}(d)$.

We define M_ρ as the set of all mvd's that must be satisfied by every relation in $\mathrm{FIXPT}(\rho)$. For example, for $\rho = \{AB, BC, CD\}$, it can be seen that $M_\rho = \{B \twoheadrightarrow A, C \twoheadrightarrow D\}^+$.

LEMMA 1. *Let r be a relation containing only two tuples. If* $r \in \mathrm{SAT}(M_\rho)$, then $r \in \mathrm{FIXPT}(\rho)$.

Proof. Let $r = \{t_1, t_2\}$ and $\rho = \{R_1, ..., R_k\}$. Suppose $r \notin \mathrm{FIXPT}(\rho)$. Let $r' = \times_i(\pi_{R_i}(r))$. Say $t \in r' - r$. Assume without loss of generality that $t[R_i] = t_1[R_i]$ for $1 \leqslant i \leqslant j$, $t[R_i] = t_2[R_i]$ for $j < i \leqslant k$. Let $S_1 = \bigcup_{i=1}^{j} R_i$, $S_2 = \bigcup_{i=j+1}^{k} R_i$. Then $t \in \pi_{S_1}(r) \times \pi_{S_2}(r)$. It follows that $r \notin \mathrm{FIXPT}\{S_1, S_2\}$, and hence (Fagin [14]) $r \notin \mathrm{SAT}(S_1 \cap S_2 \twoheadrightarrow S_1)$. However, it is shown in Mendelzon and Maier [18] that $S_1 \cap S_2 \twoheadrightarrow S_1$ is in M_ρ, contradicting our assumption that $r \in \mathrm{SAT}(M_\rho)$. ∎

LEMMA 2.[2] *Let D be a set of fd's and d a functional or multivalued dependency. If* $\mathrm{FIXPT}(\rho) \cap \mathrm{SAT}(D) \not\subseteq \mathrm{SAT}(d)$, then there exists a relation $r \in \mathrm{FIXPT}(\rho) \cap \mathrm{SAT}(D) - \mathrm{SAT}(d)$ such that r contains only two tuples.

Proof. Let M_ρ be the set of mvd's defined above. Since $\mathrm{FIXPT}(\rho) \subseteq \mathrm{SAT}(M_\rho)$, $\mathrm{SAT}(M_\rho \cup D) \not\subseteq \mathrm{SAT}(d)$. By Theorem 5, it follows that there is a relation $r = \{t_1, t_2\} \in \mathrm{SAT}(M_\rho \cup D) - \mathrm{SAT}(d)$. Since r contains only two tuples and satisfies M_ρ, it follows from Lemma 1 that $r \in \mathrm{FIXPT}(\rho)$. ∎

THEOREM 6. *In any* 4NF *database scheme,* J *holds iff* IC *holds.*

[2] This lemma was proved independently by Vardi [23].

Proof. By Theorem 2, J implies IC. We shall prove that for 4 NF schemes, IC implies lossless join and weak constraint preservation.

Suppose first that ρ is a 4NF database scheme satisfying IC but not weak constraint preservation. Then there exists a relation $s \in \text{FIXPT}(\rho)$ such that $\pi_{R_i}(s)$ satisfies $\pi_{R_i}(D)$ for all i, but $s \notin \text{SAT}(M, D)$.

Let $D' = \pi_\rho(D)$. The existence of relation s implies that there exists a functional or multivalued dependency $d \in M \cup D$ such that $\text{FIXPT}(\rho) \cap \text{SAT}(D') \nsubseteq \text{SAT}(d)$. By Lemma 2, there must exist a relation $r = \{t_1, t_2\} \in \text{FIXPT}(\rho) \cap \text{SAT}(D') - \text{SAT}(d)$.

Let $\sigma = \pi_\rho(r)$. Note that σ satisfies all the projected fd's D'. Since ρ is a 4NF schee, σ must also satisfy the projected mvd's $\pi_\rho(M)$.

We claim that there is no relation $r' \in \text{SAT}(M \cup D)$ such that $\pi_\rho(r') = \sigma$, violating IC2. In proof, suppose there existed such a relation. Since r is in $\text{FIXPT}(\rho)$, $\chi\, \sigma = r$. Thus it must be the case that $r' \subseteq r$. Hence r' has no more than two tuples. On the other hand, if r' contains only one tuple, then so does $\chi\, \sigma$, which equals r. But this contradicts the fact that r violates dependency d, since a one-tuple relation satisfies every fd and mvd. It follows that $r' = r$, so r' cannot satisfy M and D.

Now suppose that lossless join fails. Let $r \in \text{SAT}(D \cup M)$ be such that $\sigma = \pi_\rho(r)$, $\chi\, \sigma = r' \neq r$. If $r' \in \text{SAT}(D \cup M)$, then IC1 is violated. If $r' \notin \text{SAT}(D \cup M)$, then weak constraint preservation fails, since $r' \in \text{FIXPT}(\rho)$ and r' satisfies all the projected dependencies. By the first part of the proof, since weak constraint preservation fails, IC fails. ∎

To test whether J holds in a 4NF decomposition, we need to determine whether $D \cup M$ guarantee a lossless join and weak constraint preservation for ρ. The lossless join condition can be tested by the method of Aho *et al.* [1]. Weak constraint preservation, for a 4NF scheme, reduces to the question of whether $\text{FIXPT}(\rho) \cap \text{SAT}(\pi_\rho(D)) \subseteq \text{SAT}(D \cup M)$. In the terminology of Rissanen [20] (see also Beeri and Vardi [9], Sciore [22]), this is the same as testing whether certain fd's and mvd's are implied by a "join dependency" and a set of fd's. An exponential time algorithm for this problem was given by Maier *et al.* [16], and a polynomial time algorithm by Maier *et al.* [17] and Vardi [23]. Note that the best known test for the lossless join in the presence of mvd's is still exponential in time and space (Aho *et al.* [1]). Also, the computation of the D_i's seems to require exponential time. However, if we assume that the database scheme is the result of the decomposition process of Fagin [14], and covers for the projected dependencies are given, the test can be done in polynomial time by the algorithm mentioned above, since this process always yields lossless decompositions.

III. Open Questions

There are a variety of problems suggested by the foregoing results that appear to be quite difficult. Among these are:

1. Is (J) different from (IC) in general? We now know they are the same if the legal relations are determined by functional dependencies, or if ρ is a 4NF scheme.

Also, it is easy to show that there exist contrived L's for which they are different. What if L is determined by a set of multivalued dependencies, or by another type of dependency with which we are familiar?

2. Is there an effective test for (IC) for any case but the special cases noted in (1) above?

3. How do we test (AC) if the legal relations are determined by multivalued dependencies? This is a special case of the more general problem of deciding when embedded and full multivalued dependencies determine a full multivalued dependency, where a *full* dependency is one that applies to the universal set of attributes.

REFERENCES

1. A. V. AHO, C. BEERI, AND D. ULLMAN, The theory of joins in relational databases, *TODS* **4**, No. 3 (1979), 297–314.
2. W. W. ARMSTRONG, Dependency structures of database relationships, *in* "*Proceedings, 1974 IFIP Congress*," pp. 580–583, North-Holland, Amsterdam, 1974.
3. A. K. ARORA, AND C.R. CARLSON, The information preserving properties of certain relational database transformations, *in* "*Proceedings, 4th Intl. Conf. on Very Large Databases*, 1978," pp. 352–359.
4. C. BEERI, P. A. BERNSTEIN AND N. GOODMAN, A sophisticate's introduction to database normalization theory, *in* "*Proceedings, 4th Intl. Conf. on Very Large Databases*, 1978," pp. 113–124.
5. C. BEERI, R. FAGIN, AND J. H. HOWARD, A complete axiomatization for functional and multivalued dependencies, *in* "*Proceedings, ACM–SIGMOD Intl. Conf. on Management of Data*, 1977," pp. 47–61.
6. C. BEERI AND P. HONEYMAN, Preserving functional dependencies, *SIAM J. Comp.*, in press.
7. C. BEERI, A. O. MENDELZON, Y. SAGIV, AND J. D. ULLMAN, Equivalence of relational database schemes, *in* "*Proceedings, 11th ACM Symp. on Theory of Computing* 1979," pp. 319–329.
8. C. BEERI AND J. RISSANEN, "Faithful Representation of Relational Database Schemes,"IBM Research Report RJ2722, 1980.
9. C. BEERI AND M. Y. VARDI, On the properties of join dependencies, *in* "*Advances in Data Base Theory*," *Vol. 1*, Plenum, New York, in press.
10. P. A. BERNSTEIN, Synthesizing third normal form relations from functional dependencies, *TODS* **1**:4 (1976), 277–298.
11. E. F. CODD, A relational model for large shared data banks, *Comm. ACM* **13**:6, (1970), 377–387.
12. E. F. CODD, Further normalization of the data base relational model, *in* "*Data Base Systems*" (R. Rustin, Ed.), Prentice–Hall, Englewood Cliffs, N. J., 1972.
13. C. DELOBEL, Normalization and hierarchical dependencies in the relational data model, *TODS* **3**:3 (1978), 201–222.
14. R. FAGIN, Multivalued dependencies and a new normal form for relational databases, *TODS* **2**:3 (1977), 262–278.
15. R. FAGIN, Normal forms and relational database operators, *in* "*Proceedings, ACM–SIGMOD Intl. Conf. on Management of Data*, 1979," pp. 153–160.
16. D. MAIER, A. O. MENDELZON, AND Y. SAGIV, Testing implications of data dependencies, *TODS* **4**:4 (1979), 455–469.
17. D. MAIER, Y. SAGIV, AND M. YANNAKAKIS, On the complexity of testing implication of functional and join dependencies, unpublished manuscript, 1979.
18. A. O. MENDELZON AND D. MAIER, Generalized mutual dependencies and the decomposition of

database relations, *in* "*Proceedings, 5th Intl. Conf. on Very Large Databases*, 1979," pp. 75–82.

19. J. RISSANEN, Independent components of relations, *TODS* **2**:4 (1977), 317–325.
20. J. RISSANEN, Theory of relations for databases—A tutorial survey, "*Proceedings, Symp. on Mathematical Foundations of Computer Science*, Zakopane, Poland," Springer-Verlag, Berlin/New York, 1978.
21. Y. SAGIV AND R. FAGIN, "An Equivalence between Relational Database Dependencies and a Subclass of Propositional Logic," IBM Research Report RJ2500, 1979.
22. E. SCIORE, "A Complete Axiomatization of Full Join Dependencies," Princeton University, Dept. of EECS TR No. 279, July 1979.
23. J. D. ULLMAN, "Principles of Database Systems," Computer Science Press, Potomac, Md., 1980.
24. M. Y. VARDI, Inferring multivalued dependency from functional and join dependencies, Research Report, Dept. of Applied Math., Weizmann Inst. of Science, March 1980.
25. C. ZANIOLO, "Analysis and Design of Relational Schemata for Database Systems," Doctoral dissertation, UCLA, 1976.