Inclusion dependencies and their interaction with functional dependencies

(Extended abstract)

by

Marco A. Casanova*
Pontificia Universidade Catolica do Rio de Janeiro
22453 Rio de Janeiro RJ
Brazil

Ronald Fagin**
IBM Research Laboratory
San Jose, California 95193

Christos H. Papadimitriou
Lab. for Computer Science, M.I.T.
Cambridge, Massachusetts 02139

ABSTRACT: Inclusion dependencies, or INDs (which can say, for example, that every manager is an employee) are studied, including their interaction with functional dependencies, or FDs. A simple complete axiomatization for INDs is presented, and the decision problem for INDs is shown to be PSPACE-complete. (The decision problem for INDs is the problem of determining whether or not Σ logically implies σ, given a set Σ of INDs and a single IND σ). It is shown that finite implication (implication over databases with a finite number of tuples) is the same as unrestricted implications for INDs, although finite implication and unrestricted implication are distinct for FDs and INDs taken together. It is shown that, although there are simple complete axiomatizations for FDs alone and for INDs alone, there is no complete axiomatization for FDs and INDs taken together, in which every rule is k-ary for some fixed k (and in particular, there is no finite complete axiomatization.) This is true whether we consider finite implication or unrestricted implication, and is true even if no relation scheme has more than three attributes. The nonexistence of a k-ary complete axiomatization for FDs and INDs taken together is proven by giving a condition which is necessary and sufficient in general for the existence of a k-ary complete axiomatization.

Key words and phrases: inclusion dependency, functional dependency, relational database, complete axiomatization, PSPACE-complete.

CR categories: 3.73, 4.33, 5.21, 5.27.

*This research was supported by FINEP and by CNPq Grant 402090/80.

**This research was conducted while this author was a Visiting Research Fellow at Pontificia Universidade Catolica do Rio de Janeiro and was supported in part by a grant from IBM Brazil.

1. INTRODUCTION

Functional dependencies, or FDs [Co1] are certainly the most important and widely-studied integrity constraints for relational databases. Another important integrity constraint is the inclusion dependency, or IND [Fa3]. As an example, an inclusion dependency can say that every MANAGER entry of the R relation appears as an EMPLOYEE entry of the S relation. More generally, an inclusion dependency can say that the projection onto a given m columns of the R relation are a subset of the projection onto a given m columns of the S relation. Hence, INDs are valuable for database design, since they permit us to selectively define what data must be duplicated in what relations.

INDs, together with FDs, form the basis of the structural model of Wiederhold and El-Masri [WM]. They also appear when an entity-relationship schema is mapped to the relational model ([Ch], [Kl]). Yet in another perspective, INDs can be viewed as a relaxation of the controversial universal relation assumption ([BG], [Ke]), which requires that all relations in a database be projections of a single (universal) relation. Inclusion dependencies are commonly known in Artificial Intelligence applications as *ISA* relationships (cf. Beeri and Korth [BK]).

We note that INDs differ from other commonly studied database dependencies in two important respects. First, INDs may be interrelational, whereas the others deal with a single relation at a time. Second, INDs are not typed [Fa4]. INDs are special cases of extended embedded implicational dependencies [Fa4], for which the existence of "Armstrong-like databases" have been proven. For details, see [Fa4].

Although INDs have been utilized extensively for databases ([BK], [Ch], [Co2], [Fa3], [Kl], [SS], [WM], [Za]), there has been very little analysis of their properties, with only a few recent exceptions ([FV], [JK], [Li]). The purpose of this paper is to help remedy this neglect.

In Section 2, we present basic definitions. In Section 3, we present a simple complete axiomatization for INDs. Since the axiomatization is complete even if we restrict our attention to finite databases, it follows that finite implication (implication over finite databases) is the same as unrestricted implication for INDs. However, we give a simple example that shows that finite implication is distinct from unrestricted implication, for FDs and INDs taken together. A similar result has been proven for template dependencies [FMUY]. We also show that the decision problem for INDs is PSPACE-complete. In Section 4, we give a general necessary and sufficient condition for the existence of a k-ary complete axiomatization. We show how the result can be used to explain Sagiv and Walecka's [SW] result on the nonexistence of a k-ary complete axiomatization for embedded multivalued dependencies, for arbitrary k. In Section 5, we show that for no k is there a k-ary complete axiomatization for finite implication of FDs and INDs. In fact, our proof shows that this result holds, even if no relation scheme has more than two attributes. The same result holds (where no relation scheme has more than three attributes) for unrestricted implication, but the proof is more complex, and is omitted. (However, the proof appears in the full paper [CFP].)

2. DEFINITIONS

A *relation scheme* is an object R[U], where R is the *name* of the relation scheme and where U is a finite sequence $<A_1,...,A_m>$ of *attributes*. We usually write a sequence, such as $<A_1,...,A_m>$, as simply $A_1,...,A_m$. For example, we shall write simply $R[A_1,...,A_m]$ for $R[<A_1,...,A_m>]$. A *tuple* t over $U=<A_1,...,A_m>$ is a sequence $<a_1,...,a_m>$. A *relation* (over R[U], or simply over R) is a set of tuples over U. Note that our definition, which is convenient for use in this paper, is distinct from other definitions ([ABU], [Ar]) in which a tuple is a mapping, not a sequence. If $X = <A_{i_1},...,A_{i_k}>$, where $i_1...,i_k$ are distinct members of {1,...,n}, and if t is as above, then t[X] is $<a_{i_1},...,a_{i_k}>$. If r is a set of tuples over U, then $r[X] = \{t[X]: t \in r\}$.

A *database scheme* $D=\{R_1[U_1],...,R_n[U_n]\}$ is a set of relation schemes. A *database* over D is a mapping that associates each relation scheme $R_i[U_i]$ with a relation r_i over R_i. When it can cause no confusion, we may refer to $r_1,...,r_n$ as the database.

A relation is *finite* if it has a finite set of tuples; a database $r_1,...,r_n$ is finite if each r_i is finite. If C is a set, then $|C|$ is the cardinality of C; if $X=<a_1...,a_k>$ is a sequence, then $|X|=k$.

If $R[A_1,...,A_m]$ is a relation scheme, and if X is a sequence of distinct members of $A_1,...A_m$, as is Y, then we call the object R:X→Y a *functional dependency*, or FD. Although X and Y are usually taken to be sets, rather than sequences, it is necessary for us to use sequences, so that we can interrelate FDs and inclusion dependencies, defined soon. If r is a relation over R, then r *obeys* or *satisfies* the FD R:X→Y if, whenever t_1 and t_2 are tuples of r such that $t_1[X]=t_2[X]$, then $t_1[Y]=t_2[Y]$. We also say then that the FD R:X→Y *holds* for r, or is *true* about r. If the FD does not hold for r, then we say that r *violates* the FD. A similar comment applies for other dependencies, defined later.

If $R_i[A_1,...,A_m]$ and $R_j[B_1,...B_p]$ are relation schemes (not necessarily distinct), if X is a sequence of k distinct members of $A_1,...,A_m$, and if Y is a sequence of k distinct members of $B_1,...,B_p$, then we call the object $R_i[X] \subseteq R_j[Y]$ an *inclusion dependency*, or IND. (Inclusion dependencies should not be confused with the *subset dependencies* of Sagiv and Walecka [SW], which are quite different). If $r_1,...,r_n$ is a database d over $D=\{R_1[U_1],...,R_n[U_n]\}$, then d obeys the IND $R_i[X] \subseteq R_j[Y]$ if $r_i[X] \subseteq r_j[Y]$.

FDs and INDs are examples of *dependencies*, or sentences about databases [Fa4]. Let Σ be a set of dependencies over D, and let σ be a single dependency over D. When we say that Σ *logically implies* σ (in the context D), or that σ is a *logical consequence* of Σ, we mean that whenever d is a database over D that obeys every dependency in Σ, then d obeys σ. That is, there is no "counterexample database" d such that d obeys every sentence in Σ, but such that d does not obey σ. We then write $\Sigma \models_D \sigma$, or, if D is understood, simply $\Sigma \models \sigma$. If Σ and T are each sets of dependencies, then by $\Sigma \models T$, we mean that $\Sigma \models \tau$ for each $\tau \in T$. We write $\Sigma \models_{fin} \sigma$ to mean that whenever d is a *finite* database that obeys Σ, then also d obeys σ. Clearly, if $\Sigma \models \sigma$, then $\Sigma \models_{fin} \sigma$, but, as we shall see, the converse is false. Finally, we write $\Sigma \not\models \sigma$ to mean that is false that $\Sigma \models \sigma$.

3. RESULTS ON INDS ALONE

We now exhibit a complete axiomatization for INDs. However, for reasons of brevity, we omit the completeness proof (which appears in the full paper [CFP].) We note that Lin [Li] presents a set of inference rules for INDs, and conjectures their completeness. Since his rules imply ours below, his axiomatization is indeed complete.

IND1 (reflexivity): $R[X] \subseteq R[X]$, if X is a sequence of distinct attributes of R.

IND2 (projection and permutation): if $R[A_1,...,A_m] \subseteq S[B_1,...,B_m]$, then $R[A_{i_1},...,A_{i_k}] \subseteq S[B_{i_1},...,B_{i_k}]$, for each sequence $i_1,...,i_k$ of distinct integers from {1,...,m}

IND3 (transitivity): if $R[X] \subseteq S[Y]$ and $S[Y] \subseteq T[Z]$, then $R[X] \subseteq T[Z]$.

Our proof of completeness shows that the same axiomatization is complete for INDs over finite databases. Therefore, finite and unrestricted implication ($\models_{fin}$ and $\models$) are the same for INDs. However, finite and unrestricted implication are distinct for FDs and INDs taken together. Thus, let Σ be $\{R:A \to B, R[A] \subseteq R[B]\}$, and let σ be $R[B] \subseteq R[A]$. It is easy to verify [CFP] that $\Sigma \models_{fin} \sigma$; however, $\Sigma \not\models \sigma$ (just consider the relation $r = \{(i+1,i): i \geq 0\}$.)

Our proof of completeness also leads to a decision procedure for the decision problem for INDs (that is, for determining if $\Sigma \models \sigma$, where Σ is a set of INDs, and where σ is a single IND). Say σ is the IND $R_a[A_1,...,A_m] \subseteq R_b[B_1,...,B_m]$.

(1) Initialize set Z by letting it contain the single expression $R_a[A_1,...,A_m]$.
(2) If Z contains an expression S[X], and if an IND $S[X] \subseteq T[Y]$ can be obtained from a member of Σ by IND2 (projection and permutation), then add T[Y] to the set Z, unless it is already in Z.
(3) Apply step 2 repeatedly, until it is no longer possible to add expression to Z by using step 2.
(4) $\Sigma \models \sigma$ if and only if $R_b[B_1,...,B_m]$ is in the resulting set Z.

This decision procedure is quite similar to a decision procedure for FDs [BB], where Z is a set of attributes, and where attributes are added to Z on the basis of FDs. However, there is a major difference. The FD decision procedure can be implemented (with the appropriate data structure) to run in linear time. Unfortunately, however, in the case of INDs, the set Z can grow to exponential size. An example can be given [CFP] that is based on the fact that the (exponential number of) permutations over n letters are all generated by the (polynomial number of) transpositions.

We shall soon show that the decision problem for INDs is PSPACE-complete. Hence, there is no polynomial-time algorithm for this problem (unless P=PSPACE) [GJ]. However, it is easy to see that in certain special cases, our decision procedure can be implemented to run in polynomial time. For example, there is a polynomial time algorithm if we restrict our attention to INDs that are at most k-ary for some fixed k (that is, INDs $R[A_1,...,A_r] \subseteq S[B_1,...,B_r]$, where $r \leq k$). As another example, there is a polynomial-time algorithm if we restrict our attention to INDs of the form $R[X] \subseteq S[X]$. As an example of this latter type of IND, it is possible to say that every manager is an employee of the department that he manages by the IND MGR[NAME,DEPT]⊆EMP[NAME,DEPT], where, say (Hilbert, Math) is a tuple of the MGR relation if Hilbert manages the Math Department, etc.

We close this section by showing that the decision problem for INDs is PSPACE-complete.

Theorem 3.1: The decision problem for INDs is PSPACE-complete.

Proof: We first show that the decision problem for INDs is in PSPACE. We now describe a nondeterministic polynomial-space algorithm for deciding if $\Sigma \models \sigma$, where Σ is a set of INDs and where σ is a single IND. Assume that σ is $R_a[A_1,...,A_m] \subseteq R_b[B_1,...,B_m]$. Let $S_1[X_1]$ be $R_a[A_1,...,A_m]$. Given $S_i[X_i]$, the nondeterministic algorithm simply "guesses" an IND τ in Σ to apply IND2 (projection and permutation) to, in order to obtain an IND $S_i[X_i] \subseteq S_{i+1}[X_{i+1}]$, and then overwrites $S_i[X_i]$ with $S_{i+1}[X_{i+1}]$. The algorithm halts and rejects if the IND τ that it guesses cannot yield an IND with left-hand side $S_i[X_i]$ when IND2 is applied. The algorithm accepts if it ever prints $R_b[B_1,...,B_m]$ as an $S_i[X_i]$. Since the nondeterministic algorithm operates in linear space, it follows by Savitch's Theorem [Sa] that there is a deterministic quadratic space algorithm. Thus, the decision problem for INDs is in PSPACE.

We now show that the decision problem for INDs is PSPACE-complete. To show this, we shall reduce the following known PSPACE-complete problem to ours:

LINEAR BOUNDED AUTOMATON ACCEPTANCE [GJ]
Instance: A nondeterministic Turing machine M and an input $x \in \Gamma^*$.
Question: Is there a halting computation of M on input x using no more than $|x|$ tape cells?

Given an instance M;x of LINEAR BOUNDED AUTOMATON ACCEPTANCE, we shall construct a set Σ of INDs and a single IND σ such that $\Sigma \models \sigma$ if and only if M halts on x in space $|x|$. $M=(K,\Gamma,\Delta,s,h)$ is a nondeterministic 1-tape Turing machine with state set K, alphabet Γ, start state $s \in K$, halt state $h \in K$, and transition relation Δ (see [LP] for Turing machine notation). A configuration of such a machine on input x, with $|x|=n$, shall be denoted by a string in $\Gamma^*K\Gamma^+$ of length n+1. The n symbols in Γ are the tape contents, and the symbol in K denotes the current state and the head position (it is placed immediately to the left of the symbol scanned). The initial configuration is sx, and the final configuration hB^n, where $B \in \Gamma$ is the blank.

Our INDs are defined on a single relation scheme R with set of attributes $U=(K \cup \Gamma) \times \{1,2,...,n+1\}$. The intuition is that the attribute $(r,j) \in U$ corresponds to the jth symbol in a configuration being r (this will become clearer later.) The IND σ is $R[(s,1),(x_1,2),...,(x_n,n+1)] \subseteq R[(h,1),(B,2),...,(B,n+1)]$. The INDs in Σ encode the legal moves of M. These moves can be thought of as rewriting rules of the form $abc \to a'b'c'$ where $a,b,c,a',b',c' \in K \cup \Gamma$, applied on configurations. For each such move m, and each $j \in \{1,2,...,n-1\}$, we have in Σ the following IND: $R[P_j,(a,j),(b,j+1),(c,j+2)] \subseteq R[P_j,(a',j),(b',j+1),(c',j+2)]$, where P_j is one arbitrarily selected ordering of the attributes in $\Gamma \times \{1,2,...,j-1,j+3,...,n+1\}$. This completes the construction. We can show [CFP] that $\Sigma \models \sigma$ if and only if M accepts x in space n. □

4. CHARACTERIZATION OF THE EXISTENCE OF A k-ARY COMPLETE AXIOMATIZATION

In this section, we present necessary and sufficient conditions for the existence of a k-ary complete axiomatization for a set S of sentences over a database scheme D. In Section 5, we use our characterization to show that for each k, there is a database scheme D such that the set of FDs and INDs over D have no k-ary complete axiomatization. In this section, we use our characterization to explain Sagiv and Walecka's similar result for embedded multivalued dependencies.

Let $D=\{R_1,...,R_n\}$ be a database scheme, that is, each R_i has associated with it a set of attributes ($1 \le i \le n$). Let $\mathscr{S}$ be a set of dependencies, that is, sentences over $R_1,...,R_n$. In our case of primary interest, $\mathscr{S}$ is the set of all FDs and INDs over $R_1,...,R_n$. By a *rule* (over $\mathscr{S}$), we mean a statement of the form "if T then τ", where T is a finite set of sentences in $\mathscr{S}$ (each called an *antecedent* of the rule) and where τ is a single sentence in $\mathscr{S}$ (called the *consequence* of the rule). If T contains exactly k distinct members, then we call this rule *k-ary*. A 0-ary rule (one for which $T=\emptyset$) is sometimes called an axiom. The rule "if T then τ" is *sound* if $T \models_D \tau$; that is, if every database over D that obeys T also obeys τ. A set $\mathscr{R}$ of rules is said to be sound if every member of $\mathscr{R}$ is sound.

Let $\mathscr{R}$ be a set of rules over $\mathscr{S}$. Let Σ be a set of sentences in $\mathscr{S}$, and let σ be a single sentence in $\mathscr{S}$. A *proof* of σ from Σ via $\mathscr{R}$ is a finite sequence $<\tau_1,...,\tau_m>$ of sentences in $\mathscr{S}$ where τ_m, the last sentence in the sequence, is σ, and where for each i ($1 \le i \le m$), either (a) $\tau_i \in \Sigma$, or (b) there is a subset T of $\{\tau_1,...,\tau_{i-1}\}$ such that "if T then τ_i" is a rule in $\mathscr{R}$. If there is a proof of σ from Σ via $\mathscr{R}$, then we write $\Sigma \vdash_{\mathscr{R}} \sigma$ (or, if $\mathscr{R}$ is understood, simply $\Sigma \vdash \sigma$). It is easy to see that a set $\mathscr{R}$ of rules is sound under our definition if and only if whenever $\Sigma \vdash_{\mathscr{R}} \sigma$, then $\Sigma \models_D \sigma$.

A set $\mathscr{R}$ of rules over $\mathscr{S}$ and D is *complete* if whenever $\Sigma \subseteq \mathscr{S}$ and $\sigma \in \mathscr{S}$, then $\Sigma \models_D \sigma$ if and only if $\Sigma \vdash_{\mathscr{R}} \sigma$. We note that some authors weaken this definition by requiring only that if $\Sigma \models_D \sigma$, then $\Sigma \vdash_{\mathscr{R}} \sigma$. Thus, for these authors, completeness does not imply soundness, whereas for us, it does (that is, for us, every complete set of rules is sound.) We sometimes call a complete set of rules a *complete axiomatization*. A set $\mathscr{R}$ of rules is k-ary if each rule ρ in $\mathscr{R}$ is at most k-ary; in other words, if ρ is r-ary, then $r \le k$.

As an example, consider our complete axiomatization for INDs in Section 3. For a given database scheme D, each of IND1, IND2 and IND3 is really a rule scheme that represents a set of rules. For example, IND1 (reflexivity), $R[X] \subseteq R[X]$, represents a set of 0-ary rules, one for every relation scheme R in D and every sequence X of distinct attributes of R. Similarly, IND2 (projection and permutation) represents a set of 1-ary rules, and IND3 (transitivity) represents a set of 2-ary rules. For a given database scheme, the set of all of these rules (rules represented by one of IND1, IND2, IND3) is a 2-ary complete axiomatization.

We shall give a necessary and sufficient condition for the existence of a k-ary complete axiomatization for a set $\mathscr{S}$ of sentences over a database scheme D. In Section 5, we shall use this characterization to show that for each k, there is a database scheme such that if $\mathscr{S}$ is the set of all FDs and INDs over the scheme, then there is no k-ary complete axiomatization for $\mathscr{S}$. But what does this mean? Let D be a given database scheme, and let $\mathscr{S}$ be the set of all FDs and INDs over D. There are only a finite number of distinct FDs and INDs over D; let this number be k. Then there is certainly a k-ary complete axiomatization: we simply take all rules "if T then τ", where T is a set of FDs and INDs over D, where τ is a single FD or IND over D, and where $T \models_D \tau$. What our results say is that there is no *single* k that can work for every database scheme D (although, as we just saw, every database scheme D has a k-ary complete axiomatization for FDs and INDs for *some* k).

By a "complete axiomatization for FDs and INDs", one *might* mean a "uniform" complete axiomatization, good for every scheme D. For example, our complete axiomatization for INDs in Section 3 is in some sense "uniform", as are Armstrong's [Ar] complete axiomatization for FDs, Beeri, Fagin and Howard's [BFH] complete axiomatization for multivalued dependencies, Sadri and Ullman's [SU] and Beeri and Vardi's [BV] complete axiomatization for template dependencies, and Yannakakis and Papadimitriou's [YP] complete axiomatization for algebraic dependencies (i.e., embedded implicational dependencies [Fa4]). Whatever one means by a "uniform" k-ary complete axiomatization, this must at least imply that for every scheme, there is a k-ary complete axiomatization. Therefore, our result on the nonexistence of a k-ary complete axiomatization for FDs and INDs over certain schemes certainly implies the nonexistence of a "uniform" k-ary complete axiomatization for FDs and INDs.

Before we present the main result of this section, we need some more definitions. Let D be a database scheme, let $\mathcal{S}$ be a set of sentences about D, and let Γ be a subset of $\mathcal{S}$. We say that Γ is *closed under implication (with respect to D and $\mathcal{S}$)* if whenever (a) $\Sigma \subseteq \Gamma$, (b) $\sigma \in \mathcal{S}$, and (c) $\Sigma \models_D \sigma$, then $\sigma \in \Gamma$. If D and $\mathcal{S}$ are understood, then we simply say that Γ is *closed under implication.* If $k \geq 0$ is an integer, then we say that Γ is *closed under k-ary implication (with respect to D and $\mathcal{S}$)* if whenever (a), (b), and (c) above hold, and also (d) $|\Sigma| \leq k$, then $\sigma \in \Gamma$. Again, if D and $\mathcal{S}$ are understood, then we simply say that Γ is *closed under k-ary implication.*

Theorem 4.1: Let D be a database scheme, let $\mathcal{S}$ be a set of sentences about D, and let $k \geq 0$ be an integer. There is a k-ary complete axiomatization for sentences in $\mathcal{S}$ if and only if whenever $\Gamma \subseteq \mathcal{S}$ is closed under k-ary implication, then Γ is closed under implication.

Proof: See [CFP]. □

Corollary 4.2: Let D be a database scheme, let $\mathcal{S}$ be a set of sentences about D, and let $k \geq 0$ be a constant. Assume that $\Sigma \subseteq \mathcal{S}$, that $\sigma \in \mathcal{S}$, and that

(i) $\Sigma \models \sigma$,
(ii) if $\tau \in \Sigma$ then it is false that $\tau \models \sigma$, and
(iii) if Δ is a set of at most k members of Σ, if $\tau \in \mathcal{S}$ and if $\Delta \models \tau$, then there is some $\delta \in \Delta$ such that $\delta \models \tau$.

Then there is no k-ary complete axiomatization for sentences in $\mathcal{S}$.

Proof: Let $\Gamma = \{\tau \in \mathcal{S}$: there is $\tau' \in \Sigma$ such that $\tau' \models \tau\}$. Since $\Sigma \subseteq \Gamma$ but $\sigma \notin \Gamma$, it follows that Γ is not closed under implication. We now show that Γ is closed under k-ary implication. Assume that T is a set of at most k members of Γ, that $\tau \in \mathcal{S}$ and that $T \models \tau$. We must show that $\tau \in \Gamma$. For each α in T, find $\alpha' \in \Sigma$ such that $\alpha' \models \alpha$ (we know that α' exists by definition of Γ). Let $\Delta = \{\alpha' : \alpha \in T\}$. Clearly $\Delta \models \tau$, since $\Delta \models T$ and $T \models \tau$. By (iii), it follows that $\tau \in \Gamma$. Hence, Γ is closed under k-ary implication. Since Γ is not closed under implication, it follows from Theorem 4.1 that there is no k-ary complete axiomatization for sentences in $\mathcal{S}$. This was to be shown. □

Corollary 4.2 can be used to explain Sagiv and Walicka's [SW] result on the nonexistence of a k-ary complete axiomatization axiomatization for embedded multivalued dependencies (EMVDs), for each k (for a definition of EMVDs, see [Fa1].) This follows because, for each $k > 0$, they exhibit a relation scheme R, a set Σ of EMVDs over R and a single EMVD σ over R that obey the conditions of Corollary 4.2.

5. NONEXISTENCE OF A k-ARY COMPLETE AXIOMATIZATION FOR FDs AND INDs.

We can use Theorem 4.1 to prove the following two results. The first theorem deals with finite implication and the second with unrestricted implication.

Theorem 5.1: For no k is there a k-ary complete axiomatization for finite implication of FDs and INDs.

Theorem 5.2: For no k is there a k-ary complete axiomatization for FDs and INDs.

Note: By Theorem 5.1, we mean that for each k, there is a database scheme D such that there is no k-ary complete axiomatization for finite implication of FDs and INDs over D. A similar comment applies to Theorem 5.2.

Since the proof (which appears in [CFP]) of Theorem 5.2 is substantially more complicated than the proof of Theorem 5.1, we give here only the proof of Theorem 5.1. The proof of Theorem 5.1 depends in part on a counting argument, which applies only in the case of a *finite* database.

Proof of Theorem 5.1: Let k be a fixed natural number. Let $R_i[AB]$ $(0 \leq i \leq k)$ be a set of relation schemes. Define (where, henceforth, addition is module k):

(1) $\Sigma = \{R_i{:}A \rightarrow B,\ R_i[A] \subseteq R_{i+1}[B]{:}\ 0 \leq i \leq k\}$, and
(2) $\sigma = R_0[B] \subseteq R_k[A]$.

Let Γ be the union of Σ with the set of all trivial FDs and INDs (those that are tautologies). By Theorem 4.1 (where finite implication plays the role of implication, that is, where $\models_{fin}$ plays the role of $\models$), we need only show that Γ is closed under k-ary finite implication but not under finite implication.

We first show that Γ is not closed under finite implication. To do this, we need only show that $\Sigma \models_{fin} \sigma$, since it is immediate that $\Sigma \subseteq \Gamma$ and that $\sigma \notin \Gamma$. Let $\mathcal{I} = \{r_0,\ldots,r_k\}$ be a finite database satisfying Σ. Since $\mathcal{I}$ satisfies $R_i[A] \subseteq R_{i+1}[B]$, it follows that $|r_i[A]| \leq |r_{i+1}[B]|$, for $0 \leq i \leq k$. Since $\mathcal{I}$ satisfies $R_i{:}A \rightarrow B$, it follows that $|r_i[B]| \leq |r_i[A]|$ holds, for $0 \leq i \leq k$. Putting these inequalities together, we obtain $|r_0[A]| \leq |r_1[B]| \leq |r_1[A]| \leq \ldots \leq |r_k[A]| \leq |r_0[B]| \leq |r_0[A]|$. Hence, $|r_k[A]| = |r_0[B]|$. But since $\mathcal{I}$ satisfies the IND $R_k[A] \subseteq R_0[B]$ and since $\mathcal{I}$ is finite, we then have $r_k[A] = r_0[B]$. Hence, $r_0[B] \subseteq r_k[A]$, and so $\mathcal{I}$ obeys σ.

We conclude the proof by showing that Γ is closed under k-ary finite implication. That is, we shall show that if T is a set of at most k members of Γ, if τ is an FD or IND, and if $T \models_{fin} \tau$, then $\tau \in \Gamma$.

Since Σ contains k+1 INDs, we know that T does not contain some IND δ of Σ. Since Σ is symmetric with respect to INDs, we may assume without loss of generality that δ is the IND $R_k[A] \subseteq R_0[B]$. We construct a database $\mathcal{I} = \{r_0,\ldots,r_k\}$ as follows:

$r_0 = \{((0,0),\ (0,k+1)),\ ((1,0),\ (1,k+1)),\ ((2,0),\ (1,k+1))\}$

$r_i = \{((0,i),\ (0,i-1)),\ ((1,i),\ (1,i-1)),\ \ldots,\ ((2i+1,i),\ (2i+1,i-1)),\ ((2i+2,i),\ (2i+1,i-1))\}$, for $1 \leq i \leq k$.

Figure 1 exhibits $\mathcal{I}$ for k=3.

It is straightforward but tedious [CFP] to verify that the database $\mathcal{I}$ obeys precisely the FDs and INDs in $\Gamma - \delta$ (in the terminology of Fagin [Fa4], the database $\mathcal{I}$ is a *finite Armstrong database* for $\Gamma - \delta$.) Since $T \subseteq \Gamma - \delta$, it follows that $\mathcal{I}$ obeys T. Because $T \models_{fin} \tau$, we also know that $\mathcal{I}$ obeys τ. Since $\mathcal{I}$

r_0:

A	B
(0,0)	(0,4)
(1,0)	(1,4)
(2,0)	(1,4)

r_1:

A	B
(0,1)	(0,0)
(1,1)	(1,0)
(2,1)	(2,0)
(3,1)	(3,0)
(4,1)	(3,0)

r_2:

A	B
(0,2)	(0,1)
(1,2)	(1,1)
(2,2)	(2,1)
(3,2)	(3,1)
(4,2)	(4,1)
(5,2)	(5,1)
(6,2)	(5,1)

r_3:

A	B
(0,3)	(0,2)
(1,3)	(1,2)
(2,3)	(2,2)
(3,3)	(3,2)
(4,3)	(4,2)
(5,3)	(5,2)
(6,3)	(6,2)
(7,3)	(7,2)
(8,3)	(7,2)

Figure 1

obeys precisely Γ-δ, it follows that $\tau \in \Gamma$-δ. Hence, $\tau \in \Gamma$, which was to be shown. □

Let $\mathscr{P}$ be a class of dependencies such that the database $\mathscr{I}$ constructed in the proof of the previous theorem violates every nontrivial member of $\mathscr{P}$. By letting Γ be as before, along with the set of all trivial members of $\mathscr{P}$, our proof shows that there is no k-ary complete axiomatization for finite implication of FDs, INDs, and dependencies in $\mathscr{P}$. For example, if we let $\mathscr{P}$ be the class of multivalued dependencies, or MVDs [Fa1], then we know that there is no k-ary complete axiomatization for finite implication of FDs, INDs, and MVDs, since $\mathscr{I}$ obeys no nontrivial MVDs. Further, our proof shows that Theorem 5.1 holds, even if no relation scheme has more than two attributes.

To conclude, we make one more remark about the nonexistence of a k-ary complete axiomatization. Let us denote by Σ_k the set Σ of FDs and INDs in the proof of Theorem 5.1, and similarly let σ_k be σ of Theorem 5.1. Then the rule "if Σ_k then σ_k" has more than k antecedents, none of which can be eliminated and still leave a sound rule. However, the reader is cautioned against believing that this property, in and of itself, shows the nonexistence of a k-ary complete axiomatization. For, let T_k be the set $\{A_1 \rightarrow A_2, A_2 \rightarrow A_3, ..., A_{k+1} \rightarrow A_{k+2}\}$ of FDs, and let τ_k be the FD $A_1 \rightarrow A_{k+2}$. Then the rule "if T_k then τ_k" has this same property, yet FDs have a 2-ary complete axiomatization ([Ar], [Fa2]).

8. CONCLUSIONS AND DIRECTIONS FOR FUTURE RESEARCH

We have shown that inclusion dependencies have a simple complete axiomatization, just as FDs do. However, when INDs and FDs are considered together, then for no k is there a k-ary complete axiomatization. This result was obtained with the help of a general necessary and sufficient condition for the existence of a k-ary complete axiomatization. This condition is itself of interest, since it might help analyze classes of dependencies that have not yet been completely axiomatized (such as join dependencies [ABU]).

We have also shown that the decision problem for INDs is PSPACE-complete. Thus, there is no polynomial-time decision procedure (unless P=PSPACE). Although the decision problem for FDs is decidable and the decision problem for INDs is decidable, we do not know whether the decision problem for FDs and INDs together is decidable. This is one of the most interesting theoretical questions about INDs and FDs, that deserves further research.

REFERENCES

[ABU] A. V. Aho, C. Beeri and J. D. Ullman, "The Theory of Joins in Relational Databases," ACM Trans. on Database Systems 4,3 (September 1979), 297-314.

[Ar] W. W. Armstrong, "Dependency Structures of Database Relationships," Proc. IFIP 74 (1974), 580-583.

[BB] C. Beeri and P. A. Bernstein, "Computational Problems Related to the Design of Normal Form Relational Schemas," ACM Trans. on Database Systems 4,1 (March 1979), 30-59.

[BFH] C. Beeri, R. Fagin and J. H. Howard, "A Complete Axiomatization for Functional and Multivalued Dependencies in Database Relations," Proc. 1977 ACM SIGMOD Conf., 47-61.

[BK] C. Beeri and H. F. Korth, "Compatible Attributes in a Universal Relation", Proc. First ACM SIGACT-SIGMOD Conf. on Principles of Database Systems (1982), to appear.

[BV] C. Beeri and M. Y. Vardi, "Formal Systems for Tuple and Equality Generating Dependencies", Hebrew Univ. of Jerusalem Technical Report (April 1981).

[BG] P. A. Bernstein, N. Goodman, "What Does Boyce-Codd Normal Form Do?", Proc. 6th Int. Conf. on Very Large Databases (1980), 245-269.

[CFP] M. A. Casanova, R. Fagin, and C. H. Papadimitriou, "Inclusion Dependencies and their Interaction with Functional Dependencies", IBM Research Report RJ3380, San Jose, California.

[Ch] P. Chen, "The Entity-Relationship Model - Toward a Unified View of Data," ACM Trans. on Database Systems 1,1 (March 1976), 9-36.

[Co1] E. F. Codd, "Relational Completeness of Database Sublanguages," in Data Base Systems (R. Rustin, ed.), Prentice-Hall, New Jersey (1972).

[Co2] E. F. Codd, "Extending the Database Relational Model to Capture More Meaning," ACM Trans.

on Database Systems 4,4 (December 1980), 397-434.

[Fa1] R. Fagin, "Multivalued Dependencies and a New Normal Form for Relational Databases," ACM Trans. on Database Systems 2,3 (September 1977), 262-278.

[Fa2] R. Fagin, "Functional Dependencies in a Relational Database and Propositional Logic," IBM J. Res. and Develop. 21,6 (November 1977), 534-544.

[Fa3] R. Fagin, "A Normal Form for Relational Databases that is Based on Domains and Keys," ACM Trans. on Database Systems 6,3 (September 1981), pp. 387-415.

[Fa4] R. Fagin, "Horn Clauses and Database Dependencies," Proc. 1980 ACM SIGACT Symp. on Theory of Computing, 123-134. Also, to appear, J. ACM.

[FMUY] R. Fagin, D. Maier, J. D. Ullman and M. Yannakakis, "Tools for Template Dependencies," SIAM J. Computing, to appear.

[FV] R. Fagin and M. Y. Vardi, "Armstrong Databases for Interrelational Dependencies", to appear.

[GJ] M. R. Garey and D. S. Johnson, Computers and Intractibility: A Guide to the Theory of NP-Completeness, Freeman, 1979.

[JK] D. S. Johnson and A. Klug, "Testing Containment of Conjunctive Queries under Functional and Inclusion Dependencies", Proc. First ACM SIGACT-SIGMOD Conf. on Principles of Database Systems (1982), to appear.

[Ke] W. Kent, "Consequences of Assuming a Universal Relation," ACM Trans. on Database Systems 6,4 (December 1981), pp. 539-556.

[Kl] A. Klug, "Entity-Relationship Views Over Uninterpreted Enterprise Schemas," Proc. Int. Conf. Entity-Relationship Approach to Systems Analysis and Design (1979), 52-72.

[LP] H. R. Lewis and C. Papadimitriou, Elements of the Theory of Computation. Prentice-Hall (1981).

[Li] S. H. Lin, Existential Dependencies in Relational Databases. Ph.D. Thesis, UCLA (1981).

[Sa] W. J. Savitch, "Relationship Between Nondeterministic and Deterministic Tape Complexities," J. Comput. System Sci. 4 (1970), 177-192.

[SU] F. Sadri and J. D. Ullman, "A Complete Axiomatization for a Large Class of Dependencies in Relational Databases," Proc. 1980 ACM SIGACT Symp. on Theory of Computing, 117-122.

[SW] Y. Sagiv and S. Walecka, "Subset Dependencies and a Completeness Result for a Subclass of Embedded Multivalued Dependencies," to appear, J. ACM.

[SS] J. M. Smith and D. C. P. Smith, "Database Abstractions: Aggregation," Comm. ACM 20,6 (June 1977), 405-413.

[WM] G. Wiederhold, R. El-Masri, "A Structural Model for Database Systems," TR STAN-CS-79-722, Stanford University (February 1979).

[YP] M. Yannakakis and C. Papadimitriou, "Algebraic Dependencies," Proc. 21st IEEE Symp. on Found. of Computer Science (1980), 328-332.

[Za] C. Zaniolo, "Design of Relational Views over Network Schemas," Proc. 1979 ACM SIGMOD Conf., 179-190.