JOURNAL OF COMPUTER AND SYSTEM SCIENCES **28**, 29–59 (1984)

Inclusion Dependencies and Their Interaction with Functional Dependencies*

MARCO A. CASANOVA†

Brasilia Scientific Center, IBM do Brasil,
Caixa Postal 853, 70.000, Brasilia, DF, Brazil

RONALD FAGIN‡

IBM Research Laboratory, San Jose, California 95193

AND

CHRISTOS H. PAPADIMITRIOU

Department of Computer Science, Stanford University,
Stanford, California 94305

Received April 12, 1982; revised June 2, 1983

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.) Thus, no k-ary axiomatization can fully describe the interaction between FDs and INDs. This is true whether we consider finite implication or unrestricted implication. In the case of finite implication, this result holds, even if no relation scheme has more than two attributes, and if all of the dependencies are unary (a dependency is *unary* if the left-hand side and right-hand side each contain only one attribute). In the case of unrestricted implication, the result holds, even if no relation scheme has more than three attributes, each FD is unary, and each IND is binary.

* Extended abstract appeared in "Proceedings, First ACM SIGACT–SIGMOD Symposium on Principles of Database Systems," Los Angeles, pp. 171–176, 1982.

† This research was conducted while this author was at Pontificia Universidade Catolica do Rio de Janeiro, and was supported by FINEP and by CNPq Grant 402090/80.

‡ Most of 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.

0022-0000/84 $3.00

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.

1. Introduction

Functional dependencies (FDs) [Co1], are certainly the most important and widely studied integrity constraints for relational databases. Another important integrity constraint is the inclusion dependency (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. In general, an inclusion dependency is of the form

$$R[A_1,..., A_m] \subseteq S[B_1,..., B_m], \tag{1.1}$$

where R and S are relation names (possibly the same), and where the A_i's and B_i's are attributes. The inclusion dependency (1.1) holds for a database if each tuple that is a member of the relation corresponding to the left-hand side of (1.1) is also in the relation corresponding to the right-hand side of (1.1). Hence, INDs are valuable for database design, since they permit us to selectively define what data must be duplicated in what relations.

Together INDs and 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, Ul], 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]; they 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, Co3, Fa3, Kl, SS, WM, Za], there has been very little analysis of their properties, with only a few recent exceptions [CV, Da, FV, JK, KCV, Li, Mi1, Mi2, Sc], nearly all of which appeared after the first version of this paper. This paper was written to help remedy this neglect.

We show that INDs have a simple complete axiomatization. However, we also show the rather surprising fact that the decision problem for INDs (the problem of determining, given a set Σ of INDs and a single IND σ, whether or not Σ implies σ) is PSPACE-complete. Hence, there is no polynomial-time algorithm for this problem (unless P = PSPACE) [GJ].

Although INDs, like FDs, have a simple complete axiomatization, we show that the interaction of FDs and INDs is complex. In particular, we show that for no k does there exist a k-ary complete axiomatization for FDs and INDs taken together (by *k-ary*, we mean that each rule has at most k antecedents.) Thus, no k-ary axiomatization can fully describe the interaction between FDs and INDs. In particular, there is no finite complete axiomatization. To obtain this result, we give general necessary and sufficient conditions for the existence of a k-ary complete axiomatization. We also show how this characterization can be used to explain Sagiv and Walecka's [SW] result on the nonexistence of a k-ary complete axiomatization for embedded multivalued dependencies [Fa1], for arbitrary k.

In Section 2, we present basic definitions. In Section 3, we present a simple axiomatization for INDs, and show that it is complete. As a by-product of the proof, it follows that finite implication (implication over finite databases) is the same as unrestricted implication for INDs. We also show that the decision problem for INDs is PSPACE-complete. In Section 4, we give some examples that illustrate the interaction between FDs and INDs. Taken together, they can imply new dependencies, called *repeating dependencies* (RDs), which can say, for example, that in each tuple t of the R relation, the A and B entries of t are the same. We give a simple example that shows that finite implication is distinct from unrestricted implication, for FDs and INDs taken together. In Section 5, 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 6, we show that for no k does there exist 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, and if all of the dependencies are unary. (We say that a dependency is *unary* if the left-hand side and right-hand side each contain only one attribute.) In Section 7, we give the more complex construction which shows the same result for unrestricted implication (where no relation scheme has more than three attributes, each FD is unary, and each IND is binary.) In Sections 6 and 7, we show that even if RDs are included with FDs and INDs, then it is still the case that for no k does there exist a k-ary complete axiomatization. In Section 8, we present our conclusions, and suggest directions for further research.

We remark that the first author spent several months trying to find a finite axiomatization for FDs and INDs, and to prove its completeness. Of course, our results show that such a project was doomed! However, the knowledge acquired could be redirected toward proving the major result of Section 7.

We also note that by allowing the dynamic addition of new attributes, Mitchell [Mi1] has recently found a ternary "complete axiomatization" for FDs and INDs. (We put "complete axiomatization" in quotes, since Mitchell's approach involves a different notion of complete axiomatization than ours.) We also note that recently Mitchell [Mi2] and, independently, Chandra and Vardi [CV] have shown that the decision problem for FDs and INDs taken together is undecidable.

2. Definitions

A *relation scheme* is a pair $\langle R, U\rangle$, where R is the *name* of the relation scheme and where U is a finite sequence $\langle A_1,..., A_m\rangle$ of *attributes*, called the *attributes of* R. We use the notation $R[U]$ for $\langle R, U\rangle$. We usually write a sequence, such as $\langle A_1,..., A_m\rangle$, as simply $A_1,..., A_m$. For example, we shall write simply $R[A_1,..., A_m]$ for $R[\langle A_1,..., A_m\rangle]$. A *tuple* t over $U = \langle A_1,..., A_m\rangle$ is a sequence $\langle a_1,..., a_m\rangle$ of the same length m as U. A *relation* (over $R[U]$, or simply over R) is a set of tuples over U. If $\langle a_1,..., a_m\rangle$ is a tuple in relation r, then we say that a_i is an *entry* (*in column* A_i), $1 \leqslant i \leqslant m$. 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 $X = \langle A_{i_1},..., A_{i_k}\rangle$, where $i_1,..., i_k$ are distinct members of $\{1,..., n\}$, and if t is as above, then $t[X]$ is $\langle a_{i_1},..., a_{i_k}\rangle$. 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 finite 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 $X = \langle a_1,..., a_k\rangle$ 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 $R: X \to Y$ a *functional dependency* (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 $R: X \to Y$ 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 $R: X \to Y$ *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 $R_i[X] \subseteq R_j[Y]$ an *inclusion dependency* (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 $r_i[X] \subseteq r_j[Y]$.

The 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 $\Sigma \vDash_D \sigma$, or, if D is understood, simply $\Sigma \vDash \sigma$. If Σ and Γ are each sets of dependencies, then by $\Sigma \vDash \Gamma$, we mean that $\Sigma \vDash \gamma$ for each $\gamma \in \Gamma$. We write $\Sigma \vDash_{\text{fin}} \sigma$

to mean that whenever d is a *finite* database that obeys Σ, then also d obeys σ. Clearly, if $\Sigma \models \sigma$, then $\Sigma \models_{\text{fin}} \sigma$, but as we shall see, the converse is false. Finally, we write $\Sigma \not\models \sigma$ to mean that it is false that $\Sigma \models \sigma$.

3. Implication of INDs

In this section, we present a simple axiomatization for INDs, and show that it is complete. Our proof shows that the same axiomatization is complete, even if we restrict our attention to finite databases (databases with a finite number of tuples). Hence, finite implication and unrestricted implication ($\models_{\text{fin}}$ and $\models$) are the same for INDs. We show that the decision problem for INDs is PSPACE-complete. We close by relating our results to known results about special classes of first-order sentences.

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.

The axiomatization comprises the following inference rules (the first is a rule with no antecedents; such a rule is sometimes called an *axiom*).

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 $R[X] \subseteq T[Z]$.

Let Σ be a set of INDs and let σ be a single IND. A *proof* of σ from Σ is a finite sequence of INDs, where (1) each IND in the sequence is either a member of Σ, or else follows from previous INDs in the sequence by an application of the rules, and where (2) σ is the last IND in the sequence. We write $\Sigma \vdash \sigma$ to mean that there is a proof of σ from Σ.

Theorem 3.1 (Completeness theorem for INDs). *Let Σ be a set of* INDs, *and let σ be a single* IND. *The following are equivalent*:

(1) $\Sigma \models \sigma$,

(2) $\Sigma \models_{\text{fin}} \sigma$,

(3) $\Sigma \vdash \sigma$.

Proof. We shall show that (3) $\Rightarrow$ (1) $\Rightarrow$ (2) $\Rightarrow$ (3).

(3) $\Rightarrow$ (1) This is *soundness*, which simply says that the inference rules are valid. This is very easy to verify.

(1) $\Rightarrow$ (2) If every database that satisfies Σ also satisfies σ (i.e., if $\Sigma \models \sigma$), then clearly every finite database that obeys Σ also satisfies σ (i.e., $\Sigma \models_{\text{fin}} \sigma$).

(2) $\Rightarrow$ (3) Assume that $\Sigma \models_{\text{fin}} \sigma$. We must show that $\Sigma \vdash \sigma$. Assume that σ is the IND $R_a[A_1,..., A_m] \subseteq R_b[B_1,..., B_m]$. Assume that $R_1,..., R_n$ are the names of the relation schemes. We shall create a database $r_1,..., r_n$ over $R_1,..., R_n$, by adding tuples, one at a time. Later we shall make use of this database to prove that $\Sigma \vdash \sigma$.

Let p be a tuple over the attributes of R_a, such that $p[A_i] = i$ $(1 \leqslant i \leqslant m)$, and such that $p[A] = 0$ for each remaining attribute A of R_a. Initialize the database by letting r_a have the tuple p and no other tuple, and by letting each remaining r_i be empty. One at a time, we add tuples to the database by

Rule (*). Assume that the IND $R_i[C_1,..., C_k] \subseteq R_j[D_1,..., D_k]$ is in Σ, and that the tuple v is in r_i. Let t be a tuple over the attributes of R_j, where $t[D_u] = v[C_u]$, for $1 \leqslant u \leqslant k$, and where $t[A] = 0$ for each remaining attribute A of R_j. Then add the tuple t to r_j, if t is not already in r_j. We say that t is added to r_j *as a result of* the IND $R_i[C_1,..., C_k] \subseteq R_j[D_1,..., D_k]$ and of the tuple v of r_i.

Rule (*) is similar to the *chase* procedure [BV1, MMS, SU], except that instead of repeatedly introducing new undistinguished variables, we always use 0 when a "new" value is needed.

Apply Rule (*) until no more tuples can be added by applying it. Clearly, the resulting database $r_1,..., r_n$ is finite, since every entry of every tuple is in the set $\{0, 1,..., m\}$. It is easy to see that the database also satisfies Σ, or else Rule (*) could be applied to add another tuple. Since also, by assumption, $\Sigma \models_{\text{fin}} \sigma$, it follows that the database satisfies σ; that is, the database satisfies the IND $R_a[A_1,..., A_m] \subseteq R_b[B_1,..., B_m]$. So, since r_a contains the tuple p, as described above, it follows that r_b contains a tuple p', over the attributes of R_b, where $p'[B_i] = i$ $(1 \leqslant i \leqslant m)$.

Consider

Claim (**). If r_j contains a tuple t, with $t[E_u] = i_u \geqslant 1$, for $1 \leqslant u \leqslant k$, then $\Sigma \vdash R_a[A_{i_1},..., A_{i_k}] \subseteq R_j[E_1,..., E_k]$.

If we prove Claim (**), then since s contains the tuple p', it follows that $\Sigma \vdash R_a[A_1,..., A_m] \subseteq R_b[B_1,..., B_m]$; that is, $\Sigma \vdash \sigma$, which was to be shown.

Thus, we need only prove Claim (**). We prove Claim (**) by induction on when the tuple was inserted in the database (by Rule (*)). If t is the first tuple p that was inserted to initialize the database (so $r_j = r_a$), then Claim (**) is true, since $\Sigma \vdash R_a[A_{i_1},..., A_{i_k}] \subseteq R_a[A_{i_1},..., A_{i_k}]$, by IND1 (reflexivity). We now show that Claim (**) is true about tuple t, under the inductive assumption that it holds for all tuples previously inserted in the database. Assume that tuple t is inserted in relation r_j, via Rule (*), as a result of the IND $R_i[X_1,..., X_q] \subseteq R_j[Y_1,..., Y_q]$ of Σ and of the tuple v of r_i. Let us say that attribute X_w of R_i *corresponds* to attribute Y_w of R_j, for $1 \leqslant w \leqslant q$. Let F_u be the attribute of R_i that corresponds to attribute E_u of R_j $(1 \leqslant u \leqslant k)$, where the attributes E_u are as in Claim (**). Then $s[F_u] = i_u$, since $t[E_u] = i_u$ $(1 \leqslant u \leqslant k)$.

Since the IND $R_i[X_1,..., X_q] \subseteq R_j[Y_1,..., Y_q]$ is in Σ, it follows by IND2 (projection and permutation) that $\Sigma \vdash R_i[F_1,..., F_k] \subseteq R_j[E_1,..., E_k]$. By inductive assumption, Claim (**) holds when the roles of r_j and t are played by r_i and v, respectively. Hence, $\Sigma \vdash R[A_{i_1},..., A_{i_k}] \subseteq R_i[F_1,..., F_k]$. So, by IND3 (transitivity), it follows that $\Sigma \vdash R[A_{i_1},..., A_{i_k}] \subseteq R_j[E_1,..., E_k]$, which was to be shown. ∎

COROLLARY 3.2. *Let Σ be a set of* INDs, *and let σ be a single* IND $R_a[A_1,..., A_m] \subseteq R_b[B_1,..., B_m]$. *Then $\Sigma \vDash \sigma$ if and only if there is a sequence $S_1[X_1]$, $S_2[X_2]$,..., $S_w[X_w]$, where*

(i) *S_i is the name of one of the relation schemes, for $1 \leqslant i \leqslant w$;*

(ii) *X_i is a sequence of m distinct attributes of S_i, for $1 \leqslant i \leqslant w$;*

(iii) *the first member $S_1[X_1]$ of the sequence is $R_a[A_1,..., A_m]$;*

(iv) *the last member $S_w[X_w]$ of the sequence is $R_b[B_1,..., B_m]$; and*

(v) *the* IND $S_i[X_i] \subseteq S_{i+1}[X_{i+1}]$ *can be obtained from a member of Σ by* IND2 (*projection and permutation*), *for $1 \leqslant i \leqslant w$.*

By (v), *we mean that there is an* IND $S_i[C_1,..., C_k] \subseteq S_{i+1}[D_1,..., D_k]$ *in Σ, and that there is a sequence $i_1,..., i_m$ of distinct integers from $\{1,..., k\}$ such that X_i is $C_{i_1},..., C_{i_m}$ and X_{i+1} is $D_{i_1},..., D_{i_m}$.*

Proof. If there is such a sequence, then clearly $\Sigma \vDash \sigma$. Conversely, assume that $\Sigma \vDash \sigma$; we shall show that there is such a sequence. We shall make use of the database $r_1,..., r_n$ that we constructed in the proof of (2) ⇒ (3) in Theorem 3.1. By our construction, it follows that there is a sequence $(t_1, s_1),..., (t_w, s_w)$ such that

(a) s_i is one of the relations $r_1,..., r_n$ in the database, for $1 \leqslant i \leqslant w$;

(b) t_i is a tuple of s_i, for $1 \leqslant i \leqslant w$;

(c) $(t_1, s_1) = (p, r_a)$, and so t_1 is the first tuple inserted into the database in our construction;

(d) $(t_w, s_w) = (p', r_b)$, where p' is as described in the proof of Theorem 3.1;

(e) t_{i+1} is added to s_{i+1} as a result of a member of Σ and of the tuple t_i of s_i, for $1 \leqslant i < w$.

If t is a tuple that was inserted into the database, then let us say that t is *special* if t contains each of $1,..., m$ as entries. By definition, each of t_1 and t_w (i.e., each of p and p') is a special tuple. By reverse induction on i, it follows easily from (e) that each t_i is special. Thus, each t_i contains at least one occurrence of each of $1,..., m$. By (forward) induction on i, it then follows easily from (e) above that each t_i contains *exactly* one occurrence of each of $1,..., m$.

If $t_i[C_k] = k$ $(1 \leqslant k \leqslant m)$, and if s_i is r_j, then let us say that (t_i, s_i) *corresponds to* the expression $R_j[C_1,..., C_m]$ (and vice versa). Let $S_1[X_1],..., S_w[X_w]$ be the expressions that correspond to $(t_1, s_1),..., (t_w, s_w)$, respectively. It is straightforward to verify that the sequence $S_1[X_1],..., S_w[X_w]$ fulfills conditions (i)–(v) of the statement of Corollary 3.2. The member of Σ mentioned in (v) is the member of Σ mentioned in (e), for each i $(1 \leqslant i \leqslant w)$. ∎

Corollary 3.2 immediately leads to a decision procedure for the decision problem for INDs (i.e., 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 either $R_b[B_1,..., B_m]$ appears in Z, or until it is no longer possible to add an expression to Z by using step (2), whichever comes first.

(4) $\Sigma \models \sigma$ if and only if $R_b[B_1,..., B_m]$ is in the resulting set Z.

This decision procedure is nondetermininistic, since we do not specify the order in which the INDs in Σ are applied to members of Z in step (2). Our 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, our procedure requires superpolynomial time, as we now show.

Assume that there is a single relation scheme R, with m distinct attributes $A_1,..., A_m$. Associate with each permutation γ of $1,..., m$ the IND $R[A_1,..., A_m] \subseteq R[A_{\gamma(1)},..., A_{\gamma(m)}]$, which we shall denote by $\sigma(\gamma)$. Let γ_i $(1 \leqslant i \leqslant m)$ be the permutation of $1,..., m$ which maps 1 to i and maps i to 1, and which leaves everything else fixed. It is well known that the set $\{\gamma_1,..., \gamma_m\}$ of permutations generate all permutations on $1,..., m$ (i.e., every permutation of $1,..., m$ is a product $\beta_1 \cdots \beta_r$, where each β_i is a γ_j, and a given γ_j may appear many times). It follows easily that every IND over $R[A_1,..., A_m]$ is a logical consequence of the set $\{\sigma(\gamma_1),..., \sigma(\gamma_m)\}$ of INDs. Hence, if we were to make our decision procedure above deterministic by fixing the order in which INDs of Σ are applied to members of Z in step (2), then in the worst case, the procedure would run in (worse than) exponential time, since every IND over $R[A_1,..., A_m]$ will eventually appear in Z if step (2) is applied repeatedly until it cannot be applied anymore.

However, from what we have said it is still conceivable that for each pair Σ, σ for which $\Sigma \models \sigma$, there is some fortuitous choice (of how to apply the INDs of Σ to members of Z in step (2)) such that the algorithm shows that $\Sigma \models \sigma$, using only a polynomial number of steps. We now show that this is *not* the case.

If γ is a permutation, then define order(γ) to be the least integer k such that γ^k is the identity permutation. Define $f(m)$ to be max{order(γ): γ is a permutation of $1,..., m$}. Landau [La] has shown that $\log(f(m))$ is asymptotic to $(m \log m)^{1/2}$, where the logarithms are to the base e. Thus, $f(m)$ grows like $e^{(m \log m)^{1/2}}$. (Landau obtains a permutation of big order by composing it of relatively prime cycles.) Let γ be a permutation of order $f(m)$ on $1,..., m$, and let δ be $\gamma^{f(m)-1}$. Let $\sigma(\gamma)$ and $\sigma(\delta)$ be the associated INDs, as defined above. It is not hard to see that the IND $\sigma(\gamma)$ logically implies the IND $\sigma(\delta)$, and that the minimal number of applications of step (2) required for our algorithm to recognize that $\sigma(\gamma) \models \sigma(\delta)$ is $f(m) - 1$. Thus, a superpolynomial number of steps are required for our algorithm.

We remark that for the class of examples we just gave, there *are* short proofs that $\sigma(\gamma) \models \sigma(\delta)$ using our complete axiomatization from the start of Section 3. Thus, this particular class of examples is not one for which long (superpolynomial length) proofs are required under our complete axiomatization. However, if NP $\neq$ PSPACE, then long proofs are necessarily required in general, under our complete axiomatization or any other (as long as proofs are recognizable in polynomial time). This follows easily from the fact, which we shall show later in this section, that the decision problem for INDs is PSPACE-complete.

Since, as we just noted, the decision problem for INDs is PSPACE-complete, we know that there is no polynomial-time algorithm for this problem (unless P = PSPACE) [GJ]. However, it is easy to see that in certain special cases, the decision procedure given after Corollary 3.2 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 (i.e., INDs $R[A_1,..., A_r] \subseteq S[B_1,..., B_r]$, where $r \leqslant k$). (We note that Kannelakis, Cosmadakis, and Vardi [KCV] have shown that this problem, where k is fixed, is NLOGSPACE-complete.) 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 later type of IND, it is possible to say that every manager is an employee of the department that the manages by the IND MGR[NAME, DEPT] $\subseteq$ EMP[NAME, DEPT], where, say (Hilbert, Math) is a tuple of the MGR relation if Hilbert manages the Math Department, etc.

We now show the main result of this section.

THEOREM 3.3. *The decision problem for* INDs *is* PSPACE-*complete.*

Proof. We first show that the decision problem for INDs is in PSPACE. We do this by describing 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 IND $S(m, j)$

$$R[P_j, (a, j), (b, j+1), (c, j+2)] \subseteq R[P_j, (a', j), (b', j+1), (c', j+2)],$$

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 now show that $\Sigma \models \sigma$ if and only if M accepts x in space n.

Assume first that M accepts x in space n. Let $Y_1, Y_2, ..., Y_w$ be a sequence of configurations such that Y_1 is the initial configuration sx, that Y_w is the final configuration hB^n, and configuration Y_i immediately yields Y_{i+1} $(1 \leqslant i \leqslant w)$. If Y_i is $y_1 \cdots y_{n+1}$, then define X_i to be the sequence $\langle (y_1, 1), ..., (y_{n+1}, n+1) \rangle$. It is easy to see that $R[X_1], ..., R[X_w]$ is a sequence as in Corollary 3.2. Hence, $\Sigma \models \sigma$.

Conversely, assume that $\Sigma \models \sigma$. Let $R[X_1], ..., R[X_w]$ be the sequence demanded by Corollary 3.2, where $R[X_1]$ is $R[(s, 1), (x_1, 2), ..., (x_n, n+1)]$, where $R[X_w]$ is $R[(h, 1), (B, 2), ..., (B, n+1)]$. Of course, each S_i in Corollary 3.2 is R. It is easy to see, inductively on i, that each X_i can be obtained from a configuration Y_i by the transformation described in the previous paragraph, and further, that Y_1 is the initial configuration sx, Y_w is the final configuration hB^n, and that configuration Y_i immediately yields Y_{i+1} $(1 \leqslant i < w)$. Hence, M accepts x in space n, as desired. ∎

We close this section by relating our results to known results about various classes of first-order sentences. The *extended Maslov class* [DG] consists of sentences in prenex normal form with quantifier structure $\forall\exists\forall$, and whose quantifier-free part is a conjunction of binary disjunctions. It is known [DG] that a sentence in the extended Maslov class is satisfiable (true in some database) if and only if it is finitely satisfiable (true in some finite database). If Σ is a finite set of INDs, and if σ is a single IND, then it is easy to see that there is a first-order sentence which is equivalent to $\Sigma \wedge \neg\sigma$ and which is in the extended Maslov class. Hence, $\Sigma \wedge \neg\sigma$ is satisfiable if and only if it is finitely satisfiable. But this means that $\Sigma \models \sigma$ if and only if $\Sigma \models_{\text{fin}} \sigma$. Thus, the equivalence of implication and finite implication for INDs (Theorem 3.1(1),(2)) follows from known results. Further, it is well known [DG] that the equivalence of implication and finite implication for a class of sentences implies the decidability of their decision problem. Hence, the decidability of the decision problem for INDs follows from classic results. We have shown something stronger, namely, PSPACE-completeness.

We know of another example of a PSPACE-completeness result for a decision problem for dependencies. Chandra, Lewis, and Makowsky [CLM] showed that the decision problem for untyped full implicational dependencies [Fa4] for which the left-hand side and right-hand side each contain exactly one conjunct is PSPACE-complete. This result is incomparable with ours.

4. Examples of Interactions between FDs and INDs

In this section, we give a few simple results that help us to understand the interaction between FDs and INDs. As we shall see, a new class of dependencies arises quite naturally from the interplay between FDs and INDs. We also show that when we restrict our attention to finite databases, implication is distinct from unrestricted implication, for FDs and INDs taken together. A similar result was shown for template dependencies by Fagin, Maier, Ullman, and Yannakakis [FMUY], although the proof in the case of template dependencies is much harder.

Throughout this section, each of $T,..., Z$ denote sequences of distinct attributes, and XY denotes the concatentation of X and Y. Since all attributes have to be distinct on each side of an IND, when we write $R[XY] \subseteq S[TU]$, say, we are implicitly saying that X and Y are disjoint, as well as T and U.

We now give two simple propositions, the first of which describes a situation where a set of INDs and FDs imply a new FD, and the second of which describes a situation where a set of INDs and FDs imply a new IND. Other such cases will be presented in Section 7. We shall make use of Proposition 4.1 in Section 7.

Proposition 4.1. *Assume that* $|X| = |T|$. *Then* $\{R[XY] \subseteq S[TU], S: T \rightarrow U\} \models R: X \rightarrow Y$.

Proof. Let r, s be a database that satisfies $R[XY] \subseteq S[TU]$ and $S: T \to U$. Let $u, v \in r$ be such that $u[X] = v[X]$. Then there are $u', v' \in s$ such that $u'[TU] = u[XY]$ and $v'[TU] = v[XY]$. Hence, $u'[T] = v'[T]$. But, since s satisfies $S: T \to U$, and since $u'[T] = v'[T]$, it follows that $u'[U] = v'[U]$. Therefore, $u[Y] = v[Y]$. Thus, if $u, v \in r$ and $u[X] = v[X]$, then $u[Y] = v[Y]$. That is, the database satisfies $R: X \to Y$. ∎

PROPOSITION 4.2. *Assume that* $|X| = |T|$. *Then* $\{R[XY] \subseteq S[TU], R[XZ] \subseteq S[TV], S: T \to U\} \models R[XYZ] \subseteq S[TUV]$.

Proof. Let r, s be a database that satisfies $R[XY] \subseteq S[TU]$, $R[XZ] \subseteq S[TV]$ and $S: T \to U$. Take $u \in r$. Then there are v', $v'' \in s$ such that $v'[TU] = u[XY]$ and $v''[TV] = u[XZ]$. Hence, $v'[T] = v''[T]$. But since s satisfies $S: T \to U$, it follows that $v'[U] = v''[U]$. Thus, $v''[U] = u[Y]$, since $v'[U] = u[Y]$. Therefore, for each $u \in r$, there is $v'' \in s$ such that $v''[TUV] = u[XYZ]$; that is, r, s satisfies $R[XYZ] \subseteq S[TUV]$. ∎

Proposition 4.2 has one important degenerate case which we state as

PROPOSITION 4.3. *Assume that* $|X| = |T|$. *Let* $\Sigma = \{R[XY] \subseteq S[TU], R[XZ] \subseteq S[TU], S: T \to U\}$. *If* r, s *is a database satisfying* Σ, *and if* $u \in r$, *then* $u[Y] = u[Z]$.

Proof. Can be obtained by an obvious modification of the proof of Proposition 4.2. ∎

Proposition 4.3 leads us to consider a new type of dependency, called a *repeating dependency*, or RD. An RD is a statement of the form $R[X = Y]$, where X and Y are sequences of attributes of R, with $|X| = |Y|$. A relation r over R obeys the RD $R[X = Y]$, if whenever u is a tuple of r, then $u[X] = u[Y]$. Proposition 4.3 says the INDs $R[XY] \subseteq S[TU]$ and $R[XZ] \subseteq S[TU]$, taken together with the FD S: $T \to U$, imply the RD $R[Y = Z]$. It is not hard to verify that RDs are new dependencies, in the sense that if $R[X = Y]$ is a nontrivial RD (one for which $X \neq Y$), then $R[X = Y]$ is not equivalent to a set of FDs and INDs. (However, RDs are equivalent to a special case of a generalized type of IND, utilized by Mitchell [Mi1], where we allow an attribute to be repeated several times on the same side, i.e., left-hand side or right-hand side, of a generalized IND.) Note that the RD $R[A_1, \ldots, A_m = B_1, \ldots, B_m]$, where each A_i and B_i is an attribute, is equivalent to the set $\{R[A_i = B_i]: i = 1, \ldots, m\}$ of unary RDs; the comparable statement about INDs is false.

We also note that RDs are special cases of extended embedded implicational dependencies [Fa4]. The RDs arise naturally in the equijoin of Codd [Co2] and in the extended relations of Yannakakis and Papadimitriou [YP], since in these cases, there may be duplicate copies of a column.

Sometimes, by enlarging the class of dependencies under study, it is possible to cause a k-ary complete axiomatization to exist where none did before. For example,

A	B
1	0
2	1
3	2
4	3
⋮	⋮

FIGURE 4.1

for no k does there exist a k-ary complete axiomatization for embedded multivalued dependencies [SW, Sect. 5]. However, the larger class of template dependencies has a 2-ary complete axiomatization [BV2, SU]. In this paper, we show that for no k does the collection of FDs and INDs taken together have a k-ary complete axiomatization. Is it possible that the larger class of FDs, INDs, and RDs taken together have a k-ary complete axiomatization for some k? As we shall show in Sections 6 and 7, this is not the case. (We remark that Abiteboul and Vardi [AV] have shown that if we enlarge the class still further to encompass all untyped EIDs, then there is a binary complete axiomatization.)

We close this section by showing that finite implication is distinct from unrestricted implication, for FDs and INDs taken together. By contrast, we showed in Section 3 that finite and unrestricted implication are the same for INDs. Also, it is well known that a similar remark holds for FDs.

THEOREM 4.4. (a) *There is a set Σ of* FDs *and* INDs *and a single* IND σ *such that $\Sigma \models_{\text{fin}} \sigma$, but such that it is false that $\Sigma \models \sigma$.*

(b) *There is a set Σ of* FDs *and* INDs *and a single* FD σ *such that $\Sigma \models_{\text{fin}} \sigma$, but such that it is false that $\Sigma \models \sigma$.*

Proof. (a) Let Σ be $\{R: A \to B, R[A] \subseteq R[B]\}$, and let σ be $R[B] \subseteq R[A]$. We first show that $\Sigma \models_{\text{fin}} \sigma$. Let r be a finite relation satisfying Σ. We now show that r obeys σ; that is, that $r[B] \subseteq r[A]$. Since r obeys $R: A \to B$, it follows that $|r[B]| \leqslant |r[A]|$. Since $r[A] \subseteq r[B]$, it follows that $|r[A]| \leqslant |r[B]|$. Thus $|r[A]| = |r[B]|$. But since $r[A] \subseteq r[B]$ and since both $r[A]$ and $r[B]$ are finite, we then have $r[A] = r[B]$, and so $r[B] \subseteq r[A]$. This was to be shown.

To show that it is false that $\Sigma \models \sigma$, we need only exhibit a relation (necessarily infinite) that obeys Σ but not σ. Let r be the relation in Fig. 4.1, with tuples $\{(i+1, i): i \geqslant 0\}$. It is obvious that r obeys Σ but not σ.

(b) Let Σ be $\{R: A \to B, R[A] \subseteq R[B]\}$, and let σ be $R: B \to A$. We first show that $\Sigma \models_{\text{fin}} \sigma$. Let r be a finite relation satisfying Σ. We now show that r obeys the FD $B \to A$. Let k be the number of tuples of r. Since r obeys the FD $A \to B$, it follows that no two distinct tuples of r have the same A entry, and so there are k distinct values in $r[A]$. Because r obeys Σ, we know that $r[A] \subseteq r[B]$, and so $r[B]$ contains at least k (and hence exactly k) distinct values. Because r is finite, and the number of distinct values in $r[B]$ is the same as the number of tuples, it follows that no two distinct tuples of r have the same B entry. Therefore, r obeys the FD $B \to A$, which was to be shown.

A	B
1	1
2	1
3	2
4	3
⋮	⋮

FIGURE 4.2

To show that it is false that $\Sigma \models \sigma$, we need only exhibit a relation (necessarily infinite) that obeys Σ but not σ. Let r be the relation in Fig. 4.2, with tuples $\{(1, 1)\} \cup \{(i+1, i): i \geqslant 1\}$. It is obvious that r obeys Σ but not σ. ▮

5. 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 Sections 6 and 7, 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 ($1 \leqslant i \leqslant n$). 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 $T = \varnothing$) 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 $\langle \tau_1, ..., \tau_m \rangle$ of sentences in $\mathscr{S}$, where τ_m, the last sentence in the sequence, is σ, and where for each i ($1 \leqslant i \leqslant m$), 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 (i.e., 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 $r \leqslant k$.

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 later sections, 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] complete axiomatization for template dependencies, and the Beeri and Vardi [BV2] and Yannakakis and Papadimitriou [YP] complete axiomatization for embedded implicational dependencies (which Yannakakis and Papadimitriou call "algebraic dependencies"). 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.

We note also that when we speak of a "complete axiomatization," we make no assumption that the set of axioms is recursive (although, in practice, a recursive set of axioms is certainly desirable.) This makes our results stronger, since we prove the *nonexistence* of a k-ary complete axiomatization.

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

THEOREM 5. *Let D be a database scheme, let $\mathscr{S}$ be a set of sentences about D, and let $k \geqslant 0$ be an integer. There is a k-ary complete axiomatization for sentences in $\mathscr{S}$ if and only if whenever $\Gamma \subseteq \mathscr{S}$ is closed under k-ary implication, then Γ is closed under implication.*

Proof. ($\Rightarrow$) Assume that there is a k-ary complete set $\mathscr{R}$ of rules. Let Γ be a subset of $\mathscr{S}$ that is closed under k-ary implication; we must show that Γ is closed under implication. Assume that $\Sigma \subseteq \Gamma$, and that $\Sigma \models_D \sigma$. We must show that $\sigma \in \Gamma$. Since $\Sigma \models_D \sigma$, it follows by completeness that $\Sigma \vdash_{\mathscr{R}} \sigma$. Let $\langle \tau_1,..., \tau_m \rangle$ be a proof of σ from Σ via $\mathscr{R}$. Thus, $\tau_m = \sigma$. We shall show by induction on i that $\tau_i \in \Gamma$.

Basis step ($i = 1$). The first sentence τ_1 in the proof is either a member of Σ, or else there is a 0-ary rule "if $\varnothing$ then τ_1" in $\mathscr{R}$. In the former case, $\tau_1 \in \Gamma$, since $\Sigma \subseteq \Gamma$. In the latter case, by soundness of the rules $\mathscr{R}$, we know that τ_1 is a tautology. Since Γ is closed under k-ary implication, and since $k \geqslant 0$, it follows that Γ contains all tautologies in $\mathscr{S}$, and thus $\tau_1 \in \Gamma$. So in either case, $\tau_1 \in \Gamma$.

Induction step. Assume that $1 \leqslant i \leqslant m$, and that $\{\tau_1,..., \tau_i\} \subseteq \Gamma$. We shall show that $\tau_{i+1} \in \Gamma$. Either $\tau_{i+1} \in \Sigma$ (in which case $\tau_{i+1} \in \Gamma$, since $\Sigma \subseteq \Gamma$), or else there is a rule "if T then τ_{i+1}" in $\mathscr{R}$, where T is a subset of $\{\tau_1,..., \tau_i\}$. In the latter case, we know that T has at most k members, since $\mathscr{R}$ is a k-ary set of rules. Every member of T is in Γ, by induction hypothesis. Also, $T \models_D \tau_{i+1}$, by soundness of the rules. Since Γ is closed under k-ary implication, it follows that $\tau_{i+1} \in \Gamma$.

We have shown inductively that $\tau_1,..., \tau_m$ are each in Γ. In particular, $\tau_m \in \Gamma$, that is, $\sigma \in \Gamma$. This was to be shown.

($\Leftarrow$) Assume that there is no k-ary complete axiomatization for sentences in $\mathscr{S}$. We shall construct a set $\Gamma \subseteq \mathscr{S}$ that is closed under k-ary implication but is not closed under implication. Let $\mathscr{R}$ be the set of all rules "if T then τ," where T is a set of at most k members of $\mathscr{S}$, where $\tau \in \mathscr{S}$, and where $T \models_D \tau$. Now $\mathscr{R}$ is a k-ary set of rules, so by assumption, it is not complete. Clearly, $\mathscr{R}$ is sound. Since $\mathscr{R}$ is sound but not complete, it follows that there is a set $\Sigma \subseteq \mathscr{S}$ and a sentence $\sigma \in \mathscr{S}$ such that $\Sigma \models_D \sigma$ but such that it is false that $\Sigma \vdash_{\mathscr{R}} \sigma$. Let Γ be the set of all members γ of $\mathscr{S}$ for which $\Sigma \vdash_{\mathscr{R}} \gamma$. 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, which completes the proof. Assume that T is a set of at most k members of Γ, that $\tau \in \mathscr{S}$, and that $T \models_D \tau$. We must show that $\tau \in \Gamma$. By our assumptions, $\mathscr{R}$ contains the rule "if T then τ." Since $T \subseteq \Gamma$, there is a proof of each member of T from Σ, via $\mathscr{R}$. The result of concatenating all of these proofs and following this concatenation by τ is clearly a proof of τ from Σ, via $\mathscr{R}$ (where the last line τ of the proof is justified because of the rule "if T then τ" of R). Thus, $\Sigma \vdash_{\mathscr{R}} \tau$, and so $\tau \in \Gamma$. Thus, Γ is closed under k-ary implication, which was to be shown. ∎

We now give a corollary which we shall use to explain Sagiv and Walecka's [SW] result on the nonexistence of a k-ary complete axiomatization for embedded multivalued dependencies (EMVDs), for each k. Assume that each of X, Y, and Z are sets of attributes, that Y and Z are disjoint, and that r is a relation whose set of attributes includes $X \cup Y \cup Z$. The relation r is said to obey the EMVD $X \twoheadrightarrow Y \mid Z$ if whenever there are tuples t_1 and t_2 of r such that $t_1[X] = t_2[X]$, then there is a tuple t_3 of r such that $t_3[XY] = t_1[XY]$ and $t_3[XZ] = t_1[XZ]$. For a discussion (and motivation) of EMVDs, see [Fa1].

COROLLARY 5.2. *Let D be a database scheme, let $\mathscr{S}$ be a set of sentences about D, and let $k \geqslant 0$ be a constant. Assume that $\Sigma \subseteq \mathscr{S}$, that $\sigma \in \mathscr{S}$, and that*

(i) $\Sigma \vDash \sigma$,

(ii) *if $\tau \in \Sigma$ then it is false that $\tau \vDash \sigma$, and*

(iii) *if Δ is a set of at most k members of Σ, if $\tau \in \mathscr{S}$ and if $\Delta \vDash \tau$, then there is some $\delta \in \Delta$ such that $\delta \vDash \tau$.*

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

Proof. Let $\Gamma = \{\tau \in \mathscr{S}$: there is $\tau' \in \Sigma$ such that $\tau' \vDash \tau\}$. 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 $\tau \in \mathscr{S}$ and that $T \vDash \tau$. We must show that $\tau \in \Gamma$. For each α in T, find $\alpha' \in \Sigma$ such that $\alpha' \vDash \alpha$ (we know that α' exists by definition of Γ). Let $\Delta = \{\alpha' : \alpha \in T\}$. Clearly $\Delta \vDash \tau$, since $\Delta \vDash T$ and $T \vDash \tau$. 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 5.1 that there is no k-ary complete axiomatization for sentences in $\mathscr{S}$. This was to be shown. ∎

THEOREM 5.3 [SW]. *For no k does there exist a k-ary complete axiomatization for* EMVDs. *That is, given k, there is a relation scheme R such that there is no k-ary complete axiomatization for* EMVDs *over R.*

Proof. Let R be a relation scheme with at least $k+2$ distinct attributes $A_1, \ldots, A_{k+1}, B$, let Σ be the set

$$\begin{array}{c} A_1 \twoheadrightarrow A_2 \mid B \\ A_2 \twoheadrightarrow A_3 \mid B \\ \vdots \\ A_k \twoheadrightarrow A_{k+1} \mid B \\ A_{k+1} \twoheadrightarrow A_1 \mid B \end{array}$$

of EMVDs, let σ be the EMVD $A_1 \twoheadrightarrow A_{k+1} \mid B$, and let $\mathscr{S}$ be the set of EMVDs over $\mathscr{R}$. Sagiv and Walecka [SW] show that the conditions of Corollary 5.2 hold. The result follows. ∎

Because of the subtlety of the issues, it is prudent to make a warning. Let us denote by Σ_k the set Σ of EMVDs in the proof of Theorem 5.3, and similarly let σ_k be σ of Theorem 5.3. Then the rule "if Σ_k then σ_k" is a rule with $k+1$ 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

$$\begin{gathered} A_1 \to A_2 \\ A_2 \to A_3 \\ \vdots \\ A_{k+1} \to A_{k+2} \end{gathered}$$

of FDs, and let τ_k be the FD $A_1 \to A_{k+2}$. Then the rule "if T_k then τ_k" has this same property, yet FDs have a 2-ary complete axiomatization [Ar, Fa2].

6. Negative Results for Finite Implication

In this and the next section, we use the main theorem of Section 5 to prove that there is no k-ary complete axiomatization for FDs and INDs or for FDs, INDs, and RDs. In this section we concentrate on finite implication, which means that an axiomatization $\mathscr{R}$ is considered complete here when $\Sigma \vdash_{\mathscr{R}} \sigma$ if and only if $\Sigma \models_{\text{fin}} \sigma$. In the next section we deal with unrestricted implication (as we showed in Section 4, finite implication is distinct from unrestricted implication for FDs and INDs taken together.) At the end of this section, we comment on how our results in this section can be strengthened somewhat.

THEOREM 6.1. (i) *For no k does there exist a k-ary complete axiomatization for finite implication of* FDs *and* INDs.

(ii) *For no k does there exist a k-ary complete axiomatization for finite implication of* FDs, INDs, *and* RDs.

Note. By Theorem 6.1(i), 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 (ii).

Proof. We first prove (ii). Let k be a fixed natural number. Let $R_i[AB]$ $(0 \leqslant i \leqslant k)$ be a set of relation schemes. Define (where, henceforth, addition is modulo k):

(1) $\Sigma = \{R_i\colon A \to B, R_i[A] \subseteq R_{i+1}[B]\colon 0 \leqslant i \leqslant k\}$, and

(2) $\sigma = R_0[B] \subseteq R_k[A]$.

Let Γ be the union of Σ with the set of all trivial FDs, INDs, and RDs (those that are tautologies). By Theorem 5.1 (where finite implication plays the role of implication, that is, where $\models_{\text{fin}}$ 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 $\Sigma \models_{\text{fin}} \sigma$, since it is immediate that $\Sigma \subseteq \Gamma$ and that $\sigma \notin \Gamma$. Let $d = \{r_0, ..., r_k\}$ be a finite database satisfying Σ. Since d satisfies $R_i[A] \subseteq R_{i+1}[B]$, it follows that $|r_i[A]| \leqslant |r_{i+1}[B]|$, for $0 \leqslant i \leqslant k$. Since d satisfies $R_i: A \to B$, it follows that $|r_i[B]| \leqslant |r_i[A]|$ holds, for $0 \leqslant i \leqslant k$. Putting these inequalities together, we obtain $|r_0[A]| \leqslant |r_1[B]| \leqslant |r_1[A]| \leqslant \cdots \leqslant |r_k[A]| \leqslant |r_0[B]| \leqslant |r_0[A]|$. Hence, $|r_k[A]| = |r_0[B]|$. But since d satisfies the IND $R_k[A] \subseteq R_0[B]$ and since d is finite, we then have $r_k[A] = r_0[B]$. Hence, $r_0[B] \subseteq r_k[A]$, and so d obeys σ.

We conclude the proof of (ii) 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, IND, or RD, and if $T \models_{\text{fin}} \tau$, then $\tau \in \Gamma$.

Since Σ contains $k+1$ INDs, we know that T does not contain some IND δ of Σ. We shall exhibit a finite database d where precisely the dependencies (FDs, INDs, and RDs) in $\Gamma - \delta$ are true. (In the terminology of Fagin [Fa4], the database d is a *finite Armstrong database* for $\Gamma - \delta$.)

Since $T \subseteq \Gamma - \delta$, it follows that d obeys T. Because $T \models_{\text{fin}} \tau$, we also know that d obeys τ. Since d obeys precisely $\Gamma - \delta$, it follows that $\tau \in \Gamma - \delta$. Hence, $\tau \in \Gamma$, which was to be shown.

Thus, the proof of (ii) is complete if we exhibit a finite database d such that

$$\text{if } \tau \text{ is an FD, IND, or RD, then } d \text{ obeys } \tau \text{ if and only if } \tau \in \Gamma - \delta. \tag{6.1}$$

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 $d = \{r_0, ..., r_k\}$ as

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

$$r_i = \{((0,i), (0, i-1)), ((1,i), (1, i-1)), ..., ((2i+1, i), (2i+1, i-1)), ((2i+2, i), (2i+1, i-1))\}, \quad \text{for} \quad 1 \leqslant i \leqslant k.$$

Figure 6.1 exhibits d for $k = 3$.

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 6.1

We now show (6.1). If τ is trivial, then clearly (6.1) holds. So assume τ is nontrivial.

Case 1. τ is a nontrivial FD. The only nontrivial FDs are

$$R_i: A \to B, \qquad R_i: \varnothing \to A,$$
$$R_i: B \to A, \qquad R_i: \varnothing \to B,$$

for $0 \leqslant 1 \leqslant k$. (An FD with $\varnothing$ as the left-hand side means that the right-hand side entries are constants. For example, $R_i: \varnothing \to A$ means that every A entry in the R_i relation is the same.) If τ is $R_i: A \to B$, then d obeys τ (since no two A entries of r_i are the same), and $\tau \in \Gamma - \delta$ $(0 \leqslant i \leqslant k)$. In each of the other three cases, it is easy to see that d violates the FD τ, and that $\tau \notin \Gamma - \delta$.

Case 2. τ is a nontrivial IND. Each entry of $r_i[A]$ is of the form (m, i), for $0 \leqslant i \leqslant k$; each entry of $r_i[B]$ is of the form $(n, i-1)$, for $1 \leqslant i \leqslant k$; and each entry of $r_0[B]$ is of the form $(p, k+1)$. Thus, the only pairs of nondisjoint columns are

$$r_0[A], r_1[B]$$
$$r_1[A], r_2[B]$$
$$\vdots$$
$$r_{k-1}[A], r_k[B].$$

No IND $r_{i+1}[B] \subseteq r_i[A]$ holds for d $(0 \leqslant i < k)$, since $r_{i+1}[B]$ contains the entry $(2i+3, i)$, which is not in $r_i[A]$. Thus, the only possible nontrivial INDs that can hold for d are those of the form $R_i[A] \subseteq R_{i+1}[B]$, for $0 \leqslant i < k$. In fact, it is easy to verify that each of these INDs hold for d, and that these INDs are precisely the nontrivial INDs in $\Gamma - \delta$.

Case 3. Sentence τ is a nontrivial RD. Then d does not obey τ, and $\tau \notin \Gamma - \delta$.

Thus, (6.1) holds. This concludes the proof of part (ii) of the theorem. Since RDs played no essential role in the previous proof, it is clear that the same proof shows part (i). ∎

We now make a number of comments concerning Theorem 6.1 and its proof.

Let $\mathscr{S}$ be a class of dependencies such that the database d constructed in the proof of the previous theorem violates every nontrivial member of $\mathscr{S}$. Then our proof shows that there is no k-ary complete axiomatization for finite implication of FDs, INDs, and dependencies in $\mathscr{S}$. For example, if we let $\mathscr{S}$ 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 d obeys no nontrivial MVDs.

Our proof shows that Theorem 6.1 holds, even if no relation scheme has more than two attributes, and if all of the dependencies are unary. In the case of finite implication involving unary INDs and (general) FDs, Kannelakis *et al.* [KCV] have presented a complete axiomatization, which is, of course, not k-ary for any k.

Our proof demonstrates a set Γ of FDs, INDs, and perhaps RDs, that is closed under k-ary implication, along with an IND σ not in Γ such that $\Gamma \models_{\text{fin}} \sigma$. We note that there is also such an FD σ (namely, $R_0: B \to A$). In the proof of Theorem 7.1 (which deals with unrestricted implication), a similar Γ and σ are constructed. In that proof, the σ we use is an FD. We note without proof that by slightly complicating the proof, we could just as well have used an IND σ.

A key to the proof of the previous theorem is the class of inference rules "if Σ_k then σ_k," where Σ_k and σ_k are the Σ and σ of the proof. We make two observations about this class of inference rules.

The first observation is that we proved the soundness (for finite implication) of these inference rules by using a counting argument. The counting argument depended on the fact that we are dealing with finite databases; in fact, these rules are not sound for unrestricted databases.

The second observation is that an important feature of this class of inference rules "if Σ_k then σ_k" (for $k = 1, 2, 3,...$) is that for each n, it contains a rule with at least n antecedents, none of which can be eliminated and leave a sound rule. However, as we showed at the end of Section 5, this feature alone is not enough to show that for no k does there exist a k-ary complete axiomatization.

7. Negative Results for Unrestricted Implication

The main result of this section is identical to Theorem 6.1, except that we assume unrestricted implication this time.

THEOREM 7.1. (i) *For no k does there exist a k-ary complete axiomatization for* FDs *and* INDs.

(ii) *For no k does there exist a k-ary complete axiomatization for* FDs, INDs, *and* RDs.

Note. By Theorem 7.1(i), we mean that for each k, there is a database scheme D such that there is no k-ary complete axiomatization for FDs and INDs over D. A similar comment applies to (ii).

Note that the proof of Theorem 6.1 does not imply Theorem 7.1, since the proof of the former relied on the assumption of finite databases to justify a counting argument. However, as we shall see, the proof of Theorem 7.1 does imply Theorem 6.1. This remark does not make Section 6 superfluous, since the proof of Theorem 6.1 is much simpler than the one we give for Theorem 7.1.

Our proof of Theorem 7.1 shows that the theorem holds, even if no relation scheme has more than three attributes, each FD is unary, and each IND is binary. We note that in the case of unrestricted implication, it is necessary to make use of INDs that are at least binary to prove Theorem 7.1, since Kannelakis *et al.* [KCV] have presented a binary complete axiomatization for FDs and unary INDs in the case of unrestricted implication. (This should be contrasted with the case of finite implication, where we showed in Theorem 6.1 that there is no k-ary complete axiomatization for FDs and unary INDs.)

Proof of Theorem 7.1. As in the proof of Theorem 6.1, we shall use Theorem 5.1 to prove part (ii), and then note that essentially the same reasoning carries over to prove part (i). Thus, to prove part (ii), we shall show that for each $k \geqslant 0$, there is a database scheme D and a set Γ of FDs and INDs over D that is closed under k-ary implication (with respect to FDs, INDs, and RDs), but is not closed under implication.

Let k and n be two fixed natural numbers such that $k < n$. Denote by Π the set of all FDs, INDs, and RDs over the relation schemes $F[ABC]$, $G_0[ABC]$, $G_i[BC]$ $(i = 1,..., n)$, $H_i[BC]$ $(i = 0,..., n-1)$ and $H_n[BCD]$. We now define Σ, $\Gamma \subseteq \Pi$, and $\sigma \in \Pi$.

Define Σ as the set of dependencies

$$\begin{aligned}
&\alpha_0 = F[AB] \subseteq G_0[AB], && \\
&\alpha_i = F[B] \subseteq G_i[B] && (1 \leqslant i \leqslant n), \\
&\beta_i = F[B] \subseteq H_i[B] && (0 \leqslant i < n), \\
&\beta_n = F[BC] \subseteq H_n[BD], && \\
&\gamma_i' = H_i[BC] \subseteq G_i[BC] && (0 \leqslant i \leqslant n), \\
&\gamma_i'' = H_i[BC] \subseteq G_{i+1}[BC] && (0 \leqslant i < n), \\
&\delta_0 = G_0: A \to C, && \\
&\varepsilon_i = G_i: B \to C && (0 \leqslant i \leqslant n), \\
&\theta_n = H_n: C \to D. &&
\end{aligned}$$

Define σ as $F: A \to C$.

If ζ is a set of dependencies of type $t \in \{\text{FD, IND, RD}\}$, then we use ζ^+ to denote the set of all logical consequences of ζ of type t. For example, if ζ is a set of FDs, then ζ^+ is the set of all FDs that are logical consequences of ζ. Using this notation, we define the sets of dependencies

$$\begin{aligned}
\phi(F) &= \{F: A \to C, F: B \to C\}, && \\
\phi(G_0) &= \{G_0: A \to C, G_0: B \to C\}, && \\
\phi(G_i) &= \{G_i: B \to C\} && (1 \leqslant i \leqslant n),
\end{aligned}$$

$$\phi(H_i) = \{H_i: B \to C\} \qquad (0 \leqslant i < n),$$

$$\phi(H_n) = \{H_n: B \to C, H_n: C \to D\},$$

$$\phi = \phi(F) \cup \phi(G_0) \cup \cdots \cup \phi(G_n) \cup \phi(H_0) \cup \cdots \cup \phi(H_n),$$

$$\lambda = \{\tau \in \Sigma: \tau \text{ is an IND}\},$$

$$\omega = \{\tau: \tau \text{ is a trivial RD}\},$$

$$\Gamma = \phi^+ \cup \lambda^+ \cup \omega - \{F: A \to C\}.$$

We now show that $\Sigma \models \sigma$. Since $\Sigma \subseteq \Gamma$ but $\sigma \notin \Gamma$, this implies that Γ is not closed under implication.

LEMMA 7.2. $\Sigma \models \sigma$.

Proof. Let $d = \{f, g_0, ..., g_n, h_0, ..., h_n\}$ be a database satisfying Σ. We show that d satisfies σ. Suppose that

(1) $(a, b, c), (a, b', c') \in f$.

We show that $c = c'$. Using the INDs in Σ, there are a', a'', c_i, c_i', c_i'', and c_i''' $(0 \leqslant i \leqslant n)$ such that:

(2)	$(a, b, c_0), (a, b', c_0') \in g_0$	by α_0 and (1),
(3)	$(b, c_i), (b', c_i') \in g_i$	by α_i and (1), for $1 \leqslant i \leqslant n$,
(4)	$(b, c_i''), (b', c_i''') \in h_i$	by β_i and (1), for $0 \leqslant i < n$,
(5)	$(b, c_n'', c), (b', c_n''', c') \in h_n$	by β_n and (1),
(6)	$(a', b, c_0''), (a'', b', c_0''') \in g_0$	by γ_0' and (4),
(7)	$(b, c_i''), (b', c_i''') \in g_i$	by γ_i' and (4), (5), for $1 \leqslant i \leqslant n$,
(8)	$(b, c_{i-1}''), (b', c_{i-1}''') \in g_i$	by γ_i'' and (4), for $1 \leqslant i \leqslant n$,

Using the FDs in Σ, we have:

(9)	$c_i'' = c_i$	by ε_i and (2), (3), (6), (7), for $0 \leqslant i \leqslant n$,
(10)	$c_i = c_{i-1}''$	by ε_i and (3), (8), for $1 \leqslant i \leqslant n$,
(11)	$c_0 = c_0'$	by δ_0 and (2),
(12)	$c_i' = c_i'''$	by ε_i and (2), (3), (6), (7), for $0 \leqslant i \leqslant n$,
(13)	$c_i''' = c_{i+1}'$	by ε_{i+1} and (3), (8), for $0 \leqslant i < n$.

Hence, we may derive that $c_n'' = c_n'''$ using the sequence of equalities $c_n'' = c_n = c_{n-1}'' = c_{n-1} = \cdots = c_0 = c_0' = c_0''' = c_1' = c_1''' = \cdots = c_n' = c_n'''$. Thus, $c_n'' = c_n'''$. Since also d satisfies θ_n, and since (5) holds, we finally have:

(14) $c = c'$.

Thus, we may conclude that each database satisfying Σ also satisfies σ. Hence, $\Sigma \models \sigma$, as desired. ∎

Lemma 7.2 completes the proof that Γ is not closed under implication. We now show that Γ is closed under k-ary implication. Several lemmas and claims (that we shall prove) will be helpful.

LEMMA 7.3. *$\Sigma \models \phi$ and $\Sigma \models \Gamma$.*

Proof. We first show that $\Sigma \models \phi$. Recall that ϕ was defined above to be $\phi(F) \cup \phi(G_0) \cup \cdots \cup \phi(G_n) \cup \phi(H_0) \cup \cdots \cup \phi(H_n)$. Take γ in ϕ. We must show that $\Sigma \models \gamma$. If γ is in Σ, then it is immediate that $\Sigma \models \gamma$. If γ is $F: A \to C$, then γ is σ, and so $\Sigma \models \gamma$ by Lemma 7.2. If γ is another member of ϕ, then Proposition 4.1 implies that $\Sigma \models \gamma$. For example, if γ is $H_i: B \to C$, for $0 \leqslant i \leqslant n$, then $\Sigma \models \gamma$, because $\{H_i[BC] \subseteq G_i[BC], G_i: B \to C\} \models H_i: B \to C$. (In the case when γ is $F: B \to C$, we first show that $\Sigma \models \phi(H_n) \models H_n: B \to D$, and we then apply Proposition 4.1 by using also the dependency β_n.) So in every case, $\Sigma \models \gamma$. Thus, $\Sigma \models \phi$. We now show that $\Sigma \models \Gamma$. Recall that $\Gamma = \phi^+ \cup \lambda^+ \cup \omega - \{F: A \to C\}$. Take $\gamma \in \Gamma$. We must show that $\Sigma \models \gamma$. If $\gamma \in \phi^+$, then $\Sigma \models \gamma$, since we already showed that $\Sigma \models \phi$. If $\gamma \in \lambda^+$, then $\Sigma \models \gamma$, since $\lambda \subseteq \Sigma$. If $\gamma \in \omega$, then $\Sigma \models \gamma$, since γ is trivial. So in every case, $\Sigma \models \gamma$. This was to be shown. ∎

We wish to show that Γ is closed under k-ary implication. So, let T be a set of at most k members of Γ, and let τ be an FD, IND, or RD. Suppose that $T \models \tau$. We have to show that $\tau \in \Gamma$. We claim that:

Claim 1. $\tau \in \phi^+ \cup \lambda^+ \cup \omega$.

Claim 2. $T \not\models F: A \to C$ (and so τ is not $F: A \to C$).

Combining these two claims, we obtain that $\tau \in \Gamma$. Thus, we need only prove claims (1) and (2) to show that Γ is closed under k-ary implication.

To prove the first claim, we begin by giving a characterization of the logical consequences of Σ.

LEMMA 7.4. *If δ is a* RD, *then $\Sigma \models \delta$ if and only if $\delta \in \omega$.*

Proof. Consider the database of Fig. 7.1. In this database and in subsequent databases we shall exhibit, distinct variables represent distinct entries. For example, in Fig. 7.1, each of a, b, c, and d are distinct. This database satisfies all dependencies in Σ, but no nontrivial RD. Hence, no RD is a logical consequence of Σ, except the trivial ones. ∎

LEMMA 7.5. *If δ is an* FD, *then $\Sigma \models \delta$ if and only if $\delta \in \phi^+$.*

f:

A	B	C
a	b	d

g_0:

A	B	C
a	b	c

g_i:

B	C
b	c

$(1 \leqslant i \leqslant n)$

h_j:

B	C
b	c

$(0 \leqslant j < n)$

h_n:

B	C	D
b	c	d

FIGURE 7.1

Proof. ($\Rightarrow$) Let e be the database of Fig. 7.2. One can verify by inspection that

(1) e satisfies Σ, and

(2) for each FD δ, if e satisfies δ then $\phi \models \delta$.

The second task is facilitated because if δ is of the form $F: X \to Y$, say, it suffices to check that if f satisfies δ then $\phi(F) \models \delta$. We also note that the double-primed letters were included so that no relation satisfies an FD with the empty set as the left-hand side (e.g., the relation f does not satisfy the FD $F: \varnothing \to C$, which says that the C entry of the F relation is a constant).

Now, let δ be an FD such that $\Sigma \models \delta$. Since, by (1), we know that e satisfies Σ, it follows that e must satisfy δ. Therefore, by (2), we obtain that $\phi \models \delta$.

($\Leftarrow$) This is immediate from Lemma 7.3. ∎

LEMMA 7.6. *If δ is an* IND, *then $\Sigma \models \delta$ if and only if $\delta \in \lambda^+$.*

Proof. ($\Rightarrow$) Let e be the database of Fig. 7.3. One can verify by inspection that:

(1) e satisfies Σ, and

(2) for each IND δ, if e satisfies δ then $\lambda \models \delta$.

The second task is facilitated due to a careful choice of the cardinalities of the relations in e and because b_i, c_i occurs only in h_i, g_i and g_{i+1}. Now, let δ be an IND

f:

A	B	C
a	b	d
a	b'	d
a'	b	d
a''	b''	d''

g_0:

A	B	C
a	b	c
a	b'	c
a'	b	c
a''	b''	c''

g_i:

B	C
b	c
b'	c
b''	c''

$(1 \leqslant i < n)$

g_n:

B	C
b	c
b'	c
b''	c''
b'''	c'

h_j:

B	C
b	c
b'	c
b''	c''

$(0 \leqslant j < n)$

h_n:

B	C	D
b	c	d
b'	c	d
b'''	c'	d
b''	c''	d''

FIGURE 7.2

such that $\Sigma \models \delta$. Since, by (1), we know that e satisfies Σ, it follows that e must satisfy δ. Therefore, by (2), we have that $\lambda \models \delta$.

($\Leftarrow$) Follows trivially, since $\lambda \subseteq \Sigma$ by definition. ∎

We now combine Lemmas 7.4–7.6 to prove our first claim.

LEMMA 7.7. *If T is a set of at most k members of Γ, if τ is an* FD, IND, *or* RD *and if $T \models \tau$ then $\tau \in \phi^+ \cup \lambda^+ \cup \omega$.*

Proof. Assume $T \subseteq \Gamma$. Although we have phrased Lemma 7.7 as we have to show that we are proving claim (1), we do not actually need to assume that T has at most k members. Let τ be an FD, IND, or RD. Suppose that $T \models \tau$. Hence $\Gamma \models \tau$, since $T \subseteq \Gamma$. By Lemma 7.3, we know that $\Sigma \models \Gamma$. Since also $\Gamma \models \tau$, it follows that $\Sigma \models \tau$. Now, by Lemmas 7.4–7.6, we obtain that $\tau \in \phi^+ \cup \lambda^+ \cup \omega$. This concludes the proof. ∎

We now show that the second claim holds. The crucial steps of the proof are spelled out in the following two lemmas:

LEMMA 7.8. *Assume that $0 \leqslant j < n$. Then $\phi^+ \cup \lambda^+ \cup \omega - \{F\colon A \to C, F[B] \subseteq H_j[B]\} = (\phi - \{F\colon A \to C\})^+ \cup (\lambda - \{F[B] \subseteq H_j[B]\})^+ \cup \omega$.*

Proof. Let ρ be the left-hand side of the above equality, that is, ρ is $\phi^+ \cup \lambda^+ \cup \omega - \{F\colon A \to C, F[B] \subseteq H_j[B]\}$. Since, by definition, ϕ^+, λ^+ and ω contain only FDs, INDs, and RDs, respectively, we have:

(1) $\rho = (\phi^+ - \{F\colon A \to C\}) \cup (\lambda^+ - \{F[B] \subseteq H_j[B]\}) \cup \omega.$

We now show that

(2) $(\phi^+ - \{F\colon A \to C\}) = (\phi - \{F\colon A \to C\})^+.$

By definition of ϕ, and since a set of FDs over a relation scheme R can imply another FD only over the same R, it suffices to show that

(3) $(\phi(F)^+ - \{F\colon A \to C\}) = (\phi(F) - \{F\colon A \to C\})^+.$

f:

A	B	C
a	b	d

g_0:

A	B	C
a	b	c
a_0	b_0	c_0
a_0'	b_0'	c_0'

g_i:

B	C
b	c
b_i	c_i
b_{i-1}	c_{i-1}

$(1 \leqslant i \leqslant n)$

h_j:

B	C
b	c
b_j	c_j

$(0 \leqslant j < n)$

h_n:

B	C	D
b	c	d
b_n	c_n	d_n

FIGURE 7.3

The right-hand side is simply $\{F: B \to C\}^+$. It is straightforward to verify that the left-hand side also equals this same value $\{F: B \to C\}^+$. Hence (3), and thus (2), follows.

We now show that

$$(4)\quad (\lambda^+ - \{F[B] \subseteq H_j[B]\}) = (\lambda - \{F[B] \subseteq H_j[B]\})^+.$$

We first show that the right-hand side of (4) is a subset of the left-hand side, that is, that

$$(5)\quad (\lambda^+ - \{F[B] \subseteq H_j[B]\}) \supseteq (\lambda - \{F[B] \subseteq H_j[B]\})^+.$$

To show (5), we first show

$$(6)\quad F[B] \subseteq H_j[B] \notin (\lambda - \{F[B] \subseteq H_j[B]\})^+.$$

To prove (6), it suffices to consider the database of Fig. 7.4, which satisfies $(\lambda - \{F[B] \subseteq H_j[B]\})$, but not $F[B] \subseteq H_j[B]$. This proves (6). Now, we trivially have that

$$(7)\quad \lambda^+ \supseteq (\lambda - \{F[B] \subseteq H_j[B]\})^+.$$

Now (5) follows immediately from (6) and (7). We now prove that the left-hand side of (4) is contained in the right-hand side, that is,

$$(8)\quad (\lambda^+ - \{F[B] \subseteq H_j[B]\}) \subseteq (\lambda - \{F[B] \subseteq H_j[B]\})^+.$$

Let γ be an IND $R[X] \subseteq S[Y]$ in the left-hand side of (8). Thus, $\gamma \in \lambda^+$, and γ is not the IND $F[B] \subseteq H_j[B]$. We must show that γ is in the right-hand side of (8). Assume that $|X| = m$ (and hence $|Y| = m$). By Corollary 3.2, there is a sequence $S_1[X_1], S_2[X_2], \ldots, S_w[X_w]$, where

(i) S_i is the name of one of the relation schemes, for $1 \leqslant i \leqslant w$;

(ii) X_i is a sequence of m distinct attributes of S_i, for $1 \leqslant i \leqslant w$;

(iii) the first member $S_1[X_1]$ of the sequence is $R[X]$;

(iv) the last member $S_w[X_w]$ of the sequence is $S[Y]$; and

(v) the IND $S_i[X_i] \subseteq S_{i+1}[X_{i+1}]$ can be obtained from a member σ_i of λ by IND2 (projection and permutation), for $1 \leqslant i < w$.

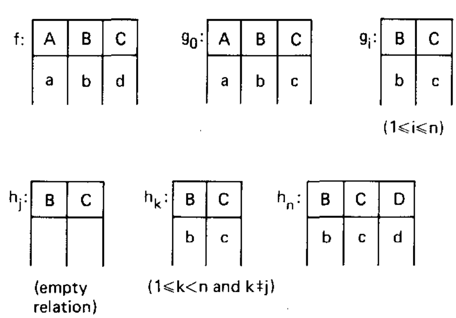

FIGURE 7.4

We can assume that w is as small as possible so that (i)–(v) hold. If $w = 1$, then γ is a trivial IND, which is therefore in the right-hand side of (8), which was to be shown. So, assume that $w > 1$.

If no σ_i (from step (v)) is the IND $F[B] \subseteq H_j[B]$, then γ is in the right-hand side of (8), and so (8) is proven. Therefore, assume that σ_k is the IND $F[B] \subseteq H_j[B]$, for some k $(1 \leqslant k < w)$. Then $S_k[X_k]$ is $F[B]$. If $k \neq 1$, then σ_{k-1} is an IND in λ with right-hand side $F[Z]$, for some Z. But there is no such IND in λ. Hence, $k = 1$. This tells us that the left-hand side $R[X]$ of γ is $F[B]$, that $S_1[X_1]$ is $F[B]$, and that $S_2[X_2]$ is $H_j[B]$. If $w = 2$, then γ would be the IND $F[B] \subseteq H_j[B]$, which we know it is not. Thus, $w > 2$. Since $S_2[X_2]$ is $H_j[B]$, we see by examining λ that $S_3[X_3]$ is either $G_j[B]$ or $G_{j+1}[B]$. But in either case, the IND $S_1[X_1] \subseteq S_3[X_3]$ can be obtained from one of $\alpha_0, \ldots, \alpha_n$ by IND2 (projection and permutation). Thus, $S_2[X_2]$ is unnecessary, that is, the sequence $S_1[X_1]$, $S_3[X_3], \ldots, S_w[X_w]$ obeys (i)–(v). This violates minimality of w. So, (8) follows. Then (4) follows from (5) and (8).

Finally, from (1), (2), and (4), we obtain

$$\rho = (\phi - \{F: A \to C\})^+ \cup (\lambda - \{F[B] \subseteq H_j[B]\})^+ \cup \omega,$$

which proves the lemma. ∎

LEMMA 7.9. *If T is a set of at most k members of Γ, then $T \not\models F: A \to C$.*

Proof. Let $T \subseteq \Gamma$ be such that $|T| \leqslant k$. Since each of the n dependencies $F[B] \subseteq H_i[B]$ is in Γ, for $0 \leqslant i < n$, and since $|T| \leqslant k < n$, it follows that T does not contain $F[B] \subseteq H_j[B]$, for some j $(0 \leqslant j < n)$. Hence, we have that $T \subseteq \rho$, where, as before,

$$\rho = \phi^+ \cup \lambda^+ \cup \omega - \{F: A \to C, R[B] \subseteq H_j[B]\}.$$

By Lemma 7.8, we obtain

$$(9) \quad \rho = (\phi - \{F: A \to C\})^+ \cup (\lambda - \{F[B] \subseteq H_j[B]\})^+ \cup \omega.$$

Let e be the database of Fig. 7.5. One can verify by inspection that e satisfies $\phi - \{F: A \to C\}$ and that e satisfies $\lambda - \{F[B] \subseteq H_j[B]\}$. Also, e satisfies ω, since ω is a

f:

A	B	C
a	b	d
a	b'	d'

g_0:

A	B	C
a	b	c
a	b'	c

g_k:

B	C
b	c
b'	c

$(1 \leqslant k \leqslant j)$

g_k:

B	C
b	c
b'	c'

$(j < k \leqslant n)$

h_k:

B	C
b	c
b'	c

$(0 \leqslant k < j)$

h_j:

B	C

(empty relation)

h_k:

B	C
b	c
b'	c'

$(j < k \leqslant n)$

h_n:

B	C	D
b	c	d
b'	c'	d'

FIGURE 7.5

set of trivial dependencies. Thus, by (9), it follows that e satisfies ρ. Since $T \subseteq \rho$, it follows that e also satisfies T. But $F: A \to C$ is false in e. Therefore $T \not\models F: A \to C$. ▮

LEMMA 7.10. *If T is a set of at most k members of Γ, if τ is an* FD, IND, *or* RD, *and if $T \models \tau$, then $\tau \in \Gamma$.*

Proof. Let T be a set of at most k members of Γ. Let τ be an FD, IND, or RD. Suppose that $T \models \tau$. By Lemmas 7.7 and 7.9, we know that $\tau \in \phi^+ \cup \lambda^+ \cup \omega - \{F: A \to C\}$, that is, $\tau \in \Gamma$. ▮

Thus, Γ is closed under k-ary implication, which was to be shown. This completes the proof of part (ii) of the theorem. The proof of part (i) is exactly the same, except that Lemma 7.4 is superfluous. ▮

We close this section by observing that nowhere in the proof of Theorem 7.1 did we use infinite databases. Therefore, Theorem 7.1 applies also to the case of finite implication. We also note that we did not see how to use Corollary 5.2 to prove Theorem 7.1. Instead, we found it necessary to use the more general Theorem 5.1. In fact, this led us to state Theorem 5.1 in its full generality.

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 does there exist 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, Ri]. (Beeri and Vardi [BV3] have given a complete axiomatization in the Gentzen style for *full* join dependencies; this axiomatization is not k-ary for any k.)

We have also shown that the decision problem for INDs is PSPACE-complete. Thus, there is no polynomial-time decision procedure (unless P = PSPACE).

From the point of view of practical database design, the results in this paper can be evaluated as follows. On the one hand, INDs are a valuable tool in database design, as we discussed in the Introduction. However, our results show that INDs have a computationally hard decision problem (provided P $\neq$ PSPACE), which implies that it might be useful to consider restricted forms of inclusion dependencies, with an easier decision problem. For example, if we restrict attention to inclusion dependencies that are k-ary or less, for fixed k, then the decision problem is solvable in polynomial time.

Acknowledgments

The authors are grateful to David Johnson and Moshe Vardi, each of whom carefully read the paper and made helpful comments, and to Nick Pippenger for bringing Landau's result (mentioned in Section 3) to our attention.

References

[AV] S. Abiteboul and M. Y. Vardi, "Formal Systems for Untyped Dependencies," to appear.

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

[Ar] W. W. Armstrong, Dependency structures of database relationships, *in* "Proceedings, IFIP 74," pp. 580–583, (1974).

[BB] C. Beeri and P. A. Bernstein, Computational problems related to the design of normal form relational schemas, *ACM Trans. Database Systems* **4** (1) (1979), 30–59.

[BFH] C. Beeri, R. Fagin, and J. H. Howard, A complete axiomatization for functional and multivalued dependencies in database relations, *in* "Proceedings, 1977 ACM SIGMOD Conf.," Toronto, pp. 47–61.

[BK] C. Beeri and H. F. Korth, Compatible attributes in a universal relation, *in* "Proceedings, First ACM SIGACT–SIGMOD Symp. on Principles of Database Systems," Los Angeles, pp. 55–62, 1982.

[BV1] C. Beeri and M. Y. Vardi, "A Proof Procedure for Data Dependencies," Hebrew Univ. of Jerusalem Technical Report, December 1980.

[BV2] C. Beeri and M. Y. Vardi, "Formal Systems for Tuple and Equality Generating Dependencies," Hebrew Univ. of Jerusalem Technical Report, April 1981; *SIAM J. Comput.*, in press.

[BV3] C. Beeri and M. Y. Vardi, "Formal Systems for Join Dependencies," *Theoretical Computer Science*, in press.

[BG] P. A. Bernstein and N. Goodman, What does Boyce–Codd normal form do? *in* "Proceedings, 6th Int. Conf. on Very Large Databases," pp. 245–269, 1980.

[CLM] A. K. Chandra, H. R. Lewis, and J. A. Makowsky, Embedded implicational dependencies and their decision problem, *in* "Proceedings, 1981 ACM SIGACT Symp. on Theory of Computation," Milwaukee, 342–354.

[CV] A. K. Chandra and M. Y. Vardi, "The Implication Problem for Functional and Inclusion Dependencies is Undecidable," IBM Research Report RC 9980, Yorktown, N.Y., 1983.

[Ch] P. Chen, The entity-relationship model—Toward a unified view of data, *ACM Trans. Database Systems* **1** (1) (1976), 9–36.

[Co1] E. F. Codd, Relational completeness of database sublanguages, *in* "Data Base Systems," (R. Rustin, Ed.), Prentice–Hall, N.J., 1972.

[Co2] E. F. Codd, Further normalization of the database relational model, *in* "Data Base Systems" (R. Rustin, Ed.), Prentice–Hall, N.J., 1972.

[Co3] E. F. Codd, Extending the database relational model to capture more meaning, *ACM Trans. Database Systems* **4** (4) (1980), 397–434.

[KCV] P. C. Kannelakis, S. S. Cosmadakis, and M. Y. Vardi, Unary inclusion dependencies have polynomial-time inference problems *in* "Proceedings, 1983 ACM SIGACT Symp. on Theory of Computation," Boston, 264–277.

[Da] C. J. Date, Referential integrity, *in* "Proceedings, 1981 Very Large Data Bases Conf.," pp. 2–12.

[DG] B. S. Dreben and W. D. Goldfarb, "The Decision Problem: Solvable Classes of Quantificational Formulas," Addison–Wesley, Reading, Mass., 1979.

[Fa1] R. Fagin, Multivalued dependencies and a new norma form for relational databases, *ACM Trans. Database Systems* **2** (3) (1977), 262–278.

[Fa2] R. Fagin, Functional dependencies in a relational database and propositional logic, *IBM J. Res. Develop.* **21** (6) (1977), 534–544.

[Fa3] R. Fagin, A normal form for relational databases that is based on domains and keys, *ACM Trans. Database Systems* **6** (3) (1981), 387–415.

[Fa4] R. Fagin, Horn clauses and database dependencies, *J. Assoc. Comput. Mach.* **29** (4) (1982), 952–985.

[FMUY] R. Fagin, D. Maier, J. D. Ullman, and M. Yannakakis, Tools for template dependencies, *SIAM J. Comput.* **12** (1) (1983), 36–59.

[FV] R. Fagin and M. Y. Vardi, Armstrong databases for functional and inclusion dependencies, *Inform. Process. Lett.* **16** (1983), 13–19.

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

[JK] D. S. Johnson and A. Klug, Testing containment of conjunctive queries under functional and inclusion dependencies, *in* "Proceedings, First ACM SIGACT–SIGMOD Symp. on Principles of Database Systems," Los Angeles, pp. 164–169, 1982.

[Ke] W. Kent, Consequences of assuming a universal relation, *ACM Trans. Database Systems* **6** (4) (1981), 539–556.

[Kl] A. Klug, Entity-relationship views over uninterpreted enterprise schemas, *in* "Proceedings, Int. Conf. Entity-Relationship Approach to Systems Analysis and Design," pp. 52–72, 1979.

[La] E. Landau, "Handbuch der Lehre von der Verteilung der Primzahlen," Leipzig, 1909. 3rd ed., Chelsea, New York, 1974.

[LP] H. R. Lewis and C. Papadimitriou, "Elements of the Theory of Computation," Prentice–Hall, Englewood Cliffs, N.J., 1981.

[Li] S. H. Lin, "Existential Dependencies in Relational Databases," Ph.D. Thesis, UCLA, Los Angeles, 1981.

[Mi1] J. C. Mitchell, Inference rules for functional and inclusion dependencies, *in* "Proceedings, 2nd ACM SIGACT–SIGMOD Symp. on Principles of Database Systems," Atlanta, pp. 58–69, 1983.

[Mi2] J. C. Mitchell, The implication problem for functional and inclusion dependencies, *Inform. and Control*, in press.

[Ri] J. Rissanen, Theory of relations for databases—A tutorial survey, *in* "Proceedings, 7th Symp. on Math. Found. of Comp. Science," Lecture Notes in Comp. Science, No. 64, pp. 537–551, Springer-Verlag, Berlin, 1978.

[Sa] W. J. Savitch, Relationship between nondeterministic and deterministic tape complexities, *J. Comput. System Sci.* **4** (1970), 177–192.

[Sc] E. Sciore, Inclusion dependencies and the universal instance, *in* "Proceedings, 2nd ACM SIGACT–SIGMOD Symp. on Principles of Database Systems," Atlanta, pp. 48–57, 1983.

[SU] F. Sadri and J. D. Ullman, A complete axiomatization for a large class of dependencies in relational databases, *J. Assoc. Comput. Mach.* **29** (2) (1981), 363–372.

[SW] Y. Sagiv and S. Walecka, Subset dependencies and a completeness result for a subclass of embedded multivalued dependencies, *J. Assoc. Comput. Mach.* **29** (1) (1982), 103–117.

[SS] J. M. Smith and D. C. P. Smith, Database abstractions: Aggregation, *Comm. ACM* **20** (6) (1977), 405–413.

[Ul] J. D. Ullman, The U.R. strikes back, *in* "Proceedings, First ACM SIGACT–SIGMOD Symp. on Principles of Database Systems," Los Angeles, pp. 10–22, 1982.

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

[YP] M. Yannakakis and C. Papadimitriou, "Algebraic Dependencies," *J. Comput. System Sci.* **25** (1982), 2–41.

[Za] C. Zaniolo, Design of relational views over network schemas, *in* "Proceedings, 1979 ACM SIGMOD Conf.," Boston, pp. 179–190.