RJ3440 (40926) 4/5/82
Computer Science

Research Report

ARMSTRONG DATABASES

Ronald Fagin

IBM Research Laboratory
San Jose, California 95193

Appeared in: 7th IBM Symposium on Mathematical Foundations of Computer Science, Kanagawa, Japan, May 1982.

IBM

Research Division
Yorktown Heights, New York • San Jose, California • Zurich, Switzerland

Copies may be requested from:
IBM Thomas J. Watson Research Center
Distribution Services
Post Office Box 218
Yorktown Heights, New York 10598

RJ3440 (40926) 4/5/82
Computer Science

ARMSTRONG DATABASES

Ronald Fagin

IBM Research Laboratory
San Jose, California 95193

ABSTRACT: An Armstrong database is a database that obeys precisely a given set of sentences (and their logical consequences) and no other sentences of a given type. This paper surveys history and results on Armstrong databases.

Key words and phrases: Armstrong relation, Armstrong database, relational database, implicational dependency, embedded dependency, functional dependency, multivalued dependency, join dependency, template dependency, direct product, faithfulness, logical consequence, mathematical logic.

CR categories: 4.33, 5.21

Appeared in: 7th IBM Symposium on Mathematical Foundations of Computer Science, Kanagawa, Japan, May 1982.

1. INTRODUCTION

We begin by discussing Armstrong relations, which are special cases of Armstrong databases (where the database consists of a single relation.) For simplicity, we further restrict our attention initially by considering only functional dependencies, or FD's [Co].

We need some basic definitions. We assume a finite set U of *attributes*. A tuple (over U) is a mapping with domain U, and a *relation* (over U) is a set of tuples (over U). If $X \subseteq U$, and if t is a tuple over U, then we denote the restriction of t to X by t[X]. If r is a relation over U, then $r[X]=\{t[X]: t \in r\}$. If A is an attribute of U, and if t is a tuple over U, then we may refer to t[A] as an *entry*, in the A *column*.

A *functional dependency (over U)* [Co], or an *FD*, is a statement, or sentence, $X \to Y$ where X, $Y \subseteq U$. A relation r over U *obeys* the FD $X \to Y$ if wherever t_1, t_2 are tuples of r with $t_1[X]=t_2[X]$, then $t_1[Y]=t_2[Y]$. We also say then that the FD *holds for* r. If the FD does not hold for r, then we say that the FD *fails* in r, or that r *violates* the FD. (Similar comments apply to other sentences besides FD's.)

Let Σ be a set of sentences, such as FD's, let σ be a single sentence. When we say that Σ *logically implies* σ or that σ is a *logical consequence* of Σ, we mean that whenever every sentence in Σ holds for a relation r, then also σ holds for r. That is, there is no "counterexample relation" or "witness" r such that every sentence in Σ holds for r, but such that σ fails in r. We write $\Sigma \models \sigma$ to mean that Σ logically implies σ (and we write $\Sigma \not\models \sigma$ to mean that Σ does not logically imply σ.) For example, $\{A \to B, B \to C\} \models A \to C$. Let Σ be a set of FD's, and let Σ^* be the set of all FD's that are logical consequences of Σ. For each FD σ not in Σ^*, we know (by definition of $\models$) that there is a relation r_σ (a witness) such that r_σ obeys Σ but not σ. An *Armstrong relation* for Σ is a relation (a global witness) that can simultaneously serve the role of all of the r_σ's. That is, an Armstrong relation is a relation that obeys Σ^* and *no other FD's*.

As an example [Fa4], let Σ be the set {EMP→DEPT, DEPT→MGR}, containing two FD's. Then Σ^* contains the FD's in Σ, along with, for example, the FD EMP→MGR. It is easy to verify (by considering all possible FD's involving only EMP, DEPT, and MGR) that the relation (call it r) in Figure 1.1 is an Armstrong relation for Σ, that is, that it obeys every FD in Σ^* and no others. For example, the FD MGR→DEPT is not an FD in Σ^*, and indeed, r does *not* obey this FD, since Gauss is the manager of two distinct departments (Math and Physics).

A closely related concept to Armstrong relations in traditional mathematics is the free algebra with countably many generators [Gr], which obeys just a specified set of equations and their logical consequences, and no other equations. (However, although the free algebra just mentioned is unique to within isomorphism, Armstrong relations are not [Fa4].) In ordinary first-order logic (where arbitrary first-order sentences, and not just FD's are allowed), there can be no Armstrong relations. For example, let Σ be the empty set $\emptyset$. Assume that r were a relation that obeyed just Σ^* (that is, just the tautologies), and no other first-order sentences. Let σ be an arbitrary first-order sentence such that neither σ nor $\neg\sigma$ is a tautology. Clearly, r must obey one of σ or $\neg\sigma$; thus, r obeys a nontautology. This is a contradiction. Thus, there is a witness for σ (a relation that shows that σ is not a tautology), and a witness for $\neg\sigma$ (a relation that shows that $\neg\sigma$ is not a tautology), but there is no global witness (a relation that simultaneously shows that σ is not a tautology and that $\neg\sigma$ is not a tautology.)

It is common to speak of a relation obeying an "accidental" dependency, that is, a dependency that is not a logical consequence of the collection of "specified" dependencies. Thus, each specified dependency is supposed to hold "for all time," that is, for every "snapshot" (instance) of the database, whereas an accidental dependency is one that happens to hold in some snapshot of the database, but may fail in other snapshots. An Armstrong relation is precisely one that obeys every specified dependency and no accidental dependency.

In Section 2, we present some applications of Armstrong databases. In Section 3, we describe their history, and in Section 4, we discuss techniques for constructing them. In Section 5, we discuss the size of Armstrong relations (for sets of functional dependencies). In Section 6, we discuss the issue of finite versus infinite Armstrong relations. In Section 7, we present conclusions.

2. Applications of Armstrong databases

We begin with an interesting "practical" application for Armstrong relations. Silva and Melkanoff [SM] have developed a database design aid, in which the database designer inputs a set of FD's and MVD's (multivalued dependencies) [Fa2]. The design aid then presents him with an Armstrong relation, that is, a "sample relation" that obeys just those dependencies that are logical consequences of those that he has inputted. (Armstrong relations exist in the presence of FD's and MVD's, and this is the case in which Silva and Melkanoff were interested.) Let us say, for example, that the designer gives as input the set {EMP→DEPT, DEPT→MGR} of FD's. The database design aid would then present the designer with an Armstrong relation, such as relation r in Figure 1.1, for this set of dependencies. The designer would then inspect the sample relation, and might observe, for example, "Here is a manager, namely Gauss, who manages two different departments. *Therefore*, the dependencies that I inputted must not have implied that no manager can manage two different departments. Since I want this to be a constraint for my database, I'd better input the FD MGR→DEPT".

In this example, the designer did not have to explicitly think about the dependency MGR$\rightarrow$DEPT and whether or not it was a consequence of the dependencies that he input; rather, by seeing the Armstrong relation, and thinking about what it said, he simply *noticed* that the FD MGR$\rightarrow$DEPT failed. Thus, Silva and Melkanoff's approach is a partial solution, in the spirit of Query-by-Example [Zl], to the problem of helping a designer think of what dependencies should be included.

We now mention two theoretical applications of Armstrong databases that are in the literature. A *key* of a relation is a set K of attributes such that the FD K$\rightarrow$U holds in the relation but such that for every proper subset K$'$ of K, the FD K$'\rightarrow$U does not hold in it. A key gives a minimal unique identifier for each tuple in a relation. Beeri, Dowd, Fagin, and Statman [BDFS] use Armstrong relations to generalize a result, obtained by Demetrovics [De] using quite complicated methods, about the possible sets of keys for a relation. Specifically, they show that that if J is an arbitrary nonempty collection of incomparable subsets of a finite set U, then there is a relation with attributes U for which the set of keys is precisely J (Demetrovics proved that there is such a relation in the special case where J is the set of all subsets of U that contain precisely $\lfloor n/2 \rfloor$ members, where $\lfloor x \rfloor$ is the greatest integer not exceeding x.)

Finally, Casanova, Fagin, and Papadimitriou [CFP] make use of an Armstrong database to help show that for each k, there is no k-ary complete axiomatization for functional dependencies and inclusion dependencies together (we shall discuss inclusion dependencies later.)

3. History

In 1974, Armstrong wrote one of the first papers [Ar] in database theory. In it, he presented a set of deduction rules (commonly called *Armstrong's axioms*) for FD's. The following is a set of deduction rules equivalent to those of Armstrong:

Arm1 (reflexivity): If $Y \subseteq X \subseteq U$, then $X \rightarrow Y$.
Arm2 (augmentation): If $X \rightarrow Y$ and $Z \subseteq U$, then $XZ \rightarrow YZ$.
Arm3 (transitivity): If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$.

By XZ in (Arm2) above, we mean $X \cup Z$, and similarly for YZ.

We now discuss the concept of a *proof*. Let Σ be a set of FD's, and let σ be a single FD. A *proof* of σ from Σ is a finite sequence of FD's, where (1) each FD in the sequence is either a member of Σ, or else follows from previous FD's in the sequence by an application of the deduction rules (Arm1)-(Arm3), and where (2) σ is the last FD in the sequence. We write $\Sigma \vdash \sigma$ to mean that there is a proof of σ from Σ. If $\Sigma \vdash \sigma$, then we may say that σ is *provable* from Σ. Let us denote by Σ^+ the set of FD's that are provable from Σ. Thus, $\Sigma^+ = \{\sigma: \Sigma \vdash \sigma\}$. By contrast, recall that $\Sigma^* = \{\sigma: \Sigma \models \sigma\}$.

The main result in Armstrong's paper is the following.

Armstrong's Theorem [Ar]. For each set Σ of FD's, there is a relation that obeys precisely the FD's in Σ^+.

Armstrong proved this result by explicitly constructing a rather complicated relation with the desired property.

For the sake of later discussion, we now state a possible generalization of Armstrong's Theorem. We assume (a) a set $\mathscr{S}$ of sentences and (b) a set of deduction rules for sentences in $\mathscr{S}$. In the case of Armstrong's Theorem, $\mathscr{S}$ is the (finite) set of all FD's over attributes U. Later, we shall consider cases in which $\mathscr{S}$ is infinite. We define *proof* (and *provable*) as before. We assume that the deduction rules are *sound*, that is, we assume that if σ is provable from Σ using the deduction rules, then σ is a logical consequence of Σ. As before, if $\Sigma \subseteq \mathscr{S}$ and if $\sigma \in \mathscr{S}$, then we write $\Sigma \vdash \sigma$ if there is a proof of σ from Σ using the deduction rules, and we define Σ^+ to be the set of members σ of $\mathscr{S}$ such that $\Sigma \vdash \sigma$. A possible generalization of Armstrong's Theorem is:

(1) For each set Σ of members of $\mathscr{S}$, there is a relation that obeys precisely the members of $\mathscr{S}$ in Σ^+.

If $\mathscr{S}$ is a set of sentences, and if $\Sigma \subseteq \mathscr{S}$, then Σ^* is the set of all sentences σ in $\mathscr{S}$ such that $\Sigma \models \sigma$. An *Armstrong relation* for Σ (with respect to $\mathscr{S}$) is a relation that obeys every member of Σ^* and no other sentence in $\mathscr{S}$. For later reference, we record the following possible statement:

(2) For each set Σ of members of $\mathscr{S}$, there is a relation that obeys precisely the members of $\mathscr{S}$ in Σ^*.

Statement (2) says precisely that each subset Σ of $\mathscr{S}$ has an Armstrong relation (with respect to $\mathscr{S}$.) If (2) holds, then we may say that $\mathscr{S}$ *enjoys Armstrong relations*. It is easy to see that if $\mathscr{T} \subseteq \mathscr{S}$, and if $\mathscr{S}$ enjoys Armstrong relations, then so does $\mathscr{T}$.

The soundness of a set of deduction rules for sentences in $\mathscr{S}$ says that $\Sigma^+ \subseteq \Sigma^*$. Consider the following statement:

(3) For each set Σ of members of $\mathscr{S}$, necessarily $\Sigma^+ = \Sigma^*$.

Statement (3) says precisely that the deduction rules are *complete*, that is, that a member of $\mathscr{S}$ is provable from Σ if and only if it is a logical consequence of Σ.

Proposition 3.1. Let $\mathscr{S}$ be fixed. Then (1) is equivalent to (2) and (3) together.

Proof: It is obvious that (2) and (3) together imply (1). Conversely, assume that (1) holds. We shall soon show that (3) holds. Since (1) and (3) together clearly imply (2), it follows that (2) holds. So, we need only show that (3) holds. We already noted that the inclusion $\Sigma^+ \subseteq \Sigma^*$ follows from soundness of the deduction rules. Let us now consider the opposite inclusion $\Sigma^* \subseteq \Sigma^+$. Assume that $\sigma \in \Sigma^*$; we must show that $\sigma \in \Sigma^+$. Let r be a relation, guaranteed to exist by (1), that obeys precisely the members of $\mathscr{S}$ in Σ^+. Then r obeys Σ, since $\Sigma \subseteq \Sigma^+$, by our definition of a proof (in fact, for each member of Σ there is a "one-line" proof.) Now $\Sigma \models \sigma$, since $\sigma \in \Sigma^*$. Since also r obeys Σ, it follows that r obeys σ. Since r obeys precisely the members of $\mathscr{S}$ in Σ^+, and since r obeys σ, it follows that $\sigma \in \Sigma^+$. This was to be shown. □

The proof we just presented was given by Fagin [Fa3] in 1977 (in the special case where $\mathscr{S}$ is the set of FD's over a given set U of attributes.)

Also in 1977, Beeri, Fagin, and Howard [BFH] gave a set of deduction rules for FD's and MVD's together. Let $\mathscr{S}$ be the set of FD's and MVD's (over a given set of attributes.) They proved (3), that is, completeness of their rules, and they also proved (1). They called (1) *strong completeness*.

In 1980, Fagin [Fa4] defined the concept of an Armstrong relation, based on (2). Thus, property (1) above, the concept of strong completeness (where Armstrong's Theorem is the special case for FD's) has been "decomposed" into two orthogonal concepts: (a) Armstrong relations (as in property (2) above), and (b) completeness (property (3) above). Completeness is a proof-theoretic concept, that deals with the power of some set of deduction rules. However, the Armstrong relation concept has nothing to do with deduction rules, but is instead a pure model-theoretic concept.

4. Techniques for constructing Armstrong databases

In this section, we discuss various techniques for constructing Armstrong databases (and the limitations of these techniques).

a. Disjoint union

This approach was first suggested by Beeri, Fagin, and Howard [BFH]. We begin by discussing it in the context of FD's only.

The *disjoint union* of a collection of relations (all with the same attributes) is obtained by first replacing each relation by an isomorphic copy, in such a way that no entry in one relation equals any entry in any of the other relations; then a new relation is formed by taking the union of all of the tuples in all of the relations.

Let Σ be a set of FD's (over a set U of attributes), and let $\sigma_1, ..., \sigma_k$ be those FD's (over U) that are *not* logical consequences of Σ. By definition of logical consequence, there are relations $r_1, ..., r_k$, where r_i obeys Σ but not σ_i $(1 \leq i \leq k)$. Let r be the disjoint union of $r_1, ..., r_k$. We now show that the relation r we have just formed is "almost" an Armstrong relation for Σ. Since a subset of the tuples of r (namely, the isomorphic copy of r_i) violates σ_i, it follows easily that r violates σ_i $(1 \leq i \leq k)$. If r were to obey every member of Σ, then it would follow immediately that r would be an Armstrong relation for Σ. Although r does not necessarily obey every member of Σ, something almost as strong is true. Let us call an FD $\emptyset \rightarrow Y$, in which the "left-hand side" is the empty set, a *nonstandard* FD (and other FD's *standard* FD's.) Let $X \rightarrow Y$ be a standard FD in Σ. We now show that r obeys $X \rightarrow Y$. Thus, r obeys every standard FD in Σ. Let t_1 and t_2 be two tuples of r such that $t_1[X] = t_2[X]$; we must show that $t_1[Y] = t_2[Y]$. Since $t_1[X] = t_2[X]$, we know (by disjointness) then t_1 and t_2 are in (the isomorphic copy of) r_i for some i. Since r_i obeys the FD $X \rightarrow Y$, it follows that $t_1[Y] = t_2[Y]$. This was to be shown.

The proof we just presented shows that if $\mathscr{S}$ is the set of all standard FD's over U, then (2) above holds. This proof is due to Beeri, Fagin, and Howard [BFH], who neglected to "patch" the proof to deal with nonstandard FD's. There is a fairly simple patch ([BDFS]; see also [AD].)

The proof we just gave (with a slightly more complicated patch) can be used to show that FD's and MVD's enjoy Armstrong relations (that is, if $\mathscr{S}$ is the set of all FD's and MVD's over a given set U of attributes, then (2) above holds.) In fact, this was the case of interest to Beeri, Fagin, and Howard. Beeri [Be] showed that with an even more complicated patch, the proof can be made to work for FD's and join dependencies [Ri]. Unfortunately, it does not seem that this technique can be pushed much further. In particular, there does not seem to be a way to make such a proof work for "embedded" dependencies, such as as embedded MVD's [Fa2].

b. Agreement sets

We now describe a characterization by Beeri, Dowd, Fagin, and Statman [BDFS] of Armstrong relations for FD's, and we show how they use their characterization to construct Armstrong relations.

Let Σ be a set of FD's, over the set U of attributes. A subset $V \subseteq U$ is *closed* if for every FD $X \rightarrow Y$ in Σ for which $X \subseteq V$, also $Y \subseteq V$. It is easy to see [Ar] that the intersection of closed sets is closed. Note that the minimal closed set containing X is X^*, where X^* is the set of all attributes A such that $\Sigma \models X \rightarrow A$.

Let M be a family of subsets of a finite set, closed under intersection. Then M contains a unique minimal subfamily M′ such that the members of M′ generate M by intersection [AD]. Thus, M′ is the smallest set such that $M = \{S_1 \cap \ldots \cap S_k\colon k \geq 0 \text{ and } S_1, \ldots, S_k \in M'\}$. The members of M′ are the *intersection generators* of M. In fact, it is not hard to see that a member V of M is in M′ if and only if V is properly contained in the intersection of the members of M that properly contain V. For a given set of Σ of FD's, denote by $CL(\Sigma)$ the family of closed sets defined by Σ. As we noted, $CL(\Sigma)$ is closed under intersection. Denote by $GEN(\Sigma)$ the intersection generators of $CL(\Sigma)$. Note that U is in $CL(\Sigma)$ but not in $GEN(\Sigma)$, since by convention, U is the intersection of the empty collection of sets.

Let t_1 and t_2 be tuples, and let X be a set of attributes. We say that t_1 and t_2 *agree exactly on X* if $t_1[X]=t_2[X]$, and if $t_1[A]\neq t_2[A]$ for each attribute A not in X. If r is a relation, then define agr(r) to be {X: there is a pair of distinct tuples in r that agree exactly on X}. The following characterization of Armstrong relations for sets of FD's is quite useful.

Theorem 4.1 [BDFS]. Let Σ be a set of FD's and let r be a relation. Then r is an Armstrong relation for Σ if and only if $GEN(\Sigma) \subseteq agr(r) \subseteq CL(\Sigma)$.

Theorem 4.1 can be utilized [BDFS] to give an algorithm for obtaining an Armstrong relation, given a set Σ of FD's. The construction is very similar to that of Gold [Go]. Let n be the number of attributes. The algorithm first cycles through each of the 2^n subsets of attributes; and checks which are closed (with respect to Σ). Let S be the collection $CL(\Sigma)$ of closed sets. (We could get away with using $GEN(\Sigma)$ instead of $CL(\Sigma)$ as S in the construction that follows, but we do not wish to spend the time to prune out the nongenerators.) Assume that the distinct members of S are $S_1, \ldots, S_r$. Let t_i $(1 \leq i \leq r)$ be a tuple where $t[A]=0$ if A is an attribute in S_i, and where $t[A]=i$ for each of the other attributes. The desired relation contains a tuple of all 0's, along with each of the tuples t_i $(1 \leq i \leq r)$. By Theorem 4.1, it follows easily that as long as $GEN(\Sigma) \subseteq S \subseteq CL(\Sigma)$, this construction produces an Armstrong relation for Σ.

It is clear that this algorithm has an exponential running time (exponential in the number of attributes), since the size of Σ is at most exponential in the number of attributes, and checking whether a set X is closed can be done in time linear in the size of Σ and the set X [BB]. There is no algorithm with faster than an exponential running time, since [BDFS] there is a set of FD's such that the number of tuples in the smallest Armstrong relation for Σ is exponential in the number of attributes, and so an exponential amount of time is required just to write down the relation. (The formal statement of this result on the worst-case size of an Armstrong relation appears in Section 5 below.)

c. Direct products

Let $<r_i: i \in I>$ be a (finite or infinite) family of relations, each with the same set U of attributes. We now define the *direct product* $\otimes<r_i: i \in I>$. The direct product has the same set U of attributes as does each of the r_i's. In particular, the direct product maps a family of d-ary relations into a d-ary relation (with the same arity d as each of the r_i's.) For notational convenience, let us assume that U contains precisely three attributes ABC. (It is obvious how to generalize the definition from this special case.) The tuple $(<a_i: i \in I>, <b_i: i \in I>, <c_i: i \in I>)$ is a tuple of the direct product if and only if (a_i, b_i, c_i) is a tuple of r_i, for every i. For example, the direct product of the first two relations in Figure 4.1 is the third relation in Figure 4.1.

Fagin [Fa4] defined a class of sentences, which he called *embedded implicational dependencies* (or *EID*'s). Beeri and Vardi [BV2] and Yannakakis and Papadimitriou [YP] have also independently defined this class. Beeri and Vardi call them *(many-sorted, or typed) tuple-generating and equality-generating dependencies*, and Yannakakis and Papadimitriou call them *algebraic dependencies*. The class of EID's includes all functional dependencies, multivalued and embedded multivalued dependencies, join and embedded join dependencies, and many others [Fa4]. We shall define EID's at the end of this subsection.

Fagin called a sentence σ *faithful* if whenever $<r_i: i \in I>$ is a nonempty family of nonempty relations, then σ holds for $\otimes<r_i: i \in I>$ if and only if σ holds for every r_i. He showed [Fa4] that every EID is *faithful*. It follows easily that EID's enjoy Armstrong relations. For, much as before, let Σ be a set of EID's (all involving the same relation symbol R), and let σ_1, σ_2, ... be those EID's (involving R) that are *not* logical consequences of Σ. By definition of logical consequence, there are relations r_1, r_2, ... where r_i obeys Σ but not σ_i, for each i. Each r_i is nonempty (i=1,2,...), or else it would obey σ_i (EID's are defined in such a way that an empty relation obeys every EID.) Let r be the direct product $\otimes<r_i: i=1,2,...>$. We now show that the relation r we have just formed is an Armstrong relation for Σ. We must show that r obeys Σ (and hence Σ^*), but that r violates each σ_i (i=1,2,...). Since each r_i obeys Σ, it follows by faithfulness that r obeys Σ. Further, since r_i violates σ_i, it follows again by faithfulness that r violates σ_i (i=1,2,...). This was to be shown.

An advantage of this direct product approach is that it is capable of yielding a *finite* Armstrong relation (one with a finite number of tuples), if we restrict our attention to a finite subset $\mathscr{S}$ of EID's. Thus, if $\mathscr{S}$ is a finite set of EID's (such as the set of all functional, multivalued, and join dependencies and embedded multivalued and join dependencies over a given set of attributes), then $\mathscr{S}$ enjoys *finite* Armstrong relations. (We remark that in considering *finite* Armstrong relations, it is necessary to deal with $\models_{fin}$ rather than with $\models$, where $\Sigma \models_{fin} \sigma$ if every *finite* relation that obeys Σ also obeys σ.) If $\mathscr{S}$

contains embedded dependencies, then it is unknown whether the direct product approach is constructive. For, one step of the approach involves finding dependencies σ that are not logical consequences of Σ; however, no decision procedure is known for deciding if $\Sigma \models_{fin} \sigma$ (or if $\Sigma \models \sigma$) when Σ can contain embedded dependencies, such as embedded multivalued dependencies. Note also that a constructive approach for producing finite Armstrong relations would *provide* a decision procedure, since to decide if $\Sigma \models_{fin} \sigma$, we can simply check some finite Armstrong relation for Σ to see whether σ holds for it.

Hull [Hu] observed that Armstrong's [Ar] original construction, which showed that FD's enjoy Armstrong relations, is a special case of the direct product construction.

By making use of McKinsey's Theorem (see [Sh, p. 95, exercise 7e]), Vardi [Va3] has recently obtained a new proof of the existence of Armstrong relations in the presence of EID's. This approach is distinct from, but related to, the direct product approach. For details, see Vardi [Va3].

We now wish to discuss databases, rather than just relations. We need some more conventions. We assume that we are given a fixed finite set of *relation symbols* **R** (usually called *relation names* in practice), and a positive integer, called the *arity*, associated with each relation symbol. A *database* is a mapping that associates a relation (of the proper arity) with each relation symbol. When it can cause no confusion, we may speak of the collection of relations themselves as the database. We can write first-order sentences about databases, just as we earlier wrote first-order sentences about single relations. Let $\mathcal{S}$ be a class of sentences about **R**, and assume that $\Sigma \subseteq \mathcal{S}$. An *Armstrong database* (with respect to $\mathcal{S}$) is a database that obeys precisely those members σ of $\mathcal{S}$ such that $\Sigma \models \sigma$.

The direct product construction can sometimes be used to produce an Armstrong database, even in the presence of inter-relational dependencies. The direct product of databases is the result of taking the direct product relationwise. Thus, if R is a relation name, then the R relation of the direct product is the direct product of the R relations. Let us assume that the only sentences of interest (that is, the members of $\mathcal{S}$) are inclusion dependencies [CFP] and standard FD's. Recall that a *standard FD* is an FD for which the left-hand side is nonempty. What are inclusion dependencies? 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

$$R[A_1,...,A_m] \subseteq S[B_1,...,B_m], \tag{4.1}$$

where R and S are relation names, and where the A_i's and B_i's are attributes. The inclusion dependency (4.1) holds for a database if each tuple that is a member of the relation corresponding to the left-hand side of (4.1) is also in the relation corresponding to the right-hand side of (4.1). Fagin and

Vardi [FV] show that if the sentences of interest are inclusion dependencies and standard FD's, then there is always an Armstrong database for each set Σ of sentences. They use a direct product construction identical to that used for EID's (except that they take the direct product of databases, rather than of relations.)

If, however, the sentences of interest are not inclusion dependencies and standard FD's, but rather, inclusion dependencies and (unrestricted) FD's, then the construction may fail. For, Fagin and Vardi [FV] show that in this case, there need not exist an Armstrong database. However, in this case, the direct product construction does produce what Fagin [Fa4] calls an *Armstrong-like* database, which is closely related to an Armstrong database.

We close this subsection by defining EID's. We need a few preliminary concepts. Let R be a relation symbol that represents the relation of interest. (When we deal with inter-relational constraints, then several relation symbols are needed.) We assume that we are given a set of *individual variables* (which represent entries in a relation.) Assume that R represents a d-ary relation. Then the *atomic* formulas are those that are either of the form $Rz_1...z_d$ (where the z_i's are individual variables), or else of the form $x=y$ (where x and y are individual variables.) We call atomic formulas $Rz_1...z_d$ *relational formulas,* and atomic formulas $x=y$ *equalities.*

Formulas (which can involve Boolean connectives and quantifiers) and *sentences* (formulas with no free variables) are defined as usual (see any standard textbook in logic, for example, Enderton [En] or Shoenfield [Sh].) We sometimes abbreviate $\forall x_1...\forall x_n\phi$, where each x_i is universally quantified, by $(\forall x_1...x_n)\phi$. Similarly, we sometimes abbreviate $\exists y_1...\exists y_r\phi$, where each y_i is existentially quantified, by $(\exists y_1...y_r)\phi$.

A formula is said to be *typed* if there are d disjoint classes, or types, of variables (where d is the arity, or degree, of relation symbol R, and where we say that a variable in the *i*th class is of *type i*), such that (a) if the relational formula $Rz_1...z_d$ appears in the formula, then z_i is of type i $(1 \le i \le d)$, and (b) if the equality $x=y$ appears in the formula, then x and y have the same type.

In a typed formula, no individual variable can represent an entry in two distinct columns. Thus, if Rxy appears in a typed formula (where x and y are individual variables), then Rzx cannot also appear, since if it did, then x would represent an entry in both the first and second columns.

An *embedded implicational dependency* (or *EID*) is a typed sentence of the form

$$(\forall x_1...x_m)((A_1 \wedge ... \wedge A_n) \Rightarrow (\exists y_1...y_r)(B_1 \wedge ... \wedge B_s)), \tag{4.2}$$

where each A_i is a relational formula and where each B_i is atomic (either a relational formula or an equality.) We assume also that each of the x_j's appears in at least one of the A_i's, and that $n \geq 1$, that is, that there is at least one A_i. We assume that $r \geq 0$ (if $r=0$ then there are no existential quantifiers), and that $s \geq 1$ (that is, there must be at least one B_i.)

d. The chase

Let us define a *dependency* [BV2] to be a sentence of the form (4.2) above, where each A_i is a relational formula and where each B_i is atomic (either a relational formula or an equality.) As in the case of EID's, we assume also that each of the x_j's appears in at least one of the A_i's, and that $n \geq 1$, that is, that there is at least one A_i. So far, everything that we have said holds for both EID's and for the more general class of *dependencies*. For EID's, we made the further assumptions that the sentence is typed and uni-relational (that is, not inter-relational.) For the general case of dependencies, we do not make either assumption. Dependencies, as we have defined them, are slightly more general than Fagin's XEID's (extended embedded implicational dependencies [Fa4]), since for XEID's, the *left-hand side* $A_1 \wedge ... \wedge A_n$ is typed and uni-relational.

Let Σ be a set of dependencies, over relation symbols **R**. We define Σ^* to be the set of all dependencies σ over **R** such that $\Sigma \models \sigma$. An *Armstrong database* for Σ (with respect to dependencies over **R**) is a database that obeys every member of Σ^* and no other dependency over **R**. Grant and Jacobs [GJ] present a technique which, given a set Σ of dependencies, will produce an Armstrong database for Σ if there is one. We shall describe this technique shortly. We note that their technique tends to produce an *infinite* Armstrong database, even though there may exist a finite Armstrong database.

Although Grant and Jacobs did not do so, it is convenient for us to describe their technique in terms of the *chase* ([BV1],[MMS],[SU]). Also, Grant and Jacobs restricted their attention to their "generalized dependency constraints", each of which is a *full* dependency. (A *full* dependency is one with no existential quantifiers.) We shall not make this restriction.

Let Σ be a set of dependencies (over relation symbols **R**), and, as before, let Σ^* be the set of all dependencies σ over **R** such that $\Sigma \models \sigma$. Let σ_1, σ_2, ... be all dependencies over **R** not in Σ^*. By consistently renaming individual variables, we can assume that no individual variable that appears in σ_i also appears in σ_j, if $i \neq j$.

We now define an initial database D_0, where the tuples are those that appear on the left-hand side of the σ_i's. What we mean by this should be clear by an example. If one of the σ_i's is, say,

$$(\forall xyzw)((Pxy \wedge Pyz \wedge Qxww) \Rightarrow \exists vPvx), \tag{4.3}$$

then we put the tuples (x,y) and (y,z) into the P relation of D_0, and we put the tuple (x,w,w) into the Q relation of D_0. (We are treating the variables like constants.)

Our final database is obtained by "chasing" ([BV1],[MMS],[SU]) the initial database D_0, using the dependencies in Σ. This chase process can cause new tuples to be added, and it can cause some constants to be equated (identified). We refer the reader to the literature ([BV1],[MMS],[SU]) for precise definitions of the chase process. We simply give two examples. In both examples, we assume that the P relation contains, possibly among other tuples, the tuples (x,y) and (y,z), and that the Q relation contains, possibly among other tuples, the tuple (x,w,w). (One reason that these tuples might be present is if the dependency (4.3) is not in Σ^*; then the tuples would be put into the initial database D_0, as described earlier.) Assume now that Σ contains the dependency

$$(\forall rstu)((Prs \wedge Qrtu \wedge Qrut) \Rightarrow \exists q(Quqs \wedge Pqq)). \tag{4.4}$$

By letting r,s,t,u in (4.4) be, respectively, x,y,w,w, the chase process would then add the tuple (w,q,y) to the Q relation, and the tuple (q,q) to the P relation, where q is a new symbol that does not appear as an entry in the database yet. If Σ were to contain the dependency

$$(\forall rstu)((Prs \wedge Qrtu \wedge Qrut) \Rightarrow r{=}u), \tag{4.5}$$

then by letting r,s,t,u in (4.5) be, respectively, x,y,w,w, the chase process would identify x and w (say, by replacing every occurrence of x in the database by w.) Let us call the final database that is the result of the chase process D_{chase}.

The next theorem is a slight generalization of a theorem of Grant and Jacobs (the dependencies that they considered were all full dependencies.) The proof is a minor modification of a proof by Vardi [Va3].

Theorem 4.2. Let Σ be a set of dependencies over relation symbols **R**. Assume that there is an Armstrong database for Σ (with respect to dependencies over **R**.) That is, assume that there is a database that obeys precisely those dependencies σ over **R** such that $\Sigma \models \sigma$. Then D_{chase} is also such an Armstrong database.

Proof: Let σ_1, σ_2, ... be the dependencies over **R** that are *not* in Σ^*. Say σ_i is $\forall \mathbf{x}_i(\phi_i \Rightarrow \exists \mathbf{y}_i\psi_i)$, where ϕ_i and ψ_i are conjunctions of atomic formulas, and where $\mathbf{x}_i$, $\mathbf{y}_i$ are strings of individual variables, for each i. As before, we assume that $\mathbf{x}_i$ and $\mathbf{x}_j$ have no variables in common, if $i \neq j$.

Assume that there is an Armstrong database D for Σ. This database simultaneously obeys Σ, $\neg\sigma_1$, $\neg\sigma_2$, ... Thus, there is a model (in the usual logical sense [En]) for

$$\Sigma,\ \phi_1,\ \phi_2,\ \ldots,\ \neg\exists y_1\psi_1,\ \neg\exists y_2\psi_2,\ \ldots, \tag{4.6}$$

where the free variables are being treated as constants. In other words, there is a database D and an assignment of the free variables to entries in the database (where two distinct free variables may be assigned the same entry in the database) such that (4.6) holds.

Assume now that D_{chase} is not an Armstrong database for Σ; we shall derive a contradiction. By the theory of the chase, we know that D_{chase} obeys Σ. Since D_{chase} is not an Armstrong database for Σ, it must obey some σ_i, say σ_1. By our construction of D_{chase}, it is not hard to see that this means that

$$\Sigma,\ \phi_1,\ \phi_2,\ \ldots,\ \models \exists y_1\psi_1 \tag{4.7}$$

But (4.7) contradicts the fact that there is a database D that obeys (4.6). This concludes the proof. $\square$

Let Σ be a set of dependencies. By Theorem 4.2, we know that if Σ has an Armstrong database, then the chase process will produce one. Given Σ, can we decide whether or not Σ has an Armstrong database? Vardi [Va1] has shown that we cannot, even if we restrict our attention to sets of uni-relational dependencies, all involving the same relation symbol.

Theorem 4.3 [Va1]. The problem of deciding whether a given finite set of uni-relational dependencies, all involving the same relation symbol, has an Armstrong relation is undecidable.

Theorem 4.4 [Va1]. The problem of deciding whether a given finite set of uni-relational dependencies, all involving the same relation symbol, has a finite Armstrong relation is undecidable.

e. Random relations: an approach that doesn't work

It is natural to conjecture that "almost all" relations obeying a given set of FD's are Armstrong relations for that set of FD's. If this were true (which, as we shall see, it is not), then an easy way to obtain an Armstrong relation for Σ would be to randomly select a relation that obeys Σ; with high probability, we would obtain an Armstrong relation for Σ after only a few such random choices.

What do we mean by "almost all"? Let us hold fixed a set U of attributes. Let $\mathcal{A}_n$ be the set of all relations with attributes U such that every entry of the relation is a member of $\{1,\ldots,n\}$. Thus, $\mathcal{A}_n$ contains 2^{n^u} members, where u is the number of attributes (that is, the size of U.) If $\mathcal{P}$ is a property of relations, then we say that "almost all relations have property $\mathscr{P}$" (or "a random relation has property $\mathscr{P}$") if the fraction of members of $\mathscr{A}_n$ with property $\mathscr{P}$ converges to 1 as $n \rightarrow \infty$. Fagin [Fa1] showed that if $\mathscr{P}$ is a first-order property of relations, then either almost all relations have property $\mathscr{P}$ or almost all relations violate property $\mathscr{P}$. From his characterization, it follows easily that for each nontrivial FD σ, almost all relations (over the appropriate attributes) violate σ. Since there are only a finite number of FD's over a given set of attributes, it follows that almost all relations simultaneously violate every nontrivial FD. Thus, almost all relations are Armstrong relations (with respect to FD's) for the empty set of FD's. If $\mathscr{P}$ and $\mathscr{Q}$ are properties of relations, then we say that "almost all relations with property $\mathscr{P}$ have property $\mathscr{Q}$" if the number of members of $\mathscr{A}_n$ with both properties $\mathscr{P}$ and $\mathscr{Q}$ divided by the number with property $\mathscr{P}$ converges to 1 as $n \rightarrow \infty$.

A natural conjecture is that almost all relations that obey a given set Σ of FD's is an Armstrong relation for Σ (with respect to FD's.) As we noted earlier, the conjecture is true when Σ is empty. Beeri, Dowd, Fagin, and Statman [BDFS] show that the conjecture is false in general. In fact, they show that if the attributes U are {A,B,C,D}, then almost all relations obeying the FD A→BCD also obey the FD BCD→A, and so are certainly not Armstrong relations for {A→BCD}.

5. The size of Armstrong relations

Beeri, Dowd, Fagin, and Statman [BDFS] give various results on the size and structure of Armstrong relations for FD's. In particular, they prove the following theorem, where S(n) is the binomial coefficient $\binom{n}{\lfloor n/2 \rfloor}$, and where $\lfloor x \rfloor$ is the greatest integer not exceeding x. Note that by Stirling's formula, it follows that S(n) is asymptotic to $(2/\pi)^{1/2} 2^n n^{-1/2}$.

Theorem 5.1 [BDFS]. There is a constant c such that for each set Σ of FD's involving n attributes, there is an Armstrong relation for Σ with less than $S(n)(1+(c/n^{1/2}))$ tuples. For each positive integer n, there is a set Σ of FD's involving n attributes such that each Armstrong relation for Σ contains more than $S(n)/n^2$ tuples. Thus, if p(n) is the maximum (over all sets Σ of FD's involving n attributes) of the minimum number of tuples (over all Armstrong relations for Σ), then $S(n)/n^2 < p(n) < S(n)(1+(c/n^{1/2}))$.

6. Finite Armstrong relations

If $\mathscr{S}$ is a finite set of EID's (such as the set of all functional, multivalued, and join dependencies and embedded multivalued and join dependencies over a given set of attributes), then as we noted, $\mathscr{S}$ enjoys *finite* Armstrong relations. What if $\mathscr{S}$ is infinite? Fagin's direct product argument still works, to show that there is an Armstrong relation (possibly infinite). However, he gives an example (involving EID's) to show that a finite Armstrong relation need not exist. Fagin, Maier, Ullman, and Yannakakis [FMUY] give such an example involving only *template dependencies*, or *TD's*. A TD is an EID, as in (4.2), where s=1 (that is, where the right-hand side consists of only one formula B_1), and where this formula B_1 is a relational formula, rather than an equality. Thus, there is a finite set Σ of TD's such that there is no relation that obeys precisely those TD's σ for which $\Sigma \models_{fin} \sigma$. (Recall that $\Sigma \models_{fin} \sigma$ if every *finite* relation that obeys Σ also obeys σ. Fagin, Maier, Ullman, and Yannakakis [FMUY] showed that $\models$ and $\models_{fin}$ are distinct for TD's.) This result on the nonexistence of finite Armstrong relations can also be obtained by using Vardi's result [Va2] that there is a single finite set Σ of TD's such that the set of all TD's σ for which $\Sigma \models_{fin} \sigma$ is not recursive. This result implies that there is no finite Armstrong relation for Σ, since we could test whether or not $\Sigma \models_{fin} \sigma$ by simply checking whether or not the finite Armstrong relation obeys σ. Vardi also showed [Va3] that there is a finite set of *full* TD's with no finite Armstrong relation.

Although there is a finite set Σ of TD's that has no finite Armstrong relation, there are certainly some sets Σ of TD's that do have a finite Armstrong relation. For example, if Σ is the set of all TD's, then Σ has a finite Armstrong relation, namely, any one-tuple relation; also [FMUY], if Σ is the empty set, then Σ has a finite Armstrong relation. Fagin, Maier, Ullman, and Yannakakis [FMUY] give several characterizations (Theorem 6.1 below) of those sets Σ of TD's that have a finite Armstrong relation. If Σ is a set of TD's, then define Σ^*_{fin} to be $\{\sigma: \sigma \text{ is a TD and } \Sigma \models_{fin} \sigma\}$. Thus, Σ^*_{fin} is the set of all TD's that hold in every finite relation obeying Σ. By a *finite Armstrong relation for* Σ we then mean, of course, a finite relation that obeys Σ^*_{fin} but no other TD's.

Theorem 6.1 [FMUY]. Let Σ be a set of TD's. The following are equivalent.

(a) There is a finite relation that obeys Σ^*_{fin} and no other TD's ("Σ has a finite Armstrong relation".)

(b) There is a finite set $\mathcal{T}$ of TD's, disjoint from Σ^*_{fin}, such that for each TD T not in Σ^*_{fin} there is a TD T′ in $\mathcal{T}$ where $T \models T'$.

(c) There is a finite set $\mathcal{T}$ of TD's, disjoint from Σ^*_{fin}, such that $T \models \vee\{T': T' \in \mathcal{T}\}$ for each TD T not in Σ^*_{fin}.

(d) There is a finite set $\mathcal{T}$ of TD's, disjoint from Σ^*_{fin}, such that $\vee\{T: T \notin \Sigma^*_{fin}\}$ is equivalent to $\vee\{T': T' \in \mathcal{T}\}$.

Note that $\{T': T' \in \mathcal{T}\}$ in (d) is a finite subset of $\{T: T \notin \Sigma^*_{fin}\}$ in (d). So, (d) is a kind of compactness result, that says that a certain set has a finite subcover (that is, it says that a finite number of disjuncts of $\vee\{T: T \notin \Sigma^*_{fin}\}$ "covers" all of it.)

Fagin and Vardi recently observed two consequences (Theorem 6.2 and Corollary 6.3 below) of Theorem 6.1. A *(finite) elementary class*, or *(finite) EC* [En] is the class of all (finite) relations that obey a given uni-relational first-order sentence σ. A *(finite) generalized elementary class*, or *(finite)* EC_Δ [En] is the class of all (finite) relations that obey a given set of uni-relational first-order sentences σ, all involving the same relation symbol. Let Σ be a fixed finite set of TD's, and let $\mathscr{A}$ be the class of all (finite) Armstrong relations for Σ. It is clear that $\mathscr{A}$ is a (finite) EC_Δ, since $\mathscr{A}$ is the class of all (finite) relations that obey Σ, $\neg\sigma_1$, $\neg\sigma_2$, ..., where σ_1, σ_2, ... are all of the TD's that are *not* logical consequences of Σ. Is $\mathscr{A}$ not just an EC_Δ, but even an EC? In the case where we restrict our attention to *finite* relations, Fagin and Vardi have observed that the answer is "yes".

Theorem 6.2 Let Σ be a finite set of TD's. The class of all finite Armstrong relations for Σ is a finite elementary class. That is, there is a first-order sentence σ_Σ such that whenever r is a finite relation, then r is a finite Armstrong relation for Σ if and only if r obeys σ_Σ.

Proof: If Σ has no finite Armstrong relation, then take σ_Σ to be a sentence that is logically false, such as $\exists x(x \neq x)$. So assume that Σ has a finite Armstrong relation. By the equivalence of (a) and (d) in Theorem 6.1, we know that (d) holds. Let $\mathscr{T}$ be as in (d), and let τ be the first-order sentence $\bigvee\{T'\colon T' \in \mathscr{T}\}$. Assume for definiteness that $\Sigma = \{\sigma_1, ..., \sigma_k\}$. Let σ be the sentence $\sigma_1 \wedge ... \wedge \sigma_k \wedge \neg\tau$. It is easy to see that this sentence σ can be taken to play the role of σ_Σ as called for in the statement of Theorem 6.2. □

Corollary 6.3 Let Σ be a fixed finite set of TD's. The problem of determining, given a finite relation r, whether or not r is a finite Armstrong relation for Σ is decidable.

Proof: Let σ_Σ be as in Theorem 6.2. The decision procedure consists of determining whether or not r obeys σ_Σ. □

Note that the decision procedure we have given in the proof of Corollary 6.3 is not "uniform" in Σ. Thus, it is an open problem as to the decidability of determining, given a finite set Σ of TD's and a finite relation r, whether or not r is a finite Armstrong relation for Σ.

7. Conclusions

The concept of Armstrong databases is of interest in relational database theory and in mathematical logic. In this paper, history and results about Armstrong databases are discussed. It is (hopefully!) premature to be writing a history of Armstrong databases, since they have been studied explicitly for only a few years. The author hopes that this survey paper will inspire others to study this fascinating topic.

8. Acknowledgments

The author is grateful to Moshe Vardi for helpful comments.

BIBLIOGRAPHY

[Ar] W. W. Armstrong, Dependency structures of database relationships. Proc. IFIP 74, North Holland (1974), 580-583.

[AD] W. W. Armstrong and C. Delobel, Decompositions and functional dependencies in relations, ACM Trans. on Database Systems 5,4 (Dec. 1980), 404-430.

[Be] C. Beeri, personal communication.

[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.

[BDFS] C. Beeri, M. Dowd, R. Fagin, and R. Statman, On the structure of Armstrong relations for functional dependencies. IBM Research Report RJ 2901 (Sept. 1980).

[BFH] C. Beeri, R. Fagin, and J. H. Howard, A complete axiomatization for functional and multivalued dependencies in database relations. Proc. 1977 ACM SIGMOD (ed. D.C.P. Smith), Toronto, 47-61.

[BV1] C. Beeri and M. Y. Vardi, A proof procedure for data dependencies. Hebrew University of Jerusalem Technical Report (August 1980).

[BV2] C. Beeri and M. Y. Vardi, The implication problem for data dependencies. Proc. XP1 Workshop (Stony Brook, NY, June 1980). Also in Proc. 8th ICALP, Acre Israel (July 1981). Appeared in: Lecture Notes in Computer Science 115, Springer-Verlag (1981), 73-85.

[CFP] M. A. Casanova, R. Fagin, and C. Papadimitriou, Inclusion dependencies and their interaction with functional dependencies. Proc. First ACM SIGACT-SIGMOD Principles of Database Systems (1982), 171-176.

[Co] E. F. Codd, Further normalization of the data base relational model. In Courant Computer Science Symposia 6: Data Base Systems, May 24-25, 1971, R. Rustin, editor, Prentice Hall, 33-64.

[De] J. Demetrovics, On the number of candidate keys. Inf. Proc. Letters 7,6 (Oct. 1978), 266-269.

[En] H. B. Enderton, A Mathematical Introduction to Logic. Academic Press (1972).

[Fa1] R. Fagin, Probabilities on finite models. J. Symbolic Logic 41,1 (Mar. 1976), 50-58.

[Fa2] R. Fagin, Multivalued dependencies and a new normal form for relational databases. ACM Trans. on Database Systems 2,3 (Sept. 1977), 262-278.

[Fa3] R. Fagin, Functional dependencies in a relational database and propositional logic. IBM J. Res. & Devel. 21,6 (November 1977), 534-544.

[Fa4] R. Fagin, Horn clauses and database dependencies. Proc. 1980 ACM SIGACT Symposium on Theory of Computing, pp. 123-134. Also, to appear, J. ACM.

[FMUY] R. Fagin, D. Maier, J. D. Ullman, and M. Yannakakis, Tools for template dependencies. To appear, SIAM J. Computing.

[FV] R. Fagin and M. Y. Vardi, Armstrong databases for functional and inclusion dependencies. To appear.

[Go] E. M. Gold, Axiomatization of attribute dependencies. To appear, J. ACM.

[GJ] J. Grant and B. E. Jacobs, On the family of generalized dependency constraints. To appear, J. ACM.

[Gr] G. Grätzer, Universal Algebra. Springer, 1973.

[Hu] R. Hull, Implicational dependency and finite specification. Univ. of Southern California Technical Report (1981).

[MMS] D. Maier, A. Mendelzon, and Y. Sagiv, Testing implications of data dependencies. ACM Trans. on Database Systems 4,4 (Dec. 1979), 455-469.

[Ri] J. Rissanen, Theory of relations for databases - a tutorial survey. Proc. 7th Symposium on Math. Found. of Comp. Science. Lecture Notes in Comp. Science, 64. Springer-Verlag, 537-551.

[SU] F. Sadri and J. D. Ullman, A complete axiomatization for a large class of dependencies in relational databases. Proc. 1980 ACM SIGACT Symposium on Theory of Computing, 117-122.

[Sh] J. R. Shoenfield, Mathematical Logic. Addison-Wesley (1967).

[SM] A. M. Silva and M. A. Melkanoff, A method for helping discover the dependencies of a relation. In Advances in Data Base Theory, Vol. 1, ed. H. Gallaire, J. Minker, and J-M. Nicolas, Plenum Publishing, NY, 1981.

[Va1] M.Y. Vardi, Global decision problems for relational databases. Proc. 1981 IEEE Symp. on Foundations of Computer Science, 198-202.

[Va2] M. Y. Vardi, The implication and finite implication problems for typed template dependencies. Proc. First ACM SIGACT-SIGMOD Principles of Database Systems (1982), 230-238.

[Va3] M. Y. Vardi, On the existence of Armstrong relations for sets of data dependencies. To appear.

[YP] M. Yannakakis and C. Papadimitriou, Algebraic dependencies. Proc. 1980 IEEE Symp. on Found. of Computer Science, 328-332.

[Zl] M. M. Zloof, Proc. 1975 AFIPS National Computer Conference, vol. 44, AFIPS Press, Arlington, Va., 431-438.

EMP	DEPT	MGR
Hilbert	Math	Gauss
Pythagoras	Math	Gauss
Turing	Computer Science	von Neumann
Einstein	Physics	Gauss

Figure 1.1

A	B	C
a_1	b_1	c_1
a_1	b_1	c'_1

A	B	C
a_2	b_2	c_2
a_2	b'_2	c_2

A	B	C
$\langle a_1,a_2\rangle$	$\langle b_1,b_2\rangle$	$\langle c_1,c_2\rangle$
$\langle a_1,a_2\rangle$	$\langle b_1,b'_2\rangle$	$\langle c_1,c_2\rangle$
$\langle a_1,a_2\rangle$	$\langle b_1,b_2\rangle$	$\langle c'_1,c_2\rangle$
$\langle a_1,a_2\rangle$	$\langle b_1,b'_2\rangle$	$\langle c'_1,c_2\rangle$

Figure 4.1