Information Processing Letters 16 (1983) 13–19
North-Holland Publishing Company

24 January 1983

ARMSTRONG DATABASES FOR FUNCTIONAL AND INCLUSION DEPENDENCIES

Ronald FAGIN

IBM Research Laboratory, San Jose, CA 95193, U.S.A.

Moshe Y. VARDI *

Stanford University, Stanford, CA 94305, U.S.A.

Communicated by Ken C. Sevcik
Received 17 June 1982
Revised 17 September 1982

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. It is shown that if the sentences of interest are inclusion dependencies and standard functional dependencies (functional dependencies for which the left-hand side is nonempty), then there is always an Armstrong database for each set of sentences. (An example of an inclusion dependency is the sentence that says that every MANAGER is an EMPLOYEE.) If, however, the sentences of interest are inclusion dependencies and unrestricted functional dependencies, then there need not exist an Armstrong database. This result holds even if we allow only 'full' inclusion dependencies. Thus, a fairly sharp line is drawn, in a case of interest, as to when an Armstrong database must exist. These results hold whether we restrict our attention to finite databases (databases with a finite number of tuples), or whether we allow unrestricted databases.

Keywords: Armstrong relation, Armstrong database, relational database, functional dependency, inclusion dependency, direct product, faithfulness, logical consequence, mathematical logic

1. Introduction

The concept of Armstrong databases (and, as a special case, Armstrong relations) is relatively new, although a number of papers have now dealt with them explicitly [4,13,14,15,16,17,18,22,26,28]. A survey paper [14] about history and results on Armstrong databases has recently appeared.

Let us begin by discussing Armstrong relations, which are a special case (where the database consists of a single relation). For simplicity, we further restrict our attention initially by considering only functional dependencies or FD's [11]. The definition of FD's (along with some other definitions) appears in Section 2.

Let Σ be a set of sentences, such as FD's, and let σ be a single sentence. When we say that Σ *logically implies* σ or the 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

* Research supported by a Weizmann Post-doctoral Fellowship, Fulbright Award and NSF Grant MCS-80-12907.

Armstrong relation is a relation that obeys Σ^* and *no other* FD's.

As an example [13], let Σ be the set

$$\{EMP \to DEPT, DEPT \to MGR\},$$

containing two FD's. Then Σ^* contains the FD's in Σ, along with, for example, the FD EMP $\to$ 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 Table 1 is an Armstrong relation for Σ, that is, that it obeys every FD in Σ^* and no others. For example, the FD MGR $\to$ 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 [19], 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 [13].) 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).

There is an interesting 'practical' application for Armstrong relations. Silva and Melkanoff [25] have developed a database design aid, in which the database designer inputs a set of FD's and MVD's (multivalued dependencies) [12]. 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 the input set. (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

$$\{EMP \to DEPT, DEPT \to MGR\}$$

of FD's. The database design aid would then present the designer with an Armstrong relation, such as relation r in Table 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,* my set of input independencies 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 $\to$ DEPT".

In this example, the designer did not have to explicity think about the dependency MGR $\to$ 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 is said, he simply *noticed* that the FD MGR $\to$ DEPT failed. Thus, Silva and Melkanoff's approach is a partial solution, in the spirit of query-by-example [31], to the problem of helping a designer think of what dependencies should be included.

We now mention another viewpoint, due to Ginsburg and Zaiddan [17], about Armstrong relations. By an *FD class* [13] we mean the collection of all relations (with a fixed set of attributes, i.e. column names) that obey a given set of FD's. Let r be a fixed relation. In the spirit of Ginsburg and Zaiddan [17], we define *the FD class generated by* r to be the smallest FD class that contains r. It is easy to see that this class consists of those relations (with attributes the same as those of r) that obey Σ, where Σ is the set of all FD's obeyed by r. A natural question is whether every FD class has a

Table 1

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

generator. The answer [17] is yes: if the FD class $\mathfrak{R}$ consists of all relations with attributes U that obey Σ, then let r be an Armstrong relation (with attributes U) for Σ; it is easy to see that r is a generator for the class $\mathfrak{R}$. Thus, a natural interpretation for Armstrong relations is as class generators.

We now begin to generalize (from FD's to more general types of sentences, and later, from relations to databases). Let $\mathbb{S}$ be a set of sentences (such as the set of all FD's over a given set of attributes). Assume that for each set Σ of members of $\mathbb{S}$, there is a relation that obeys precisely the members σ of $\mathbb{S}$ such that $\Sigma \models \sigma$. We then say that $\mathbb{S}$ *enjoys Armstrong relations.*

It is an implicit result in Armstrong's paper [2] that FD's enjoy Armstrong relations, and it is an implicit result in Beeri, Fagin and Howard's paper [5] that FD's and MVD's enjoy Armstrong relations. Beeri [3] extended the result to FD's and join dependencies [23]. In 1980, Fagin [13] showed that a large class of dependencies called *embedded implicational dependencies,* or EID's, enjoy Armstrong relations. Embedded implicational dependencies are also known as *many-sorted (typed) tuple-* and *equality-generating dependencies* [7] and as *algebraic dependencies* [30]. They are all *uni-relational sentences,* that is, sentences about single relations rather than about multi-relation databases. They include functional and multivalued dependencies as special cases. Fagin [13] gives a number of examples to show that by allowing very mild extensions of EID's (that is, by taking $\mathbb{S}$ to include sentences only slightly more general than EID's), then there are not necessarily Armstrong relations. Thus, we now have a fairly good understanding as to the 'boundary line' between classes $\mathbb{S}$ that do and do not enjoy Armstrong relations.

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 $\mathbb{S}$ be a class of sentences about **R** (for example, $\mathbb{S}$ could be the collection of all functional dependencies and inclusion dependencies involving **R**). Assume that $\Sigma \subseteq \mathbb{S}$; an *Armstrong database* for Σ (with respect to $\mathbb{S}$) is a database that obeys precisely those members σ of $\mathbb{S}$ such that $\Sigma \models \sigma$.

For databases, we may be interested in *multi-relational* sentences. Probably the most important such sentence for relational databases is the *inclusion dependency* or IND [9]. 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

$$R[A_1,\ldots,A_m] \subseteq S[B_1,\ldots,B_m] \qquad (1.1)$$

where R and S are relation names, 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, IND's are valuable for database design, since they permit us to selectively define what data must be duplicated in what relations.

IND's, together with FD's, form the basis of the structural model of Wiederhold and El-Masri [29]. They also appear when an entity-relationship schema is mapped to the relational model [10,21]. Yet, in another perspective, IND's can be viewed as a relaxation of the controversial universal relation assumption [8,20,27], 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. [6]).

The main result of this paper is that if the sentences of interest (the members of the class $\mathbb{S}$, as used above) are inclusion dependencies and standard functional dependencies (functional dependencies for which the left-hand side is nonempty), then there is always an Armstrong database for each set of sentences. If, however, the sentences of interest are inclusion dependencies and unrestricted functional dependencies, then there need not exist an Armstrong database.

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 $\langle A_1, \ldots, A_m \rangle$ 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 $\langle A_1, \ldots, A_m \rangle$, simply as $A_1, \ldots, A_m$. For example, we shall write simply $R[A_1, \ldots, A_m]$ for $R[\langle A_1, \ldots, A_m \rangle]$. A *tuple* t over $U = \langle A_1, \ldots, A_m \rangle$ is a sequence $\langle a_1, \ldots, a_m \rangle$ of the same length m as U. A *relation* (over R[U], or simply over R) is a set of tuples over U. If $\langle a_1, \ldots, a_m \rangle$ is a tuple in relation r, then we say that a_i is an *entry* (*in column* A_i), $1 \le i \le m$. Note that our definition, which is convenient for use in this paper, is distinct from other definitions [1,2] in which a tuple is a mapping, not a sequence. If $X = \langle A_{i_1}, \ldots, A_{i_k} \rangle$ where $i_1, \ldots, i_k$ are distinct members of $\{1, \ldots, n\}$, and if t is as above, then t[X] is $\langle a_{i_1}, \ldots, a_{i_k} \rangle$. If r is a set of tuples over U, then $r[X] = \{t[X]: t \in r\}$.

A *database scheme* $D = \{R_1[U_1], \ldots, R_n[U_n]\}$ 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 $\{r_1, \ldots, r_n\}$ as the database. We may refer to r_i as 'the R_i relation' ($1 \le i \le n$).

If $R[A_1, \ldots, A_m]$ is a relation scheme, and if X is a sequence of distinct members of $A_1, \ldots, A_m$, as is Y, then we call $R: X \to Y$ a *functional dependency* or FD. (If R is understood, then we may simply write $X \to Y$ for $R: X \to Y$.) If X is the empty sequence, then we call $R: X \to Y$ a *nonstandard* FD; otherwise, we call $R: X \to Y$ a *standard* 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 FD's and inclusion dependencies, defined below. 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 $t_1[Y] = t_2[Y]$. Note that if a relation r obeys the nonstandard FD $R: \emptyset \to Y$, then there is at most one entry in column A, for each A in Y. That is, if r obeys the nonstandard FD $\emptyset \to Y$, if t_1 and t_2 are tuples in r, and if $A \in Y$, then $t_1[A] = t_2[A]$.

If $R_i[A_1, \ldots, A_m]$ and $R_j[B_1, \ldots, B_p]$ are relation schemes (not necessarily distinct), if X is a sequence of k distinct members of $A_1, \ldots, A_m$, and if Y is a sequence of k distinct members of $B_1, \ldots, B_p$, then we call $R_i[X] \subseteq R_j[Y]$ an *inclusion dependency* or IND. (Inclusion dependencies should not be confused with the *subset dependencies* of Sagiv and Walecka [24], which are quite different.) If $\{r_1, \ldots, r_n\}$ is a database d over $D = \{R_1[U_1], \ldots, R_n[U_n]\}$, then d obeys the IND

$$R_i[X] \subseteq R_j[Y] \quad \text{if} \quad r_i[X] \subseteq r_j[Y].$$

3. Existence of an Armstrong database for IND's and standard FD's

In this section we show that if the sentences of interest are IND's and standard FD's, then there is always an Armstrong database for each set of sentences.

Let $\mathbf{r} = \{r_1, \ldots, r_n\}$ be a database. We say that $\mathbf{r}$ is *relationwise nonempty* is each r_i is nonempty. Let Σ be a set of sentences, and let σ be a single sentence. We say that $\Sigma \models_{\text{nonempty}} \sigma$ if every relationwise nonempty database that obeys Σ also obeys σ. Clearly, if $\Sigma \models \sigma$, then $\Sigma \models_{\text{nonempty}} \sigma$; however, simple examples show that the converse is false [13].

Lemma 3.1. *Let Σ be a set of IND's and standard FD's. There is a database that obeys precisely those IND's and standard FD's σ such that $\Sigma \models_{\text{nonempty}} \sigma$.*

Proof. This follows immediately from results of Fagin [13]. He calls a database, as described in the lemma, an *Armstrong-like database.* He shows, by a direct product argument, that if attention is restricted to "extended embedded implicational dependencies", of which FD's and IND's are special cases, then Armstrong-like databases necessarily exist. □

Lemma 3.2. *Let $\mathbf{r} = \{r_1, \ldots, r_n\}$ be a database in which 0 does not appear as an entry in any of the* *relations* r_i. *Let* r'_i $(1 \le i \le n)$ *be the result of adding a tuple of all* 0*'s to* r_i, *and let* $\mathbf{r}' = \{r'_1, \ldots, r'_n\}$. *Let* α *be an IND or a standard FD. Then* $\mathbf{r}$ *obeys* α *if and only if* $\mathbf{r}'$ *obeys* α.

Proof. *Case* 1: α *is an IND.* Let α be the IND $R_i[X] \subseteq R_j[Y]$. Let r_i (respectively, r_j) be the R_i (respectively, R_j) relation in $\mathbf{r}$, and similarly for r'_i and r'_j. Assume first that $\mathbf{r}$ obeys α; we must show that $\mathbf{r}'$ obeys α. Let t be an arbitrary tuple in r'_i; to show that $\mathbf{r}'$ obeys the IND $R_i[X] \subseteq R_j[Y]$, we must show that $t[X] \in r'_j[Y]$. If t is a tuple of all 0's, then $t[X] \in r'_j[Y]$, since r'_j contains a tuple of all 0's. Otherwise (if t is not a tuple of all 0's), then t is a tuple in r_i. So, since $\mathbf{r}$ obeys the IND $R_i[X] \subseteq R_j[Y]$, it follows that $t[X] \in r_j[Y]$, and hence $t[X] \in r'_j[Y]$, since $r_j[Y] \subseteq r'_j[Y]$. So in each case, $t[X] \in r'_j[Y]$. Thus, $\mathbf{r}'$ obeys α. We have shown that if $\mathbf{r}$ obeys α, then $\mathbf{r}'$ obeys α.

Conversely, assume that $\mathbf{r}'$ obeys α; we must show that $\mathbf{r}$ obeys α. As before, let α be the IND $R_i[X] \subseteq R_j[Y]$. Let t be an arbitrary tuple in r_i; to show that $\mathbf{r}$ obeys the IND $R_i[X] \subseteq R_j[Y]$, we must show that $t[X] \in r_j[Y]$. Since $t \in r_i$, we know that $t[X] \in r_i[X]$. So, since $r_i \subseteq r'_i$, we know that $t[X] \in r'_i[X]$. Therefore, since $\mathbf{r}'$ obeys the IND $R_i[X] \subseteq R_j[Y]$, it follows that $t[X] \in r'_j[Y]$. Now t[X] contains no 0's, because $t \in r$. Because t[X] contains no 0's, and because $t[X] \in r'_j[Y]$, it follows that $t[X] \in r_j[Y]$. This was to be shown.

Case 2: α *is a standard FD.* Let α be the standard FD $R_i : X \to Y$. Thus, X is nonempty. If $\mathbf{r}'$ obeys α (that is, if r'_i obeys α), then also $\mathbf{r}$ obeys α (that is, r_i obeys α), since $r_i \subseteq r'_i$. (We have used the simple fact that if a relation obeys an FD, then so does every subrelation, where a *subrelation* is a subset of the tuples.) Conversely, assume that r_i obeys α; we must show that r'_i obeys α. Let t_1 and t_2 be two tuples of r'_i such that $t_1[X] = t_2[X]$; we must show that $t_1[Y] = t_2[Y]$. If neither t_1 nor t_2 is a tuple of all 0's, then each is in r_i. So, since r_i obeys the FD $R_i : X \to Y$, it follows that $t_1[Y] = t_2[Y]$, which was to be shown. Assume now that one of t_1 or t_2, say t_1, is a tuple of all 0's. Since $t_1[X] = t_2[X]$, and since X is nonempty, it follows that t_2 contains at least one 0, and so t_2 is necessarily a tuple of all 0's. Hence, $t_1 = t_2$, and so once again, $t_1[Y] = t_2[Y]$, which was to be shown. □

Lemma 3.3. *Let* Σ *be a set of IND's and standard FD's, and let* σ *be an IND or a standard FD. Then* $\Sigma \models_{\text{nonempty}} \sigma$ *if and only if* $\Sigma \models \sigma$.

Proof. As noted, it is obvious that if $\Sigma \models \sigma$, then $\Sigma \models_{\text{nonempty}} \sigma$. So, assume that $\Sigma \models_{\text{nonempty}} \sigma$; we shall show that $\Sigma \models \sigma$. Let $\mathbf{r} = \{r_1, \ldots, r_n\}$ be a database that obeys Σ. We must show that $\mathbf{r}$ obeys σ. Let 0 be an entry that does not appear in any of the relations r_i, let r'_i $(1 \le i \le n)$ be the result of adding a tuple of all 0's to r_i, and let $\mathbf{r}' = \{r'_1, \ldots, r'_n\}$. Since $\mathbf{r}$ obeys Σ, it follows from Lemma 3.2 that $\mathbf{r}'$ obeys Σ. Since $\mathbf{r}'$ is relationwise nonempty (by construction), and since $\Sigma \models_{\text{nonempty}} \sigma$, it follows that $\mathbf{r}'$ obeys σ. So, by Lemma 3.2 again, it follows that $\mathbf{r}$ obeys σ. This was to be shown. □

Theorem 3.4 (Existence of Armstrong databases). *Let* Σ *be a set of IND's and standard FD's. There is a database that obeys precisely those IND's and standard FD's* σ *such that* $\Sigma \models \sigma$.

Proof. By Lemma 3.1 there is a database that obeys precisely those IND's and standard FD's σ such that $\Sigma \models_{\text{nonempty}} \sigma$. By Lemma 3.3 we know that $\Sigma \models_{\text{nonempty}} \sigma$ if and only if $\Sigma \models \sigma$. The result follows immediately. □

4. Nonexistence of an Armstrong database for IND's and FD's

In this section we show that if the sentences of interest are IND's and (unrestricted) FD's, then there is a set Σ of sentences with no Armstrong database.

Theorem 4.1 (Nonexistence of Armstrong databases). *There is a set* Σ *of IND's and FD's such that there is no database that obeys precisely those IND's and FD's* σ *where* $\Sigma \models \sigma$.

Proof. Let the database scheme consist of four unary relation schemes $R_1[A_1], \ldots, R_4[A_4]$, where each A_i is a single attribute. Let Σ be the set

$$\{\emptyset \to A_1, A_2 \subseteq A_1, A_2 \subseteq A_3\}.$$

(We are abbreviating the FD $R_1 : \emptyset \to A_1$ by simply $\emptyset \to A_1$; we are abbreviating the IND $R_2[A_2] \subseteq R_1[A_1]$ by simply $A_2 \subseteq A_1$, etc.) Let σ_1 be the IND $A_1 \subseteq A_3$, and let σ_2 be the IND $A_2 \subseteq A_4$. We shall prove below the following facts.

Fact 1. *If* $\mathbf{r}$ *is a database that obeys* Σ, *then* $\mathbf{r}$ *obeys either* σ_1 *or* σ_2.

Fact 2. $\Sigma \not\models \sigma_1$.

Fact 3. $\Sigma \not\models \sigma_2$.

Proof of Theorem 4.1 (*continued*). The theorem follows from Facts 1, 2 and 3. For, let $\mathbf{r} = \langle r_1, \ldots, r_4 \rangle$ be an Armstrong database for Σ. By Fact 1 we know that $\mathbf{r}$ obeys either σ_1 or σ_2, say σ_1. Since $\mathbf{r}$ obeys σ_1, and since $\mathbf{r}$ is an Armstrong database for Σ, it follows that $\Sigma \models \sigma_1$. This contradicts Fact 2. So, the proof is complete if we prove Facts 1, 2 and 3.

Proof of Fact 1. Let $\mathbf{r} = \langle r_1, \ldots, r_4 \rangle$ be a database that obeys Σ. If r_2 is empty, then $\mathbf{r}$ obeys σ_2. So assume that r_2 is nonempty. We shall then show that $\mathbf{r}$ obeys σ_1, which completes the proof of Fact 1. Since $\mathbf{r}$ obeys Σ, we know that r_1 contains at most one entry, because of the FD $\emptyset \to A_1$. Furthermore, since $\mathbf{r}$ obeys Σ, we know that $r_2 \subseteq r_1$ and $r_2 \subseteq r_3$. Since $r_2 \subseteq r_1$ and since r_2 is nonempty, it follows that r_1 is nonempty. Since also r_1 contains at most one entry, it follows that r_1 contains precisely one entry. So, since r_2 is nonempty and $r_2 \subseteq r_1$, it follows that $r_2 = r_1$. Thus, since $r_2 \subseteq r_3$, it follows that $r_1 \subseteq r_3$. Hence, $\mathbf{r}$ obeys σ_1, which was to be shown.

Proof of Fact 2. Let $r_1 = \{0\}$, that is, let r_1 contain precisely the single entry 0. Let r_2, r_3 and r_4 be empty. It is easy to verify that $\mathbf{r} = \langle r_1, \ldots, r_4 \rangle$ obeys Σ but not σ_1.

Proof of Fact 3. Let $r_1 = r_2 = r_3 = \{0\}$, and let r_4 be empty. It is easy to verify that $\mathbf{r} = \langle r_1, \ldots, r_4 \rangle$ obeys Σ but not σ_2. □

It is interesting to note that each of the dependencies in the proof of Theorem 4.1 are 'full' [13]. (An inclusion dependency $R[X] \subseteq S[Y]$ is *full* if X contains all of the attributes of R, and if Y contains all of the attributes of S.) Thus, Theorem 4.1 holds even if the only inclusion dependencies that are allowed are full. Note also that our proofs show that all of the results in this paper hold whether we restrict our attention to finite databases (databases with a finite number of tuples), or whether we allow unrestricted databases.

References

[1] A.V. Aho, C. Beeri and J.D. Ullman, The theory of joins in relational databases, ACM Trans. on Database Systems 4(3) (1979) 297–314.

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

[3] C. Beeri, Personal communication, 1979.

[4] C. Beeri, M. Dowd, R. Fagin and R. Statman, On the structure of Armstrong relations for functional dependencies, IBM Res. Rept. RJ 2901, 1980; J. ACM, to appear.

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

[6] C. Beeri and H.F. Korth, Compatible attributes in a universal relation, Proc. 1st ACM SIGACT-SIGMOD Conf. on Principles of Database Systems (1982) pp. 55–62.

[7] C. Beeri and M.Y. Yardi, The implication problem for data dependencies, Proc. XP1 Workshop (Stony Brook, New York, 1980); also in Proc. 8th ICALP, Lecture Notes in Comput. Sci. 115 (Springer, Berlin, 1981) pp. 73–85.

[8] P.A. Bernstein and N. Goodman, What does Boyce–Codd normal form do?, Proc. 6th Internat. Conf. on Very Large Databases (1980) pp. 245–269.

[9] M.A. Casanova, R. Fagin and C. Papadimitriou, Inclusion dependencies and their interaction with functional dependencies, Proc. 1st ACM SIGACT-SIGMOD Conf. on Principles of Database Systems (1982) pp. 171–176.

[10] P. Chen, The entity-relationship model – toward a unified view of data, ACM Trans. on Database Systems 1(1) (1976) 9–36.

[11] E.F. Codd, Further normalization of the data base relational model, in: R. Rustin, ed., Courant Computer Science Symposia 6: Data Base Systems (Prentice-Hall, Englewood Cliffs, NJ, 1971) pp. 33–64.

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

[13] R. Fagin, Horn clauses and database dependencies, Proc. 1980 ACM SIGACT Symp. on Theory of Computing (1980) pp. 123–134; J. ACM, to appear.

[14] R. Fagin, Armstrong databases, Proc. 7th IBM Symp. on Mathematical Foundations of Computer Science, Kanagawa, Japan, 1982.

[15] R. Fagin, D. Maier, J.D. Ullman and M. Yannakakis, Tools for template dependencies, SIAM J. Comput., to appear.

[16] S. Ginsburg and R. Hull, Order dependency in the relational model, Univ. of Southern California Tech. Rept., 1981.

[17] S. Ginsburg and S.M. Zaiddan, Properties of functional dependency families, J. ACM 29(4) (1982).

[18] J. Grant and B.E. Jacobs, On the family of generalized dependency constraints, J. ACM 29(4) (1982).

[19] G. Grätzer, Universal Algebra (Springer, Berlin, 2nd ed., 1979).

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

[21] A. Klug, Entity-relationship views over uninterpreted enterprise schemas, Proc. Internat. Conf. Entity-Relationship Approach to Systems Analysis and Design (1979) pp. 52–72.

[22] J.A. Makowsky, Characterizing data base dependencies, Proc. 8th ICALP, Lecture Notes in Comput. Sci. 115 (Springer, Berlin, 1981) pp. 86–97.

[23] J. Rissanen, Theory of relations for databases – a tutorial survey, Proc. 7th Symp. on Math. Found. of Comput. Sci., Lecture Notes in Comput. Sci. 64 (Springer, Berlin) pp. 537–551.

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

[25] A.M. Silva and M.A. Melkanoff, A method for helping discover the dependencies of a relation, in: H. Gallaire, J. Minker and J.-M. Nicolas, eds., Advances in Data Base Theory Vol. 1 (Plenum, New York, 1981).

[26] R. Statman, Elementary theories of database dependencies, to appear.

[27] J.D. Ullman, The U.R. strikes back, Proc. 1st ACM SIGACT-SIGMOD Conf. on Principles of Database Systems (1982) pp. 10–22.

[28] M.Y. Vardi, Global decision problems for relational databases, Proc. 1981 IEEE Symp. on Foundations of Comput. Sci. (1981) pp. 198–202.

[29] G. Wiederhod and R. El-Masri, A structural model for database Systems, Techn. Rept. STAN-CS-79-722, Stanford University, 1979.

[30] M. Yannakakis and C. Papadimitriou, Algebraic dependencies, Proc. 1980 IEEE Symp. on Found. of Comput. Sci. (1980) pp. 328–332; J. Comput. System Sci., to appear.

[31] M.M. Zloof, Query-by-example, Proc. 1975 AFIPS Nat. Comput. Conf. 44 (AFIPS Press, Arlington, VA, 1975) pp. 431–438.