Acta Informatica 20, 143–158 (1983)

On the Relationship of Minimum and Optimum Covers for a Set of Functional Dependencies*

Heikki Mannila and Kari-Jouko Räihä

Department of Computer Science, University of Helsinki, Tukholmankatu 2, SF-00250 Helsinki 25, Finland

Summary. Most algorithms in relational database theory use a set of functional dependencies as their input. The efficiency of the algorithms depends on the size of the set. The notions of a *minimum* set (with as few dependencies as possible) and an *optimum* set (which is as short as possible) were introduced by Maier. He showed that while a minimum cover for a given set of dependencies can be found in polynomial time, obtaining an optimum cover is an NP-complete problem. Here the relationship of these covers is explored further. It is shown that the length of a minimum set (i) cannot be bounded by a linear function on the length of an optimum cover, and (ii) is bounded by the square of the length of an optimum cover. It is also shown that the NP-completeness of the optimization problem is somewhat surprisingly caused solely by the difficulty of optimizing a single class of dependencies having equivalent left sides, not by the globality of the optimality condition. This result has some practical significance, since the equivalence classes appearing in practice are short. The problem of optimizing an equivalence class is studied and left and right sides of a dependency are shown to behave differently. A new representation for equivalence classes based on this observation is suggested. The optimization of single dependencies is shown to be NP-complete, and a method to produce good approximations is given.

1. Introduction

In the relational database model [6, 7], data dependencies are used to express integrity constraints for the relations that can exist as an instantiation of a relation scheme. *Functional dependencies* are the most common form of data dependencies. Informally, if a functional dependency $X \to Y$ holds in a relation scheme R, it indicates the following constraint on an acceptable relation r: whenever there are two tuples in r that agree on the X-attributes, they also agree on the Y-attributes.

* This work was supported by the Academy of Finland

A typical example of the use of functional dependencies is the schema design problem. Suppose we have the attributes AUTHOR, ADDRESS, BOOK and PUBLISHER with the functional dependencies AUTHOR → ADDRESS and AUTHOR BOOK → PUBLISHER. If the data is stored using a single schema (AUTHOR, ADDRESS, BOOK, PUBLISHER), the author's address is repeated for each book that he/she has published. This is clearly a waste of space, since the dependency AUTHOR → ADDRESS tells that the address does not depend on the book. A better idea is to use two schemas: (AUTHOR, ADDRESS) and (AUTHOR, BOOK, PUBLISHER), and to store the data in the corresponding two relations.

A popular method of designing the database schema is the so-called synthesizing algorithm [4, 5, 10]. If

$$F=\{X_i \rightarrow Y_i \mid 1 \leqq i \leqq n\}$$

is the set of functional dependencies, the database schema produced by this method is

$$\{X_i Y_i \mid 1 \leqq i \leqq n\}.$$

These relation schemes preserve all dependencies and are in third normal form. If a key of the attribute set is added to the database schema, the relation schemes have a lossless join [1], another design goal.

The set of functional dependencies representing the integrity constraints is not unique. New dependencies can be derived from a given set F using *inference rules* and *axioms*. The set of all dependencies that can be derived from F is called the *closure* of F and is denoted by F^+. There are in general many sets of functional dependencies G such that $G^+ = F^+$. Any such set could as well be used to represent the same set of integrity constraints. We say that G is a *cover* of F and vice versa.

Let $|F|$ denote the number of dependencies in F, and $\|F\|$ the number of attributes appearing in F (with repetitions counted). A set of dependencies F is *minimum*, if $|F| \leqq |G|$ for any cover G of F. Furthermore, F is *optimum*[1], if $\|F\| \leqq \|G\|$. The synthesizing algorithm is a prime example of why we are interested in finding small covers for a given set of dependencies: using a minimum cover minimizes the number of relations that need to be stored, and an optimum cover minimizes the amount of storage required for storing the relations in third normal form (at least if the domains of attributes are not considered).

In the synthesizing algorithm the cover affects the quality of the output. In many situations only the efficiency of the algorithm depends on the size of the cover. For example, computing the closure X^+ of a set X of attributes can be done in time $O(\|F\|)$ [2]; finding a lossless join decomposition into Boyce-Codd normal form takes time $O(|R|^3 \cdot \|F\|)$, where $|R|$ is the number of attributes in R [14]; maintaining the integrity constraints depends linearly on $\|F\|$; and so on.

Thus there are good reasons for trying to minimize a set of functional dependencies before using it in database design algorithms, as long as the

[1] The term optimal is used in [13]

minimization can be done efficiently. The first result on the complexity issue was not encouraging: Bernstein [3] showed that for a given set F, it is an NP-complete problem to find a minimum set G such that $G \subseteq F$ and $G^+ = F^+$.

A fundamental study of the properties of minimum covers was carried out by Maier [13]. He showed that Bernstein's complexity result is caused by the requirement $G \subseteq F$. If this requirement is dropped, a minimum dependency set G such that $G^+ = F^+$ can be found in time $O(|F| \cdot \|F\|)$ (an alternative proof can be found in [9] and an alternative algorithm in [8]). Moreover, Maier showed that although an optimum cover is always minimum, the problem of finding an optimum cover is NP-complete.

This is an unfortunate result: the suggested synthesizing algorithm produces a better result for an optimum cover. Likewise, the efficiency of most design algorithms depends on $\|F\|$, not on $|F|$. Thus optimum covers are more desirable than minimum. In this paper we explore the relationship of these concepts further.

We first study in Sect. 3 how close to optimum covers we can get by using minimum (actually LR-minimum) covers. Unfortunately, it turns out that the ratio $\|F\|/\|G\|$, where F is minimum and G its optimum cover, cannot be bounded by any constant. Thus minimum covers can be arbitrarily poor approximations of optimum covers in the worst case. We also establish an upper bound for $\|F\|$: $\|F\| \leqq \|G\|^2$.

It is therefore desirable to search for better approximations of optimum covers. Our purpose is to isolate the source of NP-completeness of the optimization problem and to study practical methods that take us closer to optimum covers. In Sect. 4 we prove our main result: the dependencies can be grouped into *equivalence classes* with equivalent left sides so that the classes can be optimized independently in an arbitrary order.

The next natural step towards achieving optimum covers would be to show that within an equivalence class the dependencies can be optimized one at a time. In Sect. 5 we show that this indeed holds for the left sides of the dependencies, but not for the right sides. We further show that while there exist transformations of an equivalence class which make the optimization possible one dependency at a time, no transformation can avoid producing at least one dependency which is about as long as the original equivalence class. These observations suggest a new representation for equivalence classes, in which the class can easily be optimized componentwise.

In Sect. 6 we consider the problem of optimizing a single dependency, i.e. given a set F and a dependency $X \to Y$ in F, finding the shortest dependency $X' \to Y'$ such that $((F - \{X \to Y\}) \cup \{X' \to Y'\})^+ = F^+$. This problem is easily seen to be NP-complete by reduction from the key of cardinality k problem [12]. We give a method for partial optimization of one dependency and investigate its effect on the optimum cover problem.

We begin by a review of notation and previous results. However, we assume that the reader is familiar with the basic concepts of relational database theory (e.g. [15]).

2. Background

As usual, capital letters from the end of the alphabet denote attribute sets, and capital letters from the beginning of the alphabet denote single attributes. Concatenation is used for union.

Let F be a given set of functional dependencies (FDs). We will use $X \xrightarrow[F]{} Y$ to denote the fact that $X \rightarrow Y \in F^+$, i.e. $X \rightarrow Y \in F$ or $X \rightarrow Y$ can be deduced from F using Armstrong's axioms. When F is understood, the subscript F will be dropped.

Two sets of FDs F and G are *equivalent* if $F^+ = G^+$; this will also be denoted by $F \equiv G$. Two attribute sets X and Y are *equivalent* under F, written $X \xleftrightarrow[F]{} Y$, if $X \xrightarrow[F]{} Y$ and $Y \xrightarrow[F]{} X$. This concept induces a partition of the set F into *equivalence classes* on the basis of equivalent left sides: we will denote by $E_F(X)$ the set of dependencies in F whose left side is equivalent with X.

A set M is said to be *minimal*, if there does not exist an equivalent set N properly contained in M. Furthermore, a set M is *minimum*, if there does not exist an equivalent set N such that $|N| < |M|$. These notions apply to attribute sets as well as to dependency sets. In particular, a minimal dependency set is also called nonredundant. Obviously, every minimum set is minimal, but the converse does not hold.

Our interest is in comparing various equivalent dependency sets. Therefore we will introduce some further classes of such sets.

(a) A minimum set of FDs F is *L-minimum*, if for every $X \rightarrow Y$ in F there exists no X' properly contained in X such that $X' \xrightarrow[F]{} Y$.

(b) An L-minimum set F is *LR-minimum*, if for every $X \rightarrow Y$ in F we have $((F - \{X \rightarrow Y\}) \cup \{X \rightarrow Y'\})^+ \neq F^+$ for every Y' properly contained in Y.

(c) A set of FDs F is *optimum*, if there does not exist an equivalent set G with fewer attribute symbols (with repetitions counted).

Obviously, the left sides of the dependencies in these classes are minimal.

The following result relating dependencies of equivalent sets was established by Bernstein.

Proposition 1. *Let F and G be equivalent minimal sets of FDs and let $Z \rightarrow W \in G$. Then there exists a FD $X \rightarrow Y$ in F such that $X \xleftrightarrow[F]{} Z$.*

Proof. [3], Lemma 5.8. □

A useful concept in proving properties of a set F of FDs is the *F-based derivation dag* (DDAG) introduced by Maier [13]. Suppose we want to show that $X \xrightarrow[F]{} Y$; the DDAG for $X \rightarrow Y$ can be constructed as follows. For each $A \in X$, include a node labeled with A into the DDAG. If $Z \rightarrow CW \in F$ and for each $B \in Z$ there already exists a node v_B labeled with B in the DDAG, then a node v_C labeled with C can be added to the DDAG together with the arcs $v_B \rightarrow v_C$. If eventually every attribute of Y labels some node of the DDAG, the DDAG represents a derivation for $X \rightarrow Y$. Figure 1 is an example of one possible F-based DDAG for $BCS \rightarrow GI$ when $F = \{CS \rightarrow ACM,\ BC \rightarrow AD,\ ACM \rightarrow DAG,\ DAG \rightarrow GI\}$. The basic property of DDAGs is that there exists an F-based DDAG for $X \rightarrow Y$ if and only if $X \rightarrow Y \in F^+$. The *use set* of a

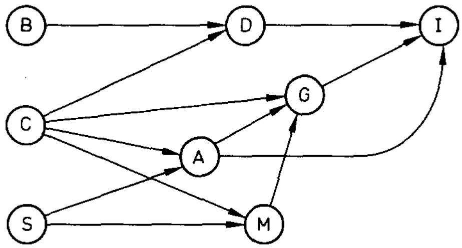

Fig. 1.

DDAG H, denoted by $U(H)$, is the set of dependencies of F used in constructing H. The following elegant proposition of Maier states a useful property of DDAGs.

Proposition 2. *If $X \rightarrow Y$ is in $U(H)$ for some F-based DDAG H of $V \rightarrow Z$, then $V \underset{F}{\rightarrow} X$.*

Proof. [13], Lemma 2. □

Another useful concept based on derivation dags is that of direct determination. Let F be a set of FDs and let $X \underset{F}{\rightarrow} Y$. We say that X *directly determines* Y, denoted by $X \overset{\cdot}{\underset{F}{\rightarrow}} Y$, if there exists an F-based DDAG H for $X \rightarrow Y$ such that $U(H) \cap E_F(X) = \emptyset$. This property is independent of the choice of the cover for F:

Proposition 3. *$X \overset{\cdot}{\underset{F}{\rightarrow}} Y$ if and only if $X \overset{\cdot}{\underset{G}{\rightarrow}} Y$ for every cover G for F.*

Proof. [13], Corollary to Lemma 4. □

Bernstein's result (Proposition 1) shows that when two equivalent sets of dependencies are partitioned on the basis of equivalent left sides, both partitions have the same number of equivalence classes. Using direct determination, it is possible to prove the following important result:

Proposition 4. *Let F and G be equivalent minimum sets of dependencies. Then $|E_F(X)| = |E_G(X)|$ for every X.*

Proof. [13], Theorem 1. □

Thus the corresponding equivalence classes have the same number of dependencies. Moreover, the following holds for the left sides of the dependencies.

Proposition 5. *Let F and G be equivalent minimum sets of dependencies and let $X \rightarrow Y \in F$. There exists a unique $Z \rightarrow W \in G$ such that $X \overset{\cdot}{\longrightarrow} Z$ and $Z \overset{\cdot}{\longrightarrow} X$.*

Proof. [13], p. 670. □

Finally, the following property can be shown for optimum sets of functional dependencies:

Proposition 6. *An optimum set of FDs is LR-minimum.*

Proof. [13], Corollary to Theorem 2. □

3. Size Bounds

Theorem 1. *For all $c>0$ there exists an LR-minimum dependency set F and an optimum cover G of F such that*

$$\frac{\|F\|}{\|G\|} \geqq c.$$

Proof. Let $c>0$ be given. Consider the following set

$$\begin{aligned} F_p = \{ & B_1 B_2 \ldots B_p \to E, \\ & E \to B_1 B_2 \ldots B_p, \\ & A_1 B_1 B_2 \ldots B_p \to C_1, \\ & A_2 B_1 B_2 \ldots B_p \to C_2, \\ & \vdots \\ & A_p B_1 B_2 \ldots B_p \to C_p \}. \end{aligned}$$

F_p is clearly LR-minimum. Let G'_p be the set

$$\begin{aligned} G'_p = \{ & B_1 B_2 \ldots B_p \to E, \\ & E \to B_1 B_2 \ldots B_p, \\ & A_1 E \to C_1, \\ & A_2 E \to C_2, \\ & \vdots \\ & A_p E \to C_p \}. \end{aligned}$$

Clearly $F_p \equiv G'_p$, and for the optimum cover G_p of F_p we have $\|G'_p\| \geqq \|G_p\|$ (in fact $G_p = G'_p$).

We have

$$\|F_p\| = p^2 + 4p + 2$$

and

$$\|G'_p\| = 5p + 2.$$

Thus

$$\frac{\|F_p\|}{\|G_p\|} \geqq \frac{\|F_p\|}{\|G'_p\|} = \frac{p^2+4p+2}{5p+2} \geqq \frac{p^2+4p+2}{7p} \geqq \frac{p}{7} \geqq c,$$

provided $p \geqq \max\{1, 7c\}$. □

This result answers a question posed by Maier: the use of LR-minimum covers as approximations for optimum covers can produce arbitrarily bad approximations. However, the next theorem shows that the size of an LR-minimum cover is less than or equal to the square of the size of an equivalent optimum cover.

Theorem 2. *Let F be an LR-minimum dependency set and G an optimum cover for F. Then*

$$\frac{\|F\|}{(\|G\|)^2} \leqq 1.$$

Proof. Let m_F be the length of the longest dependency $X \to Y$ in F. As G is a cover for F, $X \to Y \in G^+$. Therefore there exists a G-based DDAG H representing a derivation for $X \to Y$; let $U(H)$ be its use set.

As F is LR-minimum and $X \to Y \in F$, neither X nor Y contain unnecessary attributes. Thus for each $A \in X$ there is a dependency $U \to V$ in $U(H)$ such that $A \in U$, and for each $B \in Y$ there is a dependency $U' \to V'$ in $U(H)$ such that $B \in V'$. Thus

$$m_F = |X| + |Y| \leqq \sum_{U \to V \in U(H)} (|U| + |V|) \leqq \|G\|.$$

As F is minimum, we have by Proposition 6 $|F| = |G|$. Also

$$\|F\| \leqq m_F \cdot |F|.$$

Combining these inequalities we get

$$\frac{\|F\|}{(\|G\|)^2} \leqq \frac{m_F \cdot |F|}{(\|G\|)^2} \leqq \frac{m_F \cdot |G|}{(\max\{|G|, m_F\})^2} \leqq \frac{m_F \cdot |G|}{m_F \cdot |G|} = 1. \quad \square$$

4. Equivalence Classes and Optimum Covers

Definition. Let F and G be equivalent minimum sets of FDs and let

$$F = F_1 \cup \ldots \cup F_n$$
$$G = G_1 \cup \ldots \cup G_n$$

be the partitions of F and G on the basis of equivalent left sides. Suppose $F_i = E_F(X_i)$ and $G_i = E_G(Z_i)$ for all $i = 1, \ldots, n$.

The partitions $\{F_1, \ldots, F_n\}$ and $\{G_1, \ldots, G_n\}$ are *matching*, if $X_i \underset{F}{\leftrightarrow} Z_i$ for all $i = 1, \ldots, n$. The partition $\{F_1, \ldots, F_n\}$ is *ascending*, if $X_i \underset{F}{\rightarrow} X_j$ implies $j \leqq i$.

These concepts are independent of the choice of the sets X_i. Note that if $\{F_1, \ldots, F_n\}$ and $\{G_1, \ldots, G_n\}$ are matching partitions and $\{F_1, \ldots, F_n\}$ is ascending, then $\{G_1, \ldots, G_n\}$ is also ascending.

In the sequel we will use the term "partition" to mean partition on the basis of equivalent left sides.

Lemma 1. *Every minimal set has an ascending partition.*

Proof. Let

$$F = F_1 \cup \ldots \cup F_n$$

be a minimal set of FDs and suppose $F_i = E_F(X_i)$ for all $i = 1, \ldots, n$.

Define

$$F_i \leqq F_j$$

iff $X_j \xrightarrow[F]{} X_i$. Then $\leqq$ is a partial order; reflexivity and transitivity are clear, and if antisymmetry failed, we would have $X_i \overset{}{\underset{F}{\leftrightarrow}} X_j$ for $i \neq j$. Thus $\leqq$ can be extended to a total order. Renumber the sets F_i so that

$$F_1 \leqq F_2 \leqq \ldots \leqq F_n.$$

Then $\{F_1, \ldots, F_n\}$ is an ascending partition. □

Lemma 2. *Let F and G be equivalent minimum sets and suppose that*

$$F = F_1 \cup \ldots \cup F_n$$

and

$$G = G_1 \cup \ldots \cup G_n$$

are matching ascending partitions. Then

$$G_i \subseteq (F_1 \cup \ldots \cup F_i)^+$$

for all $i = 1, \ldots, n$.

Proof. Let $Z \to W \in G_i$. As $F \equiv G$, there exists an F-based DDAG H for $Z \to W$. If the use set $U(H)$ of H contains a dependency $X \to Y \in F_j$, then $Z \xrightarrow[F]{} X$ by Proposition 2.

Let $F_i = E_F(Y_i)$; then $Z \underset{F}{\leftrightarrow} Y_i$, as the partitions are matching. Thus we have $Y_i \xrightarrow[F]{} X$, and since F is ascending and $X \to Y \in F_j$, we must have $j \leqq i$. Therefore

$$U(H) \subseteq (F_1 \cup \ldots \cup F_i)^+$$

and thus

$$Z \to W \in (F_1 \cup \ldots \cup F_i)^+. \quad \square$$

The proof of Lemma 2 shows in fact that *no* F-derivation of a dependency $Z \to W$ in G_i can use dependencies from the sets $F_{i+1}, \ldots, F_n$. Let $X \to Y$ be the unique dependency of F given by Proposition 5 such that $Z \dashrightarrow X$ and $X \longrightarrow Z$. Then $X \to Y \in F_i$, and the dependency $Z \to X$ can be derived without using F_i. Combining these results for the class G_1 we get that for all $Z \to W \in G_1$ there exists $X \to Y$ such that $Z \to X$ and the derivation of this dependency

(i) does not use F_1,
(ii) cannot use $F_2, \ldots, F_n$.

Thus $X \to Z$ can be derived without using any dependencies, i.e. $Z \subseteq X$. By symmetry, $X \subseteq Z$ and so $X = Z$. Therefore F_1 and G_1 have exactly same left sides; this holds for any equivalence class whose left sides do not determine the left side of any member of any other class.

We shall next prove a theorem showing that partitioning a dependency set on the basis of equivalent left sides yields a natural grouping of dependencies.

Theorem 3. *Let F and G be equivalent minimum sets of FDs and let*

$$F = F_1 \cup \ldots \cup F_n,$$
$$G = G_1 \cup \ldots \cup G_n$$

be matching partitions. For all $i = 1, \ldots, n$

$$((G - G_i) \cup F_i)^+ = G^+.$$

Proof. By Lemma 1 we may assume without loss of generality that the partitions are ascending.

If $i = 1$, then as

$$((G - G_1) \cup F_1)^+ \subseteq (G \cup F)^+ \subseteq (G^+ \cup F^+)^+ = G^{++} = G^+,$$

and as

$$G = (G - G_1) \cup G_1,$$

only the inclusion

$$G_1 \subseteq ((G - G_1) \cup F_1)^+$$

has to be proven. But Lemma 2 states

$$G_1 \subseteq F_1^+,$$

and thus the claim holds.

Let $i > 1$ and suppose the theorem is true for all F, G and j with $j < i$. This induction hypothesis means that one can replace an equivalence class with number less than i in an ascending partition by the corresponding class of an equivalent minimum set without altering the closure. (Note that this change also yields a minimum set.)

Applying the induction hypothesis repeatedly gives that the covers

$$H_0 = G_1 \cup \ldots \cup G_n,$$
$$H_1 = F_1 \cup G_2 \cup \ldots \cup G_n,$$
$$\vdots$$
$$H_{i-1} = F_1 \cup \ldots \cup F_{i-1} \cup G_i \cup \ldots \cup G_n$$

all have the same closure, namely G^+. Let

$$H_i = F_1 \cup \ldots \cup F_i \cup G_{i+1} \cup \ldots \cup G_n.$$

Applying Lemma 2 to F and G yields

$$G_i \subseteq (F_1 \cup \ldots \cup F_i)^+.$$

Therefore

$$G_i \subseteq ((H_{i-1} - G_i) \cup F_i)^+ = H_i^+,$$

and thus

$$G^+ = H_{i-1}^+ \subseteq H_i^+ \subseteq (F \cup G)^+ = G^+,$$

so the closure of H_i is also G^+.

Using the induction assumption again gives the same closure G^+ for the sets

$$K_i = F_1 \cup \ldots \cup F_{i-1} \cup F_i \cup G_{i+1} \cup \ldots \cup G_n,$$
$$K_{i-1} = F_1 \cup \ldots \cup F_{i-2} \cup G_{i-1} \cup F_i \cup G_{i+1} \cup \ldots \cup G_n,$$
$$= (H_i - F_{i-1}) \cup G_{i-1},$$
$$\vdots$$
$$K_1 = F_1 \cup G_2 \cup \ldots \cup G_{i-1} \cup F_i \cup G_{i+1} \cup \ldots \cup G_n,$$
$$K_0 = G_1 \cup G_2 \cup \ldots \cup G_{i-1} \cup F_i \cup G_{i+1} \cup \ldots \cup G_n.$$

But

$$K_0 = (G - G_i) \cup F_i$$

is the set for which the result was wanted. □

Theorem 3 shows that a set of functional dependencies can be seen as being constructed from sets of dependencies with equivalent left sides. This motivates the following definition.

Definition. Let

$$F = F_1 \cup \ldots \cup F_n$$

be a minimum set of FDs partitioned on the basis of equivalent left sides. The set F_i is an *optimum equivalence class*, if for all sets of FDs H with

$$((F - F_i) \cup H)^+ = F^+$$

we have $\|H\| \geqq \|F_i\|$.

Proposition 6 implies that an optimum equivalence class is also a minimum equivalence class.

Theorem 4. *A minimum cover is optimum if and only if all its equivalence classes are optimum.*

Proof. 1° Only if. Suppose F is a set of FDs with a non-optimum class F_i, i.e. for some H with $\|H\| < \|F_i\|$ we have

$$((F - F_i) \cup H)^+ = F^+.$$

Then

$$\|(F - F_i) \cup H\| < \|F\|$$

and F is not optimum.

2° If. Let

$$F = F_1 \cup \ldots \cup F_n$$

be a minimum set with optimum equivalence classes. If F is not optimum, there exists

$$G = G_1 \cup \ldots \cup G_n$$

(partitioned matchingly with F) such that $\|G\| < \|F\|$. For some i, $\|G_i\| < \|F_i\|$.

But by Theorem 3

$$((F-F_i)\cup G_i)^+ = F^+,$$

contradicting the optimality of the class F_i. □

Maier showed that finding an optimum cover is an *NP*-complete problem. A minimum cover can easily be partitioned in polynomial time, and Theorem 4 shows that an optimum cover can be found by finding (a polynomial number of) optimum equivalence classes. Thus we have the following result.

Corollary. (i) *Finding an optimum equivalence class is an NP-complete problem.* (ii) *If the sizes of the equivalence classes are bounded by a constant, an optimum cover can be found in polynomial time.*

This result has some practical significance, as equivalence classes seldom contain more than a few entries.

Theorem 3 also explains the usefulness of equivalence classes in dealing with functional dependencies: it shows that a set of FDs can be seen as built from separate equivalence classes. Thus it is not surprising that e.g. in [11] equivalence classes are the starting point in constructing an improved third normal form decomposition.

5. Optimization of an Equivalence Class

The previous section showed that a minimum cover can be optimized one equivalence class at a time, and thus finding an optimum equivalence class is an *NP*-complete problem. A natural step forward would be to show that an equivalence class can be optimized a dependency at a time. For *LR*-minimum covers this does not hold, as the following simple example shows.

Let

$$\begin{aligned} F = \{&A \to BC,\\ &B \to AD,\\ &CD \to E,\\ &E \to CD\} \end{aligned}$$

and

$$\begin{aligned} G = \{&A \to BE,\\ &B \to A,\\ &CD \to E,\\ &E \to CD\}. \end{aligned}$$

F and G are clearly equivalent. The equivalence class

$$E_F(A) = \{A \to BC, B \to AD\}$$

cannot be optimized by looking at only one dependency at a time: to reach the optimum cover

$$\{A \to BE, B \to A\}$$

one has to study the whole class.

In this example the difficulties were caused by the optimization of right sides. This is no accident: left sides are easier to optimize. Proposition 5 guarantees that the left sides of an equivalence class of a minimum set and the matching class of an optimum cover correspond to each other in a simple fashion, i.e. the sides directly determine each other. No similar result holds for right sides, as the previous example shows (BC and BE are not equivalent).

Definition. A minimum dependency set F has *optimum left sides*, if for all minimum covers G of F we have

$$\sum_{X \to Y \in F} |X| \leqq \sum_{Z \to W \in G} |Z|.$$

Theorem 5. *An optimum cover has optimum left sides.*

Proof. Let F be an optimum cover and suppose that some minimum cover G of F has shorter left sides than F. Let

$$F = \{X_i \to Y_i | i = 1, \ldots, n\}$$
$$G = \{Z_i \to W_i | i = 1, \ldots, n\}.$$

By Proposition 5 we may assume that $Z_i \overset{\cdot}{\longrightarrow} X_i$ and $X_i \overset{\cdot}{\longrightarrow} Z_i$ for all $i = 1, \ldots, n$. Let k be an index such that $\|Z_k\| < \|X_k\|$. Consider the cover

$$F' = (F - \{X_k \to Y_k\}) \cup \{Z_k \to Y_k\}.$$

By definition of direct determination F' is equivalent to F. But $\|F'\| < \|F\|$, which contradicts the optimality of F. □

Proposition 5 implies thus according to Theorem 5 that local analysis is sufficient for achieving optimum left sides: for each dependency $X \to Y$ find the shortest X' such that $X' \overset{\cdot}{\longrightarrow} X$ and $X \overset{\cdot}{\longrightarrow} X'$, and substitute X' for X. This method gives a dependency set having the same left sides as some optimum cover, but finding X' is an NP-complete task (see Sect. 6). However, the size of X is usually considerably smaller than the size of the whole class, so an exponential algorithm can be practical for finding optimum left sides.

On the other hand, local analysis does not suffice for right sides of an equivalence class. Consider the class

$$\begin{aligned} H = \{&A_1 \to A_2 X_1, \\ &A_2 \to A_3 X_2, \\ &\vdots \\ &A_{n-1} \to A_n X_{n-1}, \\ &A_n \to A_1 X_n\} \end{aligned}$$

and suppose the single attributes A_i do not appear elsewhere in the dependency set. The optimum equivalence class for H is

$$\begin{aligned}\{&A_1 \to A_2\, Y,\\ &A_2 \to A_3,\\ &\vdots\\ &A_{n-1} \to A_n,\\ &A_n \to A_1\},\end{aligned}$$

where Y is the minimum key of $X_1 X_2 \ldots X_n$. Therefore one has to consider the whole equivalence class at the same time; a similar saving in the size of the instance as in the case of left sides is not possible. (Since optimization of one right side is an NP-complete problem (Sect. 6), and finding an optimum equivalence class is in NP, we know that there is a way to optimize an equivalence class using a polynomial number of optimizations of a right side. The above example shows that this method cannot avoid considering almost all attributes in the class.)

This phenomenon suggests that a collection of dependencies is not a natural representation for an equivalence class, as it imposes arbitrary connections of right-hand attributes with particular left sides.

An equivalence class could instead be expressed as a pair (A, Y), where A is the collection of equivalent attribute sets (the left sides of the class) and Y is the set of other attributes determined by these sets. Transformation from the usual representation to this form is easy: for the class $\{X_i \to Y_i | i=1, \ldots, n\}$ use the pair

$$(\{X_i | i=1, \ldots, n\}, Y_1 \cup \ldots \cup Y_n).$$

Note that this transformation does not increase the size of the representation. An equivalence class in the form (A, Y) is easy to optimize componentwise: first optimize each $X \in \mathsf{A}$ and then find the shortest Y' with

$$(\bigcup \mathsf{A}) \cup Y \to Y' \quad \text{and} \quad (\bigcup \mathsf{A}) \cup Y' \to Y.$$

The inverse transform, from the pair representation to the usual one, may increase the size needed for the dependencies. However, there is no need for this transformation, as existing algorithms can be written to use the pair representation for equivalence classes. For example, Beeri and Bernstein's [2] algorithm for calculating the closure of an attribute set X remains almost unchanged. Instead of checking whether there exists a dependency $Z \to W$ whose left side is included in the subset of X^+ already obtained, one looks for a left side of an equivalence class satisfying this condition. All the attributes occurring in the equivalence class are then added to the closure, instead of just W.

Using the (A, Y)-form for an equivalence class seems to correspond quite closely to the way in which actual functional dependencies are intuitively written; one usually has quite clear view of equivalent attribute sets and attributes determined by them.

6. Optimization of One Dependency

Consider the following two problems: given a minimum set F of FDs with a distinguished dependency $X \to Y$ and a positive integer k, does there exist a dependency $X' \to Y'$ with $|X'| \leq k$ ($|Y'| \leq k$) such that

$$(F - \{X \to Y\}) \cup \{X' \to Y'\} \equiv F?$$

These problems will be called left side of cardinality k problem and right side of cardinality k problem, respectively.

These problems can easily be seen to be NP-complete by using the key of cardinality k problem [12]: given a set G of FDs and a positive integer k, does there exist a key of length less than or equal to k for the set U of all attributes occurring in G?

Note first that G can be assumed to be minimum, as minimum covers can be found in polynomial time and as the existence of a key depends only on the closure of the set. This restricted version of the key of cardinality k problem is reduced to the left (right) side of cardinality k problem simply by taking an attribute A not in U and considering the set

$$G \cup \{U \to A\}$$
$$(G \cup \{A \to U\}).$$

U has a key of cardinality less than or equal to k if and only if the left side of $U \to A$ (the right side of $A \to U$) can be optimized to length less than or equal to k.

Thus the left (right) side of cardinality k problem is NP-hard. It is in NP, as the left (right) side can be guessed and the equivalence condition tested in polynomial time.

Although optimization of a side of a dependency is an NP-complete task, there seems to be a useful approximation method based on equivalence classes. The idea is to use only the shortest element of any equivalence class $E_F(X)$ outside the set of dependencies needed for the formation of $E_F(X)$ (i.e. those FDs needed to derive the equivalences among left sides of dependencies in $E_F(X)$). For example, in the set F_p occurring in the proof of Theorem 1 this method would substitute E for $B_1 \dots B_p$ everywhere except in the dependencies $E \to B_1 \dots B_p$ and $B_1 \dots B_p \to E$. Thus F_p would be transformed to G_p.

This method corresponds to the common way of using attributes like EMPLOYEE_NUMBER, which are artificial one-attribute keys for certain entity sets. In a dependency set describing a payroll application we might encounter dependencies

EMPLOYEE_NUMBER → E_NAME E_ADDRESS X

E_NAME E_ADDRESS → EMPLOYEE_NUMBER,

where X is the set of other attributes connected to an employee. Outside this equivalence class only the attribute EMPLOYEE_NUMBER is used, not the pair E_NAME E_ADDRESS.

Returning to the approximation theme, using this method even in combination with LR-minimality does not improve the situation from Theorem 1: the approximations can be arbitrarily bad. For consider first the set

$$\begin{aligned} F=\{&B_1 A \to C_1 C_2,\\ &C_1 C_2 \to B_1 A,\\ &B_2 A \to C_3 C_4,\\ &C_3 C_4 \to B_2 A,\\ &D C_1 C_2 C_3 C_4 \to E\}. \end{aligned}$$

F is LR-minimum and it cannot be shortened by substituting shorter sides from an equivalence class. However, F is not optimum, as the dependency $DC_1 C_2 C_3 C_4 \to E$ could be replaced by $DAB_1 B_2 \to E$. This example can be inflated to produce arbitrarily bad examples by denoting

$$\begin{aligned} H_p=\{&B_1 A_1 \ldots A_p \to C_{1,1} \ldots C_{1,p+1},\\ &\vdots\\ &B_{p+1} A_1 \ldots A_p \to C_{p+1,1} \ldots C_{p+1,p+1},\\ &C_{1,1} \ldots C_{1,p+1} \to B_1 A_1 \ldots A_p,\\ &\vdots\\ &C_{p+1,1} \ldots C_{p+1,p+1} \to B_{p+1} A_1 \ldots A_p\},\\ F_p=H_p \cup \{&D_1 C_{1,1} \ldots C_{1,p+1} C_{2,1} \ldots C_{p+1,1} \ldots C_{p+1,p+1} \to E_1,\\ &\vdots\\ &D_p C_{1,1} \ldots C_{1,p+1} C_{2,1} \ldots C_{p+1,1} \ldots C_{p+1,p+1} \to E_p\},\\ G_p=H_p \cup \{&D_1 A_1 \ldots A_p B_1 \ldots B_{p+1} \to E_1,\\ &\vdots\\ &D_p A_1 \ldots A_p B_1 \ldots B_{p+1} \to E_p\}. \end{aligned}$$

The set F_p is LR-minimal and it cannot be shortened by substituting shorter sides from an equivalence class. Clearly $F_p \equiv G_p$. We still have

$$\begin{aligned} \|F_p\| &= p^3 + 6p^2 + 11p + 4,\\ \|G_p\| &= 6p^2 + 11p + 4. \end{aligned}$$

and thus

$$\lim_{p \to \infty} (\|F_p\| / \|G_p\|) = \infty.$$

The technique of inflating an example to produce arbitrarily bad approximations used above and in the proof of Theorem 1 seems to apply to other approximation methods as well. We conjecture that there does not exist a polynomial time approximation algorithm for optimum covers which does not produce arbitrarily bad approximations.

7. Conclusions

The relationship of minimum and optimum covers for a set of functional dependencies has been studied. We have shown that *LR*-minimum covers can produce arbitrarily bad approximations for optimum covers. The source of *NP*-completeness of the optimum cover problem was shown to be the optimization of a single equivalence class. Analysis of this problem revealed a difference between the behavior of left and right sides of a dependency, and we suggested a new representation for equivalence classes (and thus for sets of FDs).

These results have shown that finding optimum covers is not intractable in practice. They also throw more light to the importance of equivalence classes in the theory of functional dependencies.

References

1. Aho, A.V., Beeri, C., Ullman, J.D.: The theory of joins in relational databases. ACM Trans. Database Syst. **4**, 3, 297–314 (1979)
2. Beeri, C., Bernstein, P.A.: Computational problems related to the design of normal form relational databases. ACM Trans. Database Syst. **4**, 1, 30–59 (1979)
3. Bernstein, P.A.: Normalization and functional dependencies in the relational data base model. Tech. Rep. CSRG-60, Computer Systems Research Group, Univ. of Toronto 1975
4. Bernstein, P.A.: Synthesizing third normal form relations from functional dependencies. ACM Trans. Database Syst. **1**, 4, 277–298 (1976)
5. Biskup, J., Dayal, U., Bernstein, P.A.: Synthesizing independent database schemas. Proc. ACM-SIGMOD Int. Conf. on Management of Data, pp. 143–151, 1979
6. Codd, E.F.: A relational model of data for large shared data banks. Commun. ACM **13**, 6, 377–387 (1970)
7. Codd, E.F.: Further normalization of the data base relational model. In: Data Base Systems. Rustin, R. (ed.). Englewood Cliffs: Prentice-Hall, pp. 33–64, 1972
8. da Silva, A.: The decomposition of relations based on relational dependencies. Ph.D. Dissertation, Univ. of California, Los Angeles 1980
9. Jou, J.H.: Theory of functional relation schemes in relational databases. Ph.D. Dissertation, Dept. of Computer Science, The Pennsylvania State University 1980
10. Kambayashi, Y., Tanaka, K., Yajima, S.: Problems of relational database design. In: Data Base Design Techniques I: Requirements and Logical Structures. Yao, S.B., Narathe, S.B., Weldon, J.L., Kunii, T.L. (eds.). Berlin, Heidelberg, New York: Springer, pp. 172–218, 1982
11. Ling, T.-W., Tompa, F.W., Kameda, T.: An improved third normal form for relational databases. ACM Trans. Database Syst. **6**, 2, 329–346 (1981)
12. Lucchesi, C.L., Osborne, S.L.: Candidate keys for relations. J. Comput. Syst. Sci. **17**, 2, 270–279 (1978)
13. Maier, D.: Minimum covers in the relational database model. J. ACM **27**, 4, 664–674 (1980)
14. Tsou, D.-M., Fischer, P.C.: Decomposition of a relation scheme into Boyce-Codd normal form. Proc. ACM Annual Conf., pp. 411–417, 1980
15. Ullman, J.D.: Principles of Database Systems. Potomac, MD: Computer Science Press 1980

Received March 15, 1983 / June 14, 1983