Acta Cybernetica **9** (4) (1990), 385-402

Georg Gottlob
Leonid Libkin

Investigations on Armstrong Relations, Dependency Inference and Excluded Functional Dependencies

Investigations on Armstrong Relations, Dependency Inference, and Excluded Functional Dependencies

Georg Gottlob and Leonid Libkin*

Department of Applied Computer Science†
University of Technology
Vienna – Austria

Abstract

This paper first presents some new results on excluded functional dependencies, i.e., FDs which do not hold on a given relation schema. In particular, we show how excluded dependencies relate to Armstrong relations, and we state criteria for deciding whether a set of excluded dependencies characterizes a set of FDs. In the rest of the paper, complexity issues related to the following three problems are studied : to construct an Armstrong relation for a cover F of functional dependencies (FDs), to construct a cover of FDs that hold in a relation R (dependency inference), and, given a cover F and a relation R, to decide if all the FDs that hold in R can be derived from F. The first two problems are known to have exponential complexity. We give a new proof for the second problem by showing that dependency inference can be used to compute all keys of a relation instance. We prove that the third problem is co-$\mathcal{NP}$-complete. Further, it is shown that the problems can be solved in polynomial time if it is known that a relation scheme satisfies some additional properties, which are polynomially recognizable themselves.

*Current mailing address of the second author: Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104, USA

†Mailing address: Institut für Angewandte Informatik, TU Wien, Paniglgasse 16, A-1040 Wien, Austria. Internet e-mail of first Author: gottlob@vexpert.at

1 Introduction

In order to express the information conveyed by a set of functional dependencies (FDs) that hold on a relation scheme, one can alternatively specify the set of all dependencies that *do not* hold on the scheme. These dependencies, called excluded functional dependencies (XFDs), are closely related to Armstrong relations. Note, however, that not every arbitrary set of XFDs corresponds to a set of FDs. In this paper we therefore introduce the notion of completeness of sets of XFDs. Informally, a set of XFDs is complete if it unambiguously characterizes a set of FDs. We also present completeness criteria which can be tested in polynomial time.

In the rest of the paper we study complexity issues related to several problems concerning *functional dependencies* (FDs for short) in relational databases. The three problems which we are interested in are the following.

Problem 1 (Constructing Armstrong Relation) [BDFS84], [MR86] *Given a set F of FDs, construct an Armstrong relation R for F.*

Problem 2 (Dependency Inference Problem) [MR87], [MR90] *Given a relation R, construct a cover F of FDs that hold in R.*

Problem 3 (FD-Relation Implication Problem) *Given a relation R and a set F of FDs, decide whether all the FDs that hold in R can be derived from F.*

The first two problems are of high practical importance, see [BDFS84, MR86, MR87, MR89]. However, it is known that these problems are inherently exponential and hence it is impossible to design polynomial algorithms for their solution [BDFS84, MR87, MR86]. The third problem seems to be important for design theory too. To our knowledge, its complexity is still unknown. We show that the problem of finding all the minimal keys of a relation instance can be polynomially transformed to the second problem. Then we prove that the Problem 3 is co-NP-complete.

Let us introduce a new problem which is close to the Problem 3.

Problem 4 (FD-Relation Equivalence Problem) *Given a relation R and a set F of FDs, decide whether the sets of FDs that hold in R and that can be derived from F coincide. In other words: decide whether R is an Armstrong Relation for F.*

This problem can be decomposed into two subproblems:

- *Decide whether all the FDs that hold in R can be derived from F, i.e., whether $F_R \subseteq F^+$.* Note that this subproblem is identical to Problem 3; and
- *Decide whether each FD of F also holds in R, i.e., whether $F^+ \subseteq F_R$.* Note that this subproblem is easily solvable in polynomial time.

Problem 4 thus consists of the conjunction of a co-$\mathcal{NP}$-complete subproblem and a polynomially decidable subproblem. Unfortunately, this knowledge does not allow us to determine its complexity.

It seems rather difficult to find the complexity class of Problem 4. To our best knowledge, this problem has never been dealt with in the literature. We therefore want highlight the complexity analysis of Problem 4 as an interesting open problem to which we plan to dedicate further research efforts.

We show that the complexity of Problems 1-4 becomes polynomial if it is known that F satisfies certain additional properties. These additional properties will be formulated for a set F of FDs and for the associated closure operator and semilattice. We also show that these properties can be recognized in polynomial time.

The paper is organized as follows. In Section 2 we state some basic definitions. In Section 3 we derive our new results concerning excluded functional dependencies. In Section 4 we show that the key-generating problem for relation instances can be solved by using dependency inference. The fifth Section is dedicated to the proof of the co-$\mathcal{NP}$-completeness of Problem 3. In Section 6 we study special cases in which our four problems become polynomial. Some concluding remarks are made in Section 7.

2 Basic Definitions

In this section we briefly remind the necessary concepts of relational database theory (cf. [Ma83], [PBGV89]) and state some preliminary results.

Let U be a set of *attributes*. With each attribute $A \in U$ associate its domain $D(A)$. A *relation* (or *relation instance*) over U is a subset of $\prod_{A \in U} D(A)$. We can think of a relation as being a set of tuples $t : U \rightarrow \bigcup_{A \in U} D(A)$ with $t(A) \in D(A)$ for each $A \in U$. Note that some authors distinguish between the terms "relation" and "relation instance" while here both terms have the same meaning.

If X and Y denote sets of attributes and A denotes an attribute, we often write XY, XA, $X - A$, etc. instead of respectively $X \cup Y$, $X \cup \{A\}$, $X - \{A\}$, etc.

A FD is an expression of form $X \rightarrow Y, X, Y \subseteq U$. We say that FD $X \rightarrow Y$ holds in R if for every $t_1, t_2 \in R, t_1(A) = t_2(A)$ for all $A \in X$ implies that $t_1(A) = t_2(A)$ for all $A \in Y$.

The set of all FDs that hold for a given relation R is denoted by F_R. F_R satisfies the following properties : $X \rightarrow Y \in F_R$ for all $Y \subseteq X$(pseudoreflexivity), and $XZ \rightarrow V \in F_R$ if $X \rightarrow Y \in F_R$ and $YZ \rightarrow V \in F_R$ (pseudotransitivity).

If we are given a set F of FDs, F^+ stands for the set of all FDs that can be derived from F by the above rules being used. Of course, for each relation R, $F_R^+ = F_R$. Furthermore, for each set F of functional dependencies, there is a relation R with $F^+ = F_R$; such a relation is called *Armstrong Relation* [FA82].

A set F of FDs is called a *cover* of G if $F^+ = G^+$. A cover F is called *nonredundant* if for each $f \in F$ we have $f \notin (F - f)^+$. A cover F is called *minimum* if $|F| \leq |F'|$ for all other covers F'.

It is well-known that each set F of FDs is equivalent to a set F' of FDs containing only single attributes as right hand sides. Indeed, each FD $X \rightarrow A_1 A_2 \ldots A_n$ can be replaced by the following n FDs: $X \rightarrow A_1$, $X \rightarrow A_2$ ..., $X \rightarrow A_n$. Therefore, we can always assume without loss of generality that a given set of FDs has only single attributes as right hand sides.

A set X is called a *key* if $X \rightarrow U \in F^+$. A key is called *minimal* if each $Y \subset X$ is not a key.

A pair $< U, F >$ is called a *relation scheme*, or RS for short. A RS is in *Boyce-Codd normal form* (BCNF) if for each $X \rightarrow A \in F^+$, where $A \notin X$, it holds: $X \rightarrow U \in F^+$.

Given a set F of FDs, define the mapping $C_F(X) = \{A \in U : X \rightarrow A \in F^+\}$ (we will write C_R instead of C_{F_R}). Then C_F is a *closure*, that is, $X \subseteq C_F(X), X \subseteq Y$ implies $C_F(X) \subseteq C_F(Y)$ and $C_F(C_F(X)) = C_F(X)$. If F is understood then $C_F(X)$ is also denoted by X^+.

The following well-known algorithm computes the closure $C_F(X)$ of a set of attributes X. Here we assume that F has only single attributes as right hand sides.

Algorithm CLOSURE
Input: a set F of FDs over U
and a set $X \subseteq U$ of attributes.
Output: $C_F(X)$
Method:
result := X;
WHILE there exists an attribute $A \in U$ such that
$A \notin$ *result* AND
there is a FD $Y \rightarrow A \in F$ such that $Y \subseteq$ *result*
DO *result* := *result* $\cup A$;
RETURN(*result*).

A set X is *closed* (w. r. t. C_F) if $C_F(X) = X$. Denote by S_F the family of all closed sets (again, we write S_R instead of S_{F_R}). Then $U \in S_F$ and S_F is a *semilattice*, i.e. $X, Y \in S_F$ implies $X \cap Y \in S_F$.

A set $X \in S_F$ is called (meet)-*irreducible* if $X = Y \cap Z$, $Y, Z \in S_F$ imply $X = Y$ or $X = Z$. The family of all irreducible sets is denoted by $GEN(F)$. Notice that the usual mathematical notation for $GEN(F)$ is $M(S_F)$, but we adopt the terminology of database theory here.

$GEN(F)$ is the unique minimal subfamily of generators in S_F such that each member of S_F can be expressed as an intersection of sets in $GEN(F)$ (where the set U is considered to be the intersection of an empty collection of sets).

It has been shown by Mannila and Räihä [MR86] that for a set F of FDs on U it holds that

$$GEN(F) = MAX(F) = \bigcup_{A \in U} MAX(F, A)$$

where $MAX(F, A) = \{Y \subseteq U \;:\; Y$ is a nonempty maximal set (with respect to $\subseteq$) such that $Y \rightarrow A \notin F^+\}$.

In [MR86] an algorithm is presented which computes an Armstrong relation R for a given FD-set F from $GEN(F)$ in time polynomial in the size of $GEN(F)$. On the other hand, if R is a given relation, then the MAX-sets for F_R, and hence also $GEN(F_R)$, can be computed in polynomial time (this follows easily from results in [BDFS84], [MR86], [MR87]).

Each $X \in MAX(F, A)$ can be written and interpreted as *excluded functional dependency (XFD)* with maximal left hand side, i.e., as an expression $X \not\to A$ such that $\forall B \in U - X : XB \to A$.

3 Some Results on Excluded Functional Dependencies

Excluded functional dependencies (in a similar way as MAX-sets) are just an alternative way of representing the information conveyed by a cover F of functional dependencies. When we speak about sets of excluded FDs we always assume that these FDs have single attributes as right hand sides, that the right hand side attribute of an XFD does not occur in the left hand side of the same XFD, and that all left hand sides corresponding to the same right hand side are maximal w.r.t. set inclusion, i.e., the set contains no pair of distinct XFDs $X \not\to A, Y \not\to A$, such that $X \subseteq Y$.

Excluded functional dependencies appear to be more intuitive than MAX-sets. However, when dealing with excluded FDs, some care has to be taken. If a set $\mathcal{X}$ of XFDs on a set of attributes U is given, we wish that this set represents *all* those dependencies which do not hold in a given situation. The corresponding set of all FDs which do hold is then represented by the cover:

$$F_{\mathcal{X}} = \{X \to A \ : \ X \subseteq U \ \wedge A \in U \ \wedge \ A \notin X \ \wedge \ \nexists Y \not\to A \in \mathcal{X} : X \subseteq Y\}.$$

Consider for example the set of excluded FDs $\mathcal{X} = \{AB \not\to C, AC \not\to B, B \not\to A, C \not\to A\}$ defined on a set of attributes $U = ABC$. Then $F_{\mathcal{X}} = \{BC \to A\}$. It is, however, important to note that there exist sets $\mathcal{X}$ of excluded FDs with maximal left hand sides, for which $F_{\mathcal{X}}$ is "unreasonable" because it implies FDs which should be forbidden (i.e. excluded) according to $\mathcal{X}$. The following example displays such a situation.

Consider a set $\mathcal{X}$ containing a single excluded FD $\mathcal{X} = \{B \not\to A\}$ defined on a set of attributes $U = ABC$. Then $F_{\mathcal{X}}$ is equivalent to the cover $\{C \to A, A \to B, C \to B, A \to C, B \to C\}$ Of course the FD $B \to A$ follows from $F_{\mathcal{X}}$; hence this FD is both excluded and requested. It can be seen that such situations arise when a set of excluded FDs is incomplete, in the sense that some necessary excluded FDs (in our case, for instance, $C \not\to A$ or $B \not\to C$) are missing. Let us therefore define the notion of *complete set of XFDs*.

A set $\mathcal{X}$ of excluded FDs is *complete* if $F_{\mathcal{X}}$ does not imply any excluded FD, i.e., if no FD $X \to A$ can be derived from $F_{\mathcal{X}}$, such that $X \not\to A \in \mathcal{X}$.

According to the semantics we give to sets of XFDs, only complete sets of XFDs make sense. Indeed, if a set of XFDs is incomplete, then it expresses that certain FDs are both forbidden and valid.

The following theorem relates complete XFD-sets to MAX-sets.

Theorem 1 *Let $\mathcal{X}$ be a set of XFDs defined on a set of attributes U. Let $RHS(\mathcal{X}, A) = \{X : X \not\to A \in \mathcal{X}\}$ for each $A \in U$. $\mathcal{X}$ is a complete set of XFDs iff $\forall A \in U : RHS(\mathcal{X}, A) = MAX(F_{\mathcal{X}}, A)$.*

Proof. *if.* Assume that $\forall A \in U : RHS(\mathcal{X}, A) = MAX(F_{\mathcal{X}}, A)$. Each $MAX(F_{\mathcal{X}}, A)$, by definition, contains only sets of attributes which do not determine A w.r.t. $F_{\mathcal{X}}$. Thus there cannot be any FD

$X \to A$ which follows from $F_\mathcal{X}$ such that X is equal to any element of $MAX(F_\mathcal{X}, A) = RHS(\mathcal{X}, A)$. Hence $\mathcal{X}$ is complete.

only if. Let $\mathcal{X}$ be a complete set of XFDs.

- We show that $\forall A \in U: \; RHS(\mathcal{X}, A) \subseteq MAX(F_\mathcal{X}, A)$.
  Assume that for some $A \in U$, $RHS(\mathcal{X}, A) \not\subseteq MAX(F_\mathcal{X}, A)$. Then there exists a XFD $X \not\to A \in \mathcal{X}$ such that $X \notin MAX(F_\mathcal{X}, A)$. X must be a (proper) subset of some element Y of $MAX(F_\mathcal{X}, A)$, otherwise $X \to A$ would hold, and $\mathcal{X}$ would not be complete. Thus there is an $Y \subseteq U$ with $XY \in MAX(F_\mathcal{X}, A)$ and $Y \neq \emptyset$ and $Y \cap XA = \emptyset$. On the other hand, since the XFD $X \not\to A$ of $\mathcal{X}$ has a maximal left hand side, it must hold by definition of $F_\mathcal{X}$ that $XY \to A \in F_\mathcal{X}$. This is in contradiction to $XY \in MAX(F_\mathcal{X}, A)$. We thus have shown that $RHS(\mathcal{X}, A) \subseteq MAX(F_\mathcal{X}, A)$.
- We show that $\forall A \in U: \; MAX(F_\mathcal{X}, A) \subseteq RHS(\mathcal{X}, A)$.
  Assume that for some $A \in U$, $MAX(F_\mathcal{X}, A) \not\subseteq RHS(\mathcal{X}, A)$. Then there exists $X \in MAX(F_\mathcal{X}, A)$ such that $X \notin RHS(\mathcal{X}, A)$. There are two cases to consider. In the first case X is not a subset of any element of $RHS(\mathcal{X}, A)$. Then $X \to A \in F_\mathcal{X}$. Contradiction to $X \in MAX(F_\mathcal{X}, A)$. In the second case, X is a proper subset of some $Y \in RHS(\mathcal{X}, A)$. Since $X \in MAX(F_\mathcal{X}, A)$ and Y is a proper superset of X the FD $Y \to A$ can be derived from $F_\mathcal{X}$; but $Y \not\to A$ is an excluded FD in $\mathcal{X}$. Thus $\mathcal{X}$ is not complete. Contradiction. Hence $MAX(F_\mathcal{X}, A) \subseteq RHS(\mathcal{X}, A)$.

The theorem is proved. □

If a set $\mathcal{X}$ of XFDs is complete, then an Armstrong relation R for $F_\mathcal{X}$ can be computed in polynomial time: Construct $GEN(F_\mathcal{X})$ by uniting all sets $RHS(\mathcal{X}, A)$ and then apply the polynomial algorithm of [MR86] to construct an Armstrong relation for $F_\mathcal{X}$ from $GEN(F_\mathcal{X})$. Note also that the cardinality of $F_\mathcal{X}$ can be exponential in the cardinality of $\mathcal{X}$.

Assume that a set $\mathcal{X}$ of XFDs on a set of attributes U is given. Assume furthermore that one has to compute the closure $C_{F_\mathcal{X}}(X)$ of a set of attributes $X \subseteq U$. One way is to compute first $F_\mathcal{X}$ and then use the CLOSURE algorithm as described in Section 2. However, this is not advisable since the size of $F_\mathcal{X}$ may be exponential in the one of $\mathcal{X}$. Fortunately there is a much simpler way of computing $C_{F_\mathcal{X}}(X)$. The following algorithm XFD-closure computes $C_{F_\mathcal{X}}(X)$ directly from $\mathcal{X}$ and X:

Algorithm XFD-CLOSURE
Input: a set $\mathcal{X}$ of XFDs over U
and a set $X \subseteq U$ of attributes.
Output: $C_{F_\mathcal{X}}(X)$
Method:
$result := X$;
WHILE there exists an attribute $A \in U$ such that
$A \notin result$ AND
there is no XFD $Y \not\to A \in \mathcal{X}$ such that $result \subseteq Y$
DO $result := result \cup A$;
RETURN($result$).

Theorem 2 *The* XFD-CLOSURE *algorithm applied to* $\mathcal{X}, U$, *and* X *effectively computes* $C_{F_{\mathcal{X}}}(X)$.

Proof. Let U be a set of attributes, let $A \in U$, and let $\mathcal{X}$ be a set of XFDs on U. By definition of $F_{\mathcal{X}}$, the following statements (1) and (2) are equivalent:

(1) there is a FD $Y \to A \in F_{\mathcal{X}}$

(2) there is no XFD $Z \not\to A$ in $\mathcal{X}$ such that $Y \subseteq Z$.

Now let *result* be an arbitrary subset of U. It follows that the following statements (1') and (2') are equivalent:

(1') there is a FD $Y \to A \in F_{\mathcal{X}}$ such that $Y \subseteq result$

(2') there is no XFD $Y \not\to A$ in $\mathcal{X}$ such that $result \subseteq Y$.

Indeed, (1') is equivalent to the statement $result \to A \in F_{\mathcal{X}}$ which in turn is equivalent to (2').

Now consider the XFD-CLOSURE algorithm for $\mathcal{X}$ and note that condition (2') occurs in the body of the algorithm. If we replace this condition with condition (1') we get exactly the body of the CLOSURE algorithm for $F_{\mathcal{X}}$. Hence the output of the XFD-CLOSURE algorithm is $C_{F_{\mathcal{X}}}(X)$. □

¿From the above theorem it follows that for each set $X \subseteq U$, $C_{F_{\mathcal{X}}}(X)$ can be computed in polynomial time from $\mathcal{X}$ and U. Moreover, the XFD-CLOSURE algorithm can be used as a tool for testing in polynomial time whether a given set $\mathcal{X}$ of XFDs is complete. Indeed, the following criterion follows trivially from the definition of completeness:

Completeness Criterion A *A set* $\mathcal{X}$ *of XFDs is complete iff for each XFD* $X \not\to A \in \mathcal{X}$, $A \notin C_{F_{\mathcal{X}}}(X)$.

Obviously, the test $A \notin C_{F_{\mathcal{X}(X)}}$ can be performed by using the XFD-CLOSURE algorithm.

Let us now derive a simple sufficient (but not necessary) condition for the completeness of a set $\mathcal{X}$ of XFDs:

Completeness Criterion B *A set* $\mathcal{X}$ *of XFDs is complete if for each XFD* $X \not\to A \in \mathcal{X}$ *and for each* $B \in U - (XA)$ *there is an XFD* $Y \not\to B \in \mathcal{X}$ *such that* $X \subseteq Y$.

Proof. Assume that Criterion B is satisfied. Let $X \not\to A$ be an XFD of $\mathcal{X}$. Note that the XFD-CLOSURE algorithm applied to $\mathcal{X}$ and X stops immediately with output X. Hence $C_{F_{\mathcal{X}}}(X) = X$. Therefore, by Completeness Criterion A, we conclude that $\mathcal{X}$ is complete. □

We will use this criterion in the proof of a theorem in Section 5.

Let us now make a remark which emphasizes the importance of the notion of completeness. Assume that an incomplete set of XFDs is given. We will show that such a set, in general, can be extended to several different (minimal) complete sets of XFDs. Hence incomplete sets of XFDs do not contain enough information for characterizing FD-families unambiguously. We will show this on hand of a simple example.

Consider again the set $\mathcal{X}$ containing a single excluded FD $\mathcal{X} = \{B \not\rightarrow A\}$ defined on a set of attributes $U = ABC$. We have already seen that this set is incomplete. We can extend $\mathcal{X}$ to a complete set either by enlarging the lhs of its XFD, yielding $\mathcal{X}_1 = \{BC \not\rightarrow A\}$, or by adding another XFD, yielding $\mathcal{X}_2 = \{B \not\rightarrow A, B \not\rightarrow C\}$. It can be easily seen by applying Completeness Criterion B that both $\mathcal{X}_1$ and $\mathcal{X}_2$ are complete. Of course $\mathcal{X}_1$ and $\mathcal{X}_2$ correspond to different sets of FDs $F_{\mathcal{X}_1}$ and $F_{\mathcal{X}_2}$. Furthermore, $\mathcal{X}_1$ and $\mathcal{X}_2$ are both minimally complete in the sense that any omission of an attribute or of an XFD would result in incompleteness.

We conclude this Section by making a few comments on related work. Excluded FDs are also studied by Thalheim in [Tha88] where their use for database design is motivated; moreover [Tha88] introduces the notion of *excluded multivalued dependency (XMVD)* and states derivation rules for FDs, MVDs, XFDs, and XMVDs. The notion of *functional independency* which is similar to the one of an XFD has been introduced by Janas [Ja88, Ja89]. Janas analyzes covers consisting of both, FDs and functional independencies. According to Janas, a set G of FDs and functional independencies is free of contradictions if there is no FD $X \rightarrow Y$ such that both $X \rightarrow Y$ and $X \not\rightarrow Y$ are implied by G. This concept seems to be close to the one of completeness; there is, however, a main difference between our approach and the one of Janas: We make the *closed world assumption* to sets of XFDs but Janas does not make this assumption for sets of functional independencies. For example, in the setting of Janas, the set $\{B \not\rightarrow A\}$ is free of contradictions, while in our setting this set is incomplete and thus expresses contradictory information.

4 Generating all Keys of a Relation Instance

The Dependency Inference Problem (Problem 2) is inherently exponential. Mannila and Räihä [MR87] show an example of a relation instance R containing $O(n)$ tuples, where $n = |U|$, such that there is a minimum cardinality cover F of F_R containing $O(2^{n/2})$ FDs. Nevertheless, a useful and practical algorithm for inferring dependencies from relation instances is developed in [MR87]. This algorithm has demonstrated a satisfactory efficiency when being used for "real-life" database design problems.

We will now show that the problem of finding all keys of a relation instance can be polynomially transformed to the Dependency Inference Problem. This transformation is useful because it allows to use highly practical algorithms for dependency inference (such as the one presented in [MR87]) for generating all keys to a given relation instance.

As a by-product of our polynomial transformation we also get a new proof for the exponential complexity of dependency inference. This complexity result follows directly from our transformation and from a well known result on the complexity of key-generation. Consider the following algorithm.

Algorithm *Input:* a relation $R = \{t_1, \ldots, t_m\}$ over U.
Output: a set F of FDs.

Step 1. Find the *equality set* $E_R = \{E_{ij} : 1 \leq i < j \leq m\}$, where $E_{ij} = \{A \in U : t_i(A) = t_j(A)\}$.

Step 2. Find the maximal sets among $E_R - \{U\}$. Denote them by $X_1, \ldots, X_p$.

Step 3. Construct a family $\{X_i - A : A \in U, i = 1, \ldots, p\}$ and denote its elements by $Y_1, \ldots, Y_r$. Suppose $Y_0 = U$.
Step 4. Construct a relation $R' = \{t'_0, ..., t'_r\}$ where

$$t'_i(A) = \begin{cases} 0 & \text{if } A \in Y_i \\ i & \text{otherwise, } A \in U, i = 1, ..., r \end{cases}$$

Step 5. Using the algorithm for solving the dependency inference problem, find a cover F' of $F_{R'}$.
Step 6. Find a minimum cover F of F'.

Clearly, all the steps except step 5 require polynomial time in $|R|$, that is, in $n \cdot m$. For a discussion and characterization of the equality sets E_R and E_{ij} see [DT88].

Theorem 3 *The output F of the above algorithm consists of FDs $K_1 \rightarrow U, \ldots, K_l \rightarrow U$, where $K_1, \ldots, K_l$ are all the minimal keys of R.*

Proof. According to [DT88], $X_1, \ldots, X_p$ are so-called *antikeys*, i.e. maximal nonkeys. According to [MR86], R' is a relation whose antikeys are $X_1, \ldots, X_p$ and by [BDK, theorem 3] the families of keys of R and R' coincide. Moreover, by [DHLM89] $F_{R'}$ is in BCNF, and hence its minimum cover consists of FDs $K_i \rightarrow U$ for $K_i, i = 1, \ldots, l$, the minimal keys of R'. □

It is shown in [MR87] that in many cases the algorithm solving dependency inference problem may work efficiently. In these case one can use the above algorithm to find the minimal keys of a relation. Remind, that this problem is inherently exponential as the number of keys of a given relation instance can be exponential in the size of the instance [BDFS84,DT87]. The last mentioned fact together with theorem 3 implies

Corollary 1 *The dependency inference problem has exponential complexity.* □

5 Deciding $F_R \subseteq F^+$ is Co-NP-Complete

In this Section we turn our attention to Problem 3. It is possible to show that this problem (FD-Relation Implication Problem) is co-$\mathcal{NP}$-complete. In order to do this, we will first define another problem and prove its co-$\mathcal{NP}$-completeness and then show the polynomial transformability of that problem to our problem.

The problem we will first consider can be described as follows:

Name: SUBSET DELIMITER COMPLEMENTARITY (SDC)
Instance: a finite set S, a collection $G_1 \ldots G_n$ of subsets of S, and a collection $D_1 \ldots D_m$ of subsets of S.
Question: Is it true that $\forall X \subseteq S :$ ($(\exists i, 1 \leq i \leq n : G_i \subseteq X)$ or $(\exists j, 1 \leq j \leq m : X \subseteq D_j)$) ?

In order to show the co-$\mathcal{NP}$-completeness of SDC, we will use the MONOTONE 3SAT problem which is known to be $\mathcal{NP}$-complete [Go78, GJ79]:

> **Name:** MONOTONE 3SAT (M3SAT)
> **Instance:** a finite set U of propositional variables and a collection C of clauses over U such that each clause contains exactly three literals and each clause contains either only negated or only un-negated literals.
> **Question:** Is there a satisfying truth assignment for C ?

Theorem 4 *The SDC Problem is co-$\mathcal{NP}$-complete.*

Proof. It is easy to see that the problem is in co-$\mathcal{NP}$. In order to show that its solution is negative, guess a subset $Z \subseteq S$ nondeterministically such that Z is neither a superset of any G_i nor a subset of any D_j.

Let us now show that the complement of M3SAT can be reduced polynomially to our problem. Consider an instance (U, C) of M3SAT. Assume without loss of generality that C consists of k clauses $C_1 \ldots C_k$ such that the first n clauses are positive and the remaining m clauses are negative (with $m = k - n$).

We construct an instance of the SDC problem from (U, C) as follows. Let $S = U$. For each $1 \leq j \leq n$ let $D_j = U - C_j$ and for each $1 \leq i \leq m$ let $G_i = \{p : \neg p \in C_{n+i}\}$. Clearly the D_j and G_i can be constructed in polynomial time from C.

In the sequel of this proof, any truth value assignment for the propositional variables of U is represented as the subset of U consisting of all those propositional variables which are assigned "true".

C is unsatisfiable, iff for each truth value assignment $\tau \subseteq U$ there exists a clause C_i, $1 \leq i \leq k$ such that C_i is falsified by τ. In particular:

• A positive clause $C_j \in C$ is falsified by τ iff no propositional variable appearing in τ also appears in C_j, i.e., iff $\tau \subseteq U - C_j = D_j$.

• A negative clause $C_i \in C$ is falsified by τ iff all propositional variables occurring in C_i (in negated form) have truth value "true" under τ, i.e., iff $G_{i-n} \subseteq \tau$.

Thus C is unsatisfiable iff for each $\tau \subseteq S$, it holds that $(\exists i, 1 \leq i \leq n : G_i \subseteq \tau)$ or $(\exists j, 1 \leq j \leq m : \tau \subseteq D_j)$. We thus have polynomially transformed the complement of the M3SAT problem to the SDC problem. This completes our proof. □

The following Corollary shows the co-$\mathcal{NP}$-completeness of a slightly stronger version of the SDC problem.

Corollary 2 *The SDC problem remains co-$\mathcal{NP}$-complete even if it is restricted to those instances for which the family of sets D_j is an antichain, i.e., no D_j is a subset of a D_i, for $i \neq j$ and $1 \leq i, j \leq m$.*

Proof. Consider an instance of *SDC* whose sets D_j do not form an antichain. By eliminating all those D_j which are contained in any other D_i, we get an equivalent instance satisfying our restriction. Of course this transformation can be done in polynomial time. □

We are now ready for proving our complexity result for Problem 2.

Theorem 5 *It is co-$\mathcal{NP}$-complete to decide whether for a given relation (instance) R and for a given set F of FDs it holds that $F_R \subseteq F^+$.*

Proof. Clearly the problem is in co-$\mathcal{NP}$. Indeed, in order to show that $F_R \not\subseteq F^+$ it is sufficient to guess nondeterministically an FD which is in F_R (testable in polynomial time) but which is not in F^+ (again testable in polynomial time). Let us now show completeness in co-$\mathcal{NP}$.

Consider an instance of the SDC problem consisting of a set S and of families of subsets $G_1 \dots G_n$ and $D_1 \dots D_m$. According to Corollary 2 we may assume that the sets $D_1 \dots D_m$ form an antichain.

¿From this instance we will construct a set F of FDs and a set $\mathcal{X}$ of XFDs as follows. Let us view the elements of S as attributes and consider a new attribute $A \notin S$. In the sequel of this proof, all FDs and XFDs are defined on the set of attributes $S' = S \cup \{A\}$.

Let $F = \{G_i \rightarrow A \ : \ 1 \le i \le n\}$ and
let $\mathcal{X} = \{D_j \not\rightarrow A \ : \ 1 \le j \le m\} \cup \{(S' - B) \not\rightarrow B \ : \ B \in S\}$.

Note that the set $F_{\mathcal{X}}$ contains only FDs with right hand side A. More precisely, $F_{\mathcal{X}}$ consists of all FDs of the form $X \rightarrow A$ such that $X \subseteq S$ and $X \not\subseteq D_j$ for $1 \le j \le m$. Furthermore, $F_{\mathcal{X}}^+$, besides the trivial FDs over S', contains exactly the FDs of $F_{\mathcal{X}}$. (This follows from the fact that the pseudotransitivity rule cannot be applied to the FDs of $F_{\mathcal{X}}$ in order to generate new nontrivial FDs.)

On the other hand, the set F^+ consists of all FDs $X \rightarrow A$ such that X is a superset of some G_i with $1 \le i \le n$ plus the trivial FDs over S'.

¿From these observations it follows that $F_{\mathcal{X}}^+ \subseteq F^+$ iff each subset of S which is not a subset of any D_j is a superset of some G_i. In other words, $F_{\mathcal{X}}^+ \subseteq F^+$ iff our SDC Problem-instance has a positive solution.

Since the D_j ($1 \le j \le m$) form an antichain, the XFDs of $\mathcal{X}$ all have maximal left hand sides. Moreover, the set $\mathcal{X}$ of XFDs satisfies the Completeness Criterion B of Section 3. Hence $\mathcal{X}$ is complete and a relation instance R can be found in polynomial time such that $F_R = F_{\mathcal{X}}^+$. Now our SDC problem instance has a positive solution iff $F_R \subseteq F^+$.

We thus have shown how an instance of the SDC problem can be transformed into an instance of the FD-Relation Implication Problem (Problem 3). It is immediately verifiable that this transformation can be performed in polynomial time in the size of the given SDC instance. It follows that Problem 3 is co-$\mathcal{NP}$-complete. □

Of course, the converse problem, that is, to check up if $F^+ \subseteq F_R$, can be solved in polynomial time. However, as pointed out in the introduction, it is still unknown if the problem 4 (FD-Relation Equivalence Problem) is polynomially solvable or not.

Here we show that if F does not contain FDs with small left-hand sides then both problems 3 and 4 can be solved in polynomial time.

Proposition 1 *Suppose for each $X \to Y \in F$ one has $|U| - |X| \leq k$, where k is a constant. Then both problems 3 and 4 can be solved in polynomial time.*

Proof. Given a relation instance R and a set $X \subseteq U$, to find $C_R(X)$ requires polynomial time in $|R|$. Hence we can check in polynomial time if $C_R(X) = X$ for all X with $|U| - |X| = k - 1$. Since S_R is a semilattice, for each nontrivial FD $X \to Y \in F_R$ it holds that $|U| - |X| \leq k$. Therefore, to make sure that $F_R \subseteq F^+$, we just have to consider all sets X with $|U| - |X| \leq k$ (there are less than $|U|^k$) and to check that $C_R(X) \subseteq C_F(X)$. □

6 Complexity of the Main Problems : Special Cases

As it has been shown at the end of the previous section, the problem which is generally co-$\mathcal{NP}$-complete can be solved in polynomial time if some additional properties hold. This fact leads us to the idea to study several special types of relation schemes in order to find out if problems 1-4 are polynomial for these relation schemes.

In this section we are going to study three types of relation schemes. All these types have already been investigated more or less widely. We formulate the properties for a relation scheme $< U, F >$ and for its associated closure L_F and semilattice S_F.

Property 1 *There is a cover of F consisting of unary FDs, i.e. of FDs of type $A \to B, A, B \in U$.*

Property 2 *There is a cover of F of type $\{X_1 \to A_1, \ldots, X_r \to A_r\}$ such that $X_1 \subseteq \ldots \subseteq X_r$.*

Property 3 *A relation scheme $< U, F >$ is in BCNF.*

The properties 1 and 3 seem to be simply explained from the practical point of view, note that property 3 is very desirable. Property 2 is interesting from a mathematical point of view because it corresponds to a relevant class of semilattices and closures.

First, we establish the equivalent formulations of the main properties.

Proposition 2 *Given a relation scheme $< U, F >$, the following are equivalent:*
1) $< U, F >$ satisfies property 1,
2) C_F is topological, i.e. $C_F(X \cup Y) = C_F(X) \cup C_F(Y)$,
3) S_F is a distributive lattice.

The proof is straightforward. □

Proposition 3 *([DLM89]) Given a relation scheme $< U, F >$, the following are equivalent:*
1) $< U, F >$ satisfies property 2,

2) C_F is separatory, that is, if $C_F(X) \neq X$ and $C_F(Y) \neq Y$, then $C_F(X \cap Y) \neq X \cap Y$,
3) S_F is separatory, that is, $\mathbf{2}^U - S_F$ is a semilattice again. □

Proposition 4 *([DHLM89]) Given a relation scheme $< U, F >$, the following are equivalent:*
1) $< U, F >$ satisfies property 3,
2) For each $X \subseteq U$ either $C_F(X) = X$ or $C_F(X) = U$,
3) $S_F - \{U\}$ is an ideal of $\mathbf{2}^U$, i.e. if $Y \subseteq X \in S_F - \{U\}$, then $Y \in S_F$. □

Further we will show that some considered problems can be solved in polynomial time if it is known that a relation scheme satisfies property 1 or 2 or 3. Hovewer, in order to use an algorithm solving a problem in a special case one has to make sure that either scheme or relation satisfies the required property. Therefore, it would be desirable if all the properties 1-3 could be recognized in polynomial time. The next Theorem shows that this fact holds.

Theorem 6 *All the properties 1-3 are polynomially recognizable for both relation schemes and relations.*

Proof. Property 1. a) for relation schemes. It is almost obvious that unary FDs cannot be derived from other FDs. Hence, a relation scheme satisfies property 1 iff a nonredundant cover of F consists of unary FDs only.

b) For relations. Given a relation R, we can find $GEN(F_R)$ in polynomial time in $|R|$, see [DT88]. Let us first prove that F_R satisfies property 1 iff $X \cup Y \in S_R$ for every $X, Y \in GEN(F_R)$. Really, if F_R satisfies property 1, then it follows from proposition 2 that $X \cup Y = C_R(X) \cup C_R(Y) = C_R(X \cup Y)$ and $X \cup Y \in S_R$. Conversely, if $X \cup Y \in S_R$ for every $X, Y \in GEN(F_R)$, consider arbitrary $V, W \in S_R$. Suppose $V = X_1 \cap \ldots \cap X_k, W = Y_1 \cap \ldots \cap Y_l$, where $X_1, \ldots, X_k, Y_1, \ldots, Y_l \in GEN(F_R)$. Then $V \cup W = (X_1 \cap \ldots \cap X_k) \cup (Y_1 \cap \ldots \cap Y_l) = \bigcap_{i=1}^{k} \bigcap_{j=1}^{l} (X_i \cup Y_j) \in S_R$, i.e. C_R is topological. Since to find a closure C_R requires polynomial time, the above property can be checked polynomially.

Property 2. a) For relation schemes. First we prove that if a relation scheme $< U, F >$ satisfies property 2 and $X \to A, Y \to B \in F^+$ then either $X \cap Y \to A \in F^+$ or $X \cap Y \to B \in F^+$, where $A \notin X, B \notin Y$. Really, if it is not true, then $A, B \notin C_F(X \cap Y)$. Hence, both $X \cup C_F(X \cap Y)$ and $Y \cup C_F(X \cap Y)$ are nonclosed, and by proposition 3 $C_F(X \cap Y) = (X \cup C_F(X \cap Y)) \cap (Y \cup C_F(X \cap Y))$ is nonclosed, a contradiction.

Suppose without loss of generality that F consists of FDs $X \to A$, where A is an attribute. Hence, if a relation scheme satisfies property 2, for every two FDs $X \to A, Y \to B \in F$ either $(F - \{X \to A\}) \cup \{X \cap Y \to A\}$ or $(F - \{Y \to B\}) \cup \{X \cap Y \to B\}$ is a cover of F. Since the membership problem for FDs is polynomial [Ma83], we need only the following to finish the proof: if we are given a family $\mathcal{A} = \{X_1, \ldots, X_k\}$ of subsets of U, and by one step we can change either X_i or X_j to $X_i \cap X_j$, then $\mathcal{A}$ can be transformed to a chain by a polynomial number of steps.

First we show how to transform $\mathcal{A}$ to $\mathcal{A}' = \{X'_1, \ldots, X'_k\}$ where $X'_i = X_i$ for some i and $X'_j \subseteq X'_i$ for all $j \neq i$. We use induction on k.

If $\mathcal{A}$ contains unique maximal element X_i, we are done. If X_i, X_j are two maximal elements of $\mathcal{A}$, consider $\mathcal{A} - \{X_i\}$ and transform it to $\mathcal{A}^0 = \{X^0_l : l \neq i\}$ where $X^0_p = X_p$ for some p and $X^0_l \subseteq X^0_p$ for all $l \neq i$. If $X_p \subseteq X_i$, we are done. If X_i and X_p are incomparable, consider all the pairs $\{X_i, X_l^0\}, l \neq i$. If for some l we can change X_i to $X_i \cap X_l^0$, then $\mathcal{A}' = \mathcal{A}^0 \cup \{X_i \cap X_l^0\}$. If for all the pairs we can only change X_l^0 to $X_i \cap X_l^0$, then $\mathcal{A}' = \{X_i\} \cup \{X_i \cap X_l^0 : l \neq i\}$.

If $k = 2$, it takes one step to transform $\mathcal{A}$ to a chain. Since each ith iteration takes no more than i additional steps, it takes $O(k^2)$ steps to transform $\mathcal{A}$ to $\mathcal{A}'$. Then, if we apply the above algorithm to $\mathcal{A}' - \{X_i\}$ etc, we obtain a chain by no more than $k-1$ iterations. Hence, $\mathcal{A}$ can be transformed to a chain by $O(k^3)$ steps being used. This shows the polynomiality of the recognition of property 2 for relation schemes.

b) For relations. It follows immediately from proposition 3 that if F_R satisfies property 2, then all the elements of $GEN(F_R)$ have cardinality $n, n-1$ or $n-2$. Moreover, S_R is separatory if and only if the matrix $a = \| a_{ij} \|, i, j = 1, ..., n$:

$$a_{ij} = \begin{cases} 1 & \text{if } U - \{A_i, A_j\} \in S_R \\ 0 & \text{otherwise,} \end{cases}$$

where $U = \{A_1, \ldots, A_n\}$ is *absolutely determined*, that is, each submatrix of a has a saddle point [GL90]. The last property can be checked in time $O(n^4)$ [GL90].

Property 3. a) For relation schemes. It is wellknown that the BCNF property of relation schemes can be tested in polynomial time. It can be shown, for instance, as follows. It is almost evident that a relation scheme $< U, F >$ is in BCNF iff its minimum cover consists of FDs $\{K_i \to U, i = 1, \ldots, l\}$, where $K_i, i = 1, \ldots, l$, are the minimal keys of $< U, F >$. Since to find a minimum cover takes polynomial time [Ma83], and testing whether a set of arttributes is a minimal key also takes polynomial time, BCNF can be recognized in polynomial time.

b) For relations. See [DHLM89] for a polynomial algorithm.

The proof is complete. □

Now we are ready to present the main result about the complexity of problems 1-4 if it is known that a relation scheme $< U, F >$ (or $< U, F_R >$ if input is R) satisfies additional properties.

Theorem 7 *The problems 1-4 can be solved in polynomial time if it is known that a relation scheme $< U, F >$ (for problems 1,3,4) or $< U, F_R >$ (for problem 2) satisfies property 1 or 2.*

Proof. Property 1. The polynomiality of constructing Armstrong relation was proved in [MR89], the polynomiality of the other problems is almost evident.

Property 2. a) Problem 1. According to the proof of previous theorem (see also [GL90]) $GEN(F)$ can be computed in polynomial time. Applying algorithm of [MR86, p.136], we find an Armstrong relation.

b) Problem 2. We use the concepts of $nec(A)$ and $gendep(A)$ (see [MR87]). Let $R = \{t_1, \ldots, t_n\}$ be a relation over U. Let $disag(i,j) = \{A \in U : t_i(A) \neq t_j(A)\}$ and $nec(A) = \{disag(i,j) - A : A \in disag(i,j)\}$. Suppose $gendep(A) = \{\{A_1, \ldots, A_r\} \to A : A_i \in X_i, i = 1, \ldots, r\}$, where $nec(A) = \{X_1, \ldots, X_r\}$. Then $\bigcup\{gendep(A) : A \in U\}$ is a cover of F_R. Suppose $X_A = \bigcap(\{A_1, \ldots, A_r\} : A_i \in$

$X_i, i = 1, \ldots, r$). If a relation scheme $< U, F_R >$ satisfies property 2, it follows from the proof of Theorem 6 that $\{X_A \to A : A \in U\}$ is a cover of F_R. Clearly, $B \in X_A$ iff $\{B\} = X_i$ for some $X_i \in nec(A)$, and $nec(A)$ can be computed in polynomial time. Therefore, it takes polynomial time to find a cover of F_R.

c) Problems 3-4. According to [DLM89], $F_R \subseteq F^+$ iff $S_F \subseteq S_R$, or iff $GEN(F) \subseteq S_R$. Since $GEN(F)$ can be computed in polynomial time, the checking of the last condition takes polynomial time too.

The theorem is completely proved. □

Property 1 can be easily generalized if we allow FDs $X \to A$ with $|X| < k$, $k > 1$. However, as the following theorem shows, it is impossible to get a polynomiality result for Problem 1 w.r.t. such relation schemes.

Proposition 5 *Problem 1 has exponential complexity even if it is known that a relation schema $< U, F >$ satisfies the property: for each FD $X \to A \in F^+$ there is an FD $Y \to A \in F^+$ with $Y \subseteq X$ and $|Y| < k$, $k > 1$.*

Proof. In [BDFS84] an example of a RS with $k = 2$ was constructed that satisfies the above property and provides a minimal Armstrong relation exponential in the number of FDs. □

Finishing this section, we discuss the complexity of the main problems for relations and relation schemes in BCNF.

Let $< U, F >$ be a relation scheme in BCNF. We can think without loss of generality that F consists of FDs $K_i \to U, i = 1, \ldots, l$, where $K_i, i = 1, \ldots, l$, are the minimal keys (if not, we compute a minimum cover in polynomial time). Let R be an Armstrong relation for $< U, F >$. Then we can find *antikeys*, that is, maximal nonkeys [Thi86], in polynomial time in $|R|$, see [DT88]. Conversely, if we have the family of antikeys, we can construct an Armstrong relation for $< U, F >$ according to the algorithm of section 2. Thus, we obtain

Proposition 6 *Problem 1 for relation schemes in BCNF is polynomially equivalent to finding the antikeys of a family of minimal keys.* □

The last problem was discussed in [Thi86]. The problem is inherently exponential. Hovewer, it can be solved in polynomial time, with some additional conditions being added.

Proposition 7 *Problem 1 for relation schemes in BCNF can be solved in polynomial time if the number of minimal keys is bounded by a constant.*

Proof. It follows from [Thi86] and proposition 6. □

Now we prove an auxiliary result.

Proposition 8 *Problem 2 can be solved in polynomial time if the number of tuples of a relation is bounded by a constant.*

Proof. Let m be the number of tuples of a relation R. Then $nec(A)$ contains no more that m^2 sets (see the proof of theorem 7), and $gendep(A)$ has no more that n^{m^2} FDs. Hence, a cover of F_R can be computed in polynomial time. □

Corollary 3 *If the number of tuples of a relation is bounded by a constant, it takes polynomial time to find all its minimal keys.*

Proof. According to [DT88], the number of antikeys is no more than m^2, where m is the number of tuples of R. Hence, the number of minimal keys is no more than $n \cdot m^2$. By proposition 8, we can compute a cover of F_R in polynomial time, and, by [LO78], given a relation scheme, we can find its minimal keys in polynomial time in size of input and output. Hence, the minimal keys of R can be found in polynomial time. □

Now we immediately obtain from theorem 6, proposition 7, and corollary 3:

Proposition 9 *The problems 3 and 4 can be solved in polynomial time for a relation scheme in BCNF if either the number of minimal keys or the number of tuples of a relation is bounded by a constant.* □

We can demonstrate another example providing the problem 4 to be polynomial for relation schemes in BCNF. Remind that an antichain $\mathcal{A}$ is called *saturated* [BDK87,Thi86] if $\mathcal{A} \cup \{X\}$ is not antichain for every $X \notin \mathcal{A}$.

Proposition 10 *Let $< U, F >$ be a relation scheme in BCNF and R a relation in BCNF. If either the family of minimal keys of $< U, F >$ or the family of antikeys of R is saturated, the problem 4 can be solved in polynomial time.*

Proof. Let the family $\{K_1, \ldots, K_l\}$ of the minimal keys of $< U, F >$ be saturated. Find in polynomial time the family $\{X_1, \ldots, X_r\}$ of antikeys of R [DT88]. Then $\{X_1, \ldots, X_r\}$ is the family of antikeys of $\{K_1, \ldots, K_l\}$ iff for all $i = 1, \ldots, l$, K_i is a minimal set that is not contained in some $X_j, j = 1, \ldots, r$. Clearly, the last condition can be checked in polynomial time. If the family of antikeys of R is saturated, the proof is the same. □

Several criteria providing the families of minimal keys and antikeys to be saturated are established in [Thi86].

7 Conclusion

In this paper we have investigated several aspects of Armstrong relations, dependency inference, and excluded functional dependencies. In particular, we have characterized those sets of excluded dependencies which effectively correspond to sets of FDs (and hence to Armstrong relations). We have shown that the problem of findings all minimal keys of a given relation instance can be solved by using practical algorithms for dependency inference. We proved that the problem whether all FDs that are valid in a given relation instance R do follow from a given cover F is co-$\mathcal{NP}$-complete. Finally, we have analyzed several conditions under which the main problems become polynomially solvable.

One relevant problem remains open: given a relation instance R and a cover F of FDs, what is the complexity of deciding whether $F_R = F^+$? This problem is important; it can be reformulated as follows: what is the complexity of recognizing that a given relation is an Armstrong relation for a given set of FDs. We plan to dedicate further research to this problem.

ACKNOWLEDGMENTS. The authors are grateful to Maddalena Boschetti, Thomas Eiter, and Ernesto Noce for useful comments and corrections to the first version of the manuscript.

References

[BDFS84] C.Beeri, M.Dowd, R.Fagin and R.Statman,
On the structure of Armstrong relations for functional dependencies,
J.Assoc. Comput. Mach. **31** (1984), 30-46.

[BDK87] G.Burosch, J.Demetrovics and G.O.H.Katona,
The poset of closures as a model of changing databases,
Order **4** (1987), 127-142.

[DHLM89] J.Demetrovics, G.Hencsey, L.O.Libkin and I.B.Muchnik,
Normal form relation schemes : a new characterization, *Manuscript.*

[DLM89] J.Demetrovics, L.O.Libkin and I.B.Muchnik,
Functional dependencies and the semilattice of closed classes,
MFDBS 89, Springer LNCS **364** (1989), 136-147.

[DT87] J.Demetrovics and V.D.Thi, Keys, antikeys and prime attributes,
Annales Univ. Sci. Budapest Sect. Comp. **8** (1987), 35-52.

[DT88] J.Demetrovics and V.D.Thi, Some results about functional dependencies,
Acta Cybernetica **8** (1988), 273-278.

[FA82] R. Fagin, Horn Clauses and Database Dependencies, *Journal of the ACM* **29:4** (1982), 952-985.

[GJ79] M.R. Garey and D.S. Johnson, Computers and Intractability - A Guide to the Theory of NP-Completeness, Freeman and Company, New York, 1979.

[Go78] E.M. Gold, Complexity of Automaton Identification from Given Data, *Information and Control*, **37** (1978), 302-320.

[GL90] V.A.Gurvich and L.O.Libkin, Absolutely determined matrices,
to appear in *Math. Soc. Sci.*

[Ja88] J.M.Janas, On Functional Independencies, In: *Foundations of Software Technology and Theoretical Computer Science*, K.V. Nori and S. Kumar Eds., Springer LNCS *338* (1988) 487-508.

[Ja89] J.M.Janas, Covers for Functional Independencies, In: *Proceedings of the MFDBS 89 Conference*, J.Demetrovics and B. Thalheim Eds., Springer LNCS **364** (1989) 254-268.

[LO78] C.L.Lucchesi and S.L.Osborn, Candidate keys for relations,
J. of Computer and System Sciences **17** (1978), 270-279.

[Ma83] D.Maier, *"The Theory of Relational Databases"*, Comp.Sci.Press, Rockville, MD, 1983.

[MR86] H.Mannila and K.-J.Räihä, Design by example: an application of Armstrong relations,
J. of Computer and System Sciences **33** (1986), 126-141.

[MR87] H.Mannila and K.-J.Räihä, "Algorithms for Inferring Functional Dependencies" (Extended Abstract), Proceedings of the Thirteenth International Conference on Very Large Data Bases, Brighton, September 1987.

[MR89] H.Mannila and K.-J.Räihä,
Practical algorithms for findiding prime attributes and testing normal forms,
PODS 89, pp. 128-133.

[MR 90] H.Mannila and K.-J.Räihä,
On the Complexity of Inferring Functional Dependencies, manuscript, 1990.

[PBGV89] J.Paredaens, P.De Bra, M.Gyssens and D.Van Gucht,
The Structure of the Relational Database Model, Springer-Verlag, Berlin, 1989.

[Tha88] B. Thalheim, Logical Relational Database Design Tools Using Different Classes of Dependencies, *J. of New Generation Comput. Syst*, *1*:3 (1988), 211-228.

[Thi86] V.D.Thi, Minimal keys and antikeys, *Acta Cybernetica* **7** (1986), 361-371.