Normal Form Relation Schemes: A New Characterization[*]

János Demetrovics [†] Gusztáv Hencsey[†] Leonid Libkin[‡]
Ilya Muchnik[§]

Abstract

A new characterization of relational database schemes in normal forms is given. This characterization is based on the properties of the semilattice of closed sets of attributes. For the problems testing third and Boyce-Codd normal forms, which are known to be $\mathcal{NP}$-complete for relation schemes, this new characterization helps establish polynomial algorithms if the input is a relation (matrix) rather than a relation scheme. The problem of approximation of an arbitrary family of functional dependencies by one in a normal form is also addressed.

1 Introduction

The relational datamodel defined by E.F. Codd remains one of the most powerful database models. In this model a relation is just a matrix in which rows correspond to records and columns to attributes. Theoretical and practical aspects of this model have been studied over the past 20 years. Database design has always been and still is among the most important aspects attracting the attention of almost all database theorists. For relational databases, the design theory is based on the well-developed theory of dependencies and constraints. Functional dependencies, being the simplest and easiest to understand, underwent a deep investigation. Enormous number of papers on functional dependencies have been written, [10,11,18,22,23,24] being just examples of surveys referring to hundreds of other papers and books. Surprisingly enough, many issues in dependency theory, lying on the very surface, have not been paid attention to for many years. One of them is the lattice-theoretic approach to the study of functional dependecies. It was observed very early that families of functional dependencies correspond to closure operators and to semilattices, but very little has been done in order to bring the methods and tools of lattice

[*]Research partially supported by Hungarian Foundation for Scientific Research (OTKA), Grant 2575. While working on the revised version, L. Libkin was partially supported by NSF Grants IRI-86-10617 and CCR-90-57570 and ONR Grant NOOO14-88-K0634.

[†]Computer and Automation Institute, P.O.Box 63, Budapest H-1502, Hungary; E-mail: h935dem@ella.hu and h103hen@ella.hu.

[‡]Department of Computer and Information Science, University of Pennsylvania, Philadelphia PA 19104, USA; E-mail: libkin@saul.cis.upenn.edu.

[§]24 Chestnut St., Waltham MA 02145, USA; E-mail: ilya@darwin.bu.edu.

theory to the database theory. The situation started changing a few years ago, and in a number of papers functional dependencies were investigated from the lattice theoretic point of view [4,6,7,25]. For example, an easily described relationship between relations and irreducible elements of the semilattice of closed sets made it possible to design a polynomial algorithm for a problem that is well-known to be $\mathcal{NP}$-complete if the input is a relation scheme rather than a relation, see [8,9].

Functional dependencies are closely related to *normalization* of relations or relations schemes. Being in a normal form, or normalized, means that a family of functional dependencies satisfies certain properties. The basic idea behind normalization is that a relational database must be unambigously reconstructed from some of its projections which are normalized. Databases in normal forms are easy to work with, and normal forms are well motivated from the practical point of view, see [5,22,24].

However, to the best of the authors' knowledge, no attempts have been made to apply lattice theoretic techniques, used for functional dependencies, to normalization. We think that doing so would benefit both normalization theory by looking at normalization from a new point of view, and lattice theoretic approach to the study of functional dependencies by extending it to normalization.

The main goal of this paper is to give a lattice theoretic characterization of relation schemes in normal forms, i.e. to describe normal form relation schemes by the semilattices of closed sets they generate. Doing only this would be of little interest. We prefer to view the characterization theorems as important tools in demonstrating advantages of the lattice theoretic approach. In this paper we are going to elaborate on two points:

- the lattice characterization of normal forms will enable us to prove that two problems related to normalization, which are known to be $\mathcal{NP}$-complete for relation schemes, are solved in polynomial time for relations (i.e. databases themselves);

- it will turn out that arbitrary families of functional dependencies can be approximated by normalized ones, and these approximations are effectively computable for relations; for relation schemes, however, it may take exponential time to find approximations.

Let us give a brief sketch of the rest of the paper. The next section contains all necessary definitions and facts. Most of them are standard, but some are not. We define all the concepts because we feel that a paper in an area where different people use slightly different terminology and completely different notation, must be self-contained.

Section 3,4 and 5 deal with the second, third and Boyce-Codd normal forms, respectively. (First normal form basically says that a database is just a relation. Therefore, functional dependency families can be characterized in one word – arbitrary). For each normal form we consider four problems:

- a lattice-theoretic characterization;

- closure properties in the lattice of families of functional dependencies;

- algorithms testing relations and relation schemes for this normal form and their complexity;

- approximation of arbitrary relation schemes by those in normal forms.

Characterisation theorems will be stated in the following form: a relation scheme is in normal form iff the semilattice of sets closed under the closure operation induced by the given scheme satisfies certain properties.

It is known that closure operations on an arbitrary set form a lattice [4,7]. We will show the properties of the subsets of this lattice corresponding to normal form schemes.

Before giving the results on complexity, recall that the *prime attribute problem* is to decide whether a given attribute is prime, i.e. belongs to a minimal key. It is not a complete description of the prime attribute problem for we did not indicate what the input is - a relation scheme or a relation. In the first case the problem is known to be $\mathcal{NP}$-complete [12,17]. However, it was shown in [8] that the problem becomes polynomial for relations. It was done by using the representation of irreducible elements of the semilattice of closed sets which can be obtained from a relation in polynomial time.

Two important problems related to normalization – 3NFTEST and BCNFTEST – are known to be $\mathcal{NP}$-complete [1,14]. They test whether a given relation scheme or its subscheme is in third or Boyce-Codd normal form. Using the techniques similar to those in [8] and our lattice characterization or normal forms, we shall prove that these problems can be solved in polynomial time if the input is a relation.

By *approximation* we mean finding a relation scheme that approximates a given one. "Approximates" should be explained here. First, the approximation must be taken from the class in which it is sought (otherwise the name would not be justified!). Second, it must be greater than the given scheme in some sense. Here we use the ordering on the families of functional dependencies (or closures) introduced and studied in [4,7] to define "greater". Finally, it is desired that approximation be unique. Uniqueness, as it will be shown, depends on closure properties of the given normal form, and can be guaranteed for third and Boyce-Codd normal forms.

If we want to find an approximation, we would like to know the complexity of an algorithm. It will turn out that the situation here resembles the one in the case of testing normal forms: for relations there exist polynomial algorithms, while for relation schemes the problems are superpolynomial[1], provided that $\mathcal{P} \neq \mathcal{NP}$.

2 Basic definitions and results

In this section, that we shall try to make as concise as possible, all definitions and facts to be used in the sequel will be given. Theorems in this section will have negative numbers so that our first result is theorem 1.

Let $\mathcal{U} = \{A_1, \dots, A_n\}$ be a set of attributes. With each attribute A_i associate a domain of its values $dom(A_i)$. A *relation* over $\mathcal{U}$ is a subset of Cartesian product of all $dom(A_i)$'s. Relations will be usually denoted by R, possibly with indices. Alternatively, we can think of a relation R as being a set of maps $h : \mathcal{U} \to \bigcup_i dom(A_i), h(A_i) \in dom(A_i)$ rather than a set of tuples. This does not change the nature of relations, but often makes the notation easier. $R = \{h_1, \dots, h_m\}$ means that R is a relation consisiting of m tuples/maps $h_1, \dots, h_m$.

A *functional dependency* (FD for short) is an expression $X \to Y$, where $X, Y \subseteq \mathcal{U}$. If $A \in \mathcal{U}$, we shall write $X \to A$ instead of $X \to \{A\}$. A FD $X \to Y$ *holds* in a relation $R = \{h_1, \dots, h_m\}$ if for any $h_i, h_j \in R$ the following holds: $\forall A \in Y : h_i(A) = h_j(A)$ whenever $\forall A \in X : h_i(A) = h_j(A)$. A family of FDs $F_R = \{X \to Y : X \to Y \text{ holds in } R\}$ is called a *full family of FDs.*

[1] That is, there are no polynomial algorithms that solve these problems.

Let $\mathcal{P}(\mathcal{U})$ be the powerset of $\mathcal{U}$. We can think of $\rightarrow$ as being a binary relation on $\mathcal{P}(\mathcal{U})$, thus representing a family of functional dependencies as a binary relation (i.e. a set of pairs (X, Y)) as well. A subset F of $\mathcal{P}(\mathcal{U}) \times \mathcal{P}(\mathcal{U})$ is called an *f-family* if the following (Armstrong's Axioms) hold:

(F1) $(X, X) \in F$;
(F2) $(X, Y) \in F, (Y, Z) \in F$ imply $(X, Z) \in F$;
(F3) $(X, Y) \in F, X \subseteq X', Y' \subseteq Y$ imply $(X', Y') \in F$;
(F4) $(X, Y) \in F, (Z, V) \in F$ imply $(X \cup Z, Y \cup V) \in F$.

For any binary relation $F \subseteq \mathcal{P}(\mathcal{U}), F^+$ stands for the minimal binary relation containing F and satisfying (F1)-(F4). The existence of F^+ is ensured by the fact that f-families are closed under intersection. F_R is an f-family for any relation R, i.e. $F_R^+ = F_R$.

A map $L : \mathcal{P}(\mathcal{U}) \rightarrow \mathcal{P}(\mathcal{U})$ is called a *closure* if it is expanding, monotone and idempotent, i.e. $X \subseteq L(X), X \subseteq Y$ implies $L(X) \subseteq L(Y)$ and $L(L(X)) = L(X)$. For a binary relation F on $\mathcal{P}(\mathcal{U})$ define $L_F(X) = \{A \in \mathcal{U} : (X\{A\}) \in F^+$. L_F thus defined is known to be a closure. If $F = F_R$, we write L_R instead of L_{F_R}.

A family of subsets $S \subset \mathcal{P}(\mathcal{U})$ is called a *(meet)-semilattice* if it is closed under intersection, i.e. $X, Y \in S$ implies $X \cap Y \in S$. Given a closure L, define $S_L = \{X \subseteq \mathcal{U} : L(X) = X\}$ and $F_L = \{(X, Y) : Y \subseteq L(X)\}$. The elements of S_L are called *closed* sets. Given a semilattice S containing $\mathcal{U}$, define a map L_S on $\mathcal{P}(\mathcal{U})$ by $L_S(X) = \bigcap\{Y : Y \in S, X \subseteq Y\}$.

Theorem -3 *a) $F \subseteq \mathcal{P}(\mathcal{U}) \times \mathcal{P}(\mathcal{U})$ is an f-family iff there is a relation R such that $F_R = F$.*

b) The maps $F \rightarrow L_F$ and $L \rightarrow F_L$ defined above are mutually inverse and set up a 1-1 correspondence between closures and f-families on $\mathcal{U}$.

c) The maps $L \rightarrow S_L$ and $S \rightarrow L_S$ defined above are mutually inverse and set up a 1-1 correspondence between closures on $\mathcal{U}$ and semilattices of subsets of $\mathcal{U}$, containing $\{\mathcal{U}\}$. □

This theorem shows that we do not have to redefine concepts, once introduced for families of FDs or relations or closures or semilattices, if we need their interpretations for other objects - they can be easily obtained from the 1-1 correspondences of theorem -3.

In the sequel, by *relation scheme* we shall mean a pair $\langle \mathcal{U}, F \rangle$. All concepts defined for a relation scheme are automatically defined for any relation R by taking the relation scheme $\langle \mathcal{U}, F_R \rangle$.

Given a relation scheme $\langle \mathcal{U}, F \rangle$, a set $K \subseteq \mathcal{U}$ is called a *key* if $K \rightarrow \mathcal{U} \in F^+$ (equivalently, $L_F(K) = \mathcal{U}$). A key is called *minimal* (sometimes *candidate*) if it contains no key as a proper subset. All minimal keys form an *antichain* (i.e. $K_1 \not\subseteq K_2$ for any two distinct minimal keys K_1, K_2) and vice versa: any antichain in $\mathcal{P}(\mathcal{U})$ can be represented as a family of minimal keys of a relation scheme or a relation over $\mathcal{U}$.

Given a relation or a relation scheme, an attribute A is called *prime* if it belongs to a minimal key, and *nonprime* otherwise. The sets of prime and nonprime attributes will be denoted by $\mathcal{U}_p$ and $\mathcal{U}_n$ (or $\mathcal{U}_p(F), \mathcal{U}_p(L), \mathcal{U}_p(R)$ etc. if R or F or L is not clear from the context).

Given a relation scheme $\langle \mathcal{U}, F \rangle$, it is said to be in

- *Second Normal Form* (or 2NF for short) if for any minimal key K and a nonprime attribute A, $K' \rightarrow A \in F^+$ for no $K' \subset K$;

- *Third Normal Form* (or 3NF for short) if for any nonprime attribute A and X not containing A, X is a key whenever $X \rightarrow A \in F^+$;
- *Boyce-Codd Normal Form* (or BCNF for short) if X is a key whenever $X \rightarrow A \in F^+$ for $A \notin X$.

All the definitions given above are fairly standard. Now we introduce some terminology that appeared relatively recently in [4,7,8,9].

An *antikey* is a maximal non-key. In other words, let $\mathcal{K} = \{K_1, \ldots, K_r\}$ be a family of minimal keys of a relation or a relation scheme. Then X is an antikey if $K_i \subseteq X$ for no i and X is maximal such. The set of antikeys will be denoted by $\mathcal{K}^{-1}$.

Given a semilattice $S, M(S)$ stands for the set of (meet)-irreducible elements, i.e. such $X \in S$ that $X = Y \cap Z, Y, Z \in S$ implies either $Y = X$ or $Z = X$. Every element of a finite semilattice is an intersection of irreducibles. Maximal elements of $S - \{\mathcal{U}\}$ are called *coatoms*. The set of coatoms is denoted by $CA(S)$.

Theorem -2 [8,9] $\bigcup \mathcal{K} = \mathcal{U} - \bigcap \mathcal{K}^{-1}$. □

Theorem -1 [8,9] *Given a closure L, the set of antikeys it generates is $CA(S_L)$.* □

Given a relation $R = \{h_1, \ldots, h_m\}$, let $E_{ij} = \{A \in \mathcal{U} : h_i(A) = h_j(A)\}$, and $E_R = \{E_{ij} : 1 \leq i < j \leq m\} \bigcup \{\mathcal{U}\}$. E_R is called the *equality* set of R. It turns out that E_R contains all information about dependencies in R, i.e. L_R can be obtained from E_R by $L_R(X) = \bigcap \{Y : Y \in E_R, X \subseteq Y\}$. $CA(S_R)$ contains exactly the maximal sets from $E_R - \{\mathcal{U}\}$ [8,9].

Let $Cl_{\mathcal{U}}$ be the set of all closures on $\mathcal{U}$. Without loss of generality we shall also denote it by Cl_n if $|\mathcal{U}| = n$. Define $\geq$ on Cl_n by letting $L_1 \geq L_2$ iff $L_1(X) \subseteq L_2(X)$ for all X (in other words, $L_1 \cdot L_2 = L_2$).

Theorem 0 [7] *Cl_n is a lattice in which infimum ($\wedge$) and supremum ($\vee$) are defined as follows: $L = L_1 \wedge L_2$ iff $S_L = S_{L_1} \bigcap S_{L_2}, L = L_1 \vee L_2$ iff $S_L = S_{L_1} \bigcup S_{L_2} \bigcup \{X \cap Y : X \in S_{L_1}, Y \in S_{L_2}\}$.* □

In fact, $\wedge$ and $\vee$ can be expressed directly, but for our purposes this semilattice definition suffices.

A subset of Cl_n closed w.r.t. $\wedge$ ($\vee$ or both $\wedge$ and $\vee$) is called a *meet-subsemilattice (join-subsemilattice* and *sublattice*) respectively.

Given $X \subseteq \mathcal{U}$, let $Cl_n(X) = \{L \in Cl_n : \mathcal{U}_p(L) = X\} \bigcup \{L^1\}$, where L^1 is the top element of Cl_n, i.e. $L^1(Y) = Y$ for any Y.

The last definition to be given in this section is that of *interval:* if $X \subseteq Y \subseteq \mathcal{U}$, then $[X, Y]$ is the family of $Z \subseteq \mathcal{U}$ such that $X \subseteq Z \subseteq Y$.

2.1 Second Normal Form

In this section we give a semilattice characterisation of the second normal form (2NF). The set $2NF_n \subseteq Cl_n$ of the closures generated by relation schemes in 2NF will be shown to be neither meet- nor join- subsemilattice of Cl_n. An approximation of a closure defined as the one generated by a 2NF relation scheme and having the same set of prime attributes will be shown to exist.

Let L be a closure. A closed set, which can be obtained as the closure of a proper subset of a minimal key, will be called *prime*. In other words, a closed set X is prime if $X = L(Y)$ where $Y \subset K$ and K is a minimal key.

Theorem 1 *Let R be a relation over $\mathcal{U}$. Then R is in 2NF iff $[X \cap \mathcal{U}_p, X] \subseteq S_R$ for any prime $X \subseteq \mathcal{U}$.*

Proof. Suppose $[X \cap \mathcal{U}_p, X] \subseteq S_R$ for all prime $X \subseteq \mathcal{U}$. Assume R is not in 2NF, i.e. for some $A \in \mathcal{U}_n$, a minimal key K and $K' \subset K$ one has $K' \to A \in F_R$ and $A \notin K'$. Let $X = L_R(K')$. Clearly, X is prime, $X \neq \mathcal{U}$ and $A \in X$. Since X is closed, $X - A \to A \in F_R$, and $X - A \notin S_R$ (if $X - A \in S_R, X - A$ is a closed set containing K' which is a subset of X). On the other hand, $A \notin \mathcal{U}_p$, and $X \cap \mathcal{U}_p \subseteq X - A \subseteq X$, i.e. $X - A \in [X \cap \mathcal{U}_p, X] \subseteq S_R$, a contradiction. Thus, R is in 2NF.

Conversely, let R be in 2NF. Take a prime X where $X = L(Y), Y \subset K, K$ a minimal key. Let $A \in \mathcal{U}_n \cap X$. Then $A \notin Y$. If $X - A \to A \in F_R$, then $Y \to A \in F_R$, which contradicts our assumption that R is in 2NF. Hence $X - A \to A \notin F_R$, and since X is closed, so is $X - A$. Since S_R is a semilattice, $[X - \mathcal{U}_n, X] = [X \cap \mathcal{U}_p, X] \subseteq S_R$. The theorem is proved.

Define $2NF_n \subseteq Cl_n$ to be the subset of Cl_n consisting of all closures induced by relation schemes in 2NF. This set does not have any particular structure as a subset of Cl_n, i.e. it is neither meet- nor join- subsemilattice, as the following examples show. Let $\mathcal{U} = \{A_1, \ldots, A_{11}\}$, and consider three semilattices: $S = \{\emptyset, A_1, A_2, A_3, \{A_1, A_2, A_4\}, \{A_1, A_3, A_5\}, \mathcal{U}\}$, $S_1 = S \bigcup \{\{A_1, A_2, A_4, A_6\}, \{A_1, A_3, A_5, A_7\}, A_{10}\}$, $S_2 = S \bigcup \{\{A_1, A_2, A_4, A_8\}, \{A_1, A_3, A_5, A_9\}, A_{11}\}$. Then both L_{S_1} and L_{S_2} are in $2NF_{11}$, but $L_S = L_{S_1} \wedge L_{S_2}$ is not. A counterexample in the case of $\vee$ is even simpler. Let $\mathcal{U} = \{A_1, \ldots, A_4\}$, and again, consider three semilattices: $S_1 = [0, \{A_2, A_3\}] \bigcup \{A_4, \{A_1, A_4\}, \mathcal{U}\}$, $S_2 = [0, \{A_2, A_3, A_4\}] \bigcup \{\mathcal{U}\}$, and $S = S_1 \bigcup S_2$. Then both L_{S_1} and L_{S_2} are in $2NF_4$, but $L_S = L_{S_1} \vee L_{S_2}$, is not.

The approximation problem was studied for so-called *choice functions* [16] (i.e. functions on sets satisfying $C(X) \subseteq X$), and it was shown that being closed under intersection/union is necessary and sufficient for the existence of a unique upper/lower approximation. This result can be easily generalized for the functions that, being ordered, form a distributive lattice (notice that choice functions ordered by $\subseteq$ form a Boolean lattice). Unfortunately, the lattice Cl_n is not close to distributive (its properties are studied in [7] and [15]), and a counterexample can be found that shows nonexistence of the unique approximation for the 2NF.

However, we can try to approximate an arbitrary scheme by that in 2NF with the same set of prime attributes. In other words, we say that a closure L' is a *2NF-approximation* of a closure L if $L' \geq L$ and $L' \in Cl_n(\mathcal{U}_p(L)) \cap 2NF_n$. (Notice that we speak of *a* 2NF-approximation).

Let us give a procedure that finds a 2NF-approximation of a given closure $L \in Cl_n$.

1. For all prime X in S_L add $[X \cap \mathcal{U}_p(L), X]$ to S_L. Denote the extended S by $\hat{S}$.
2. Extend $\hat{S}$ to a semilattice. Denote this semilattice by S'. (In other words, S' is the minimal semilattice containing $\hat{S}$.
3. Let $2NF(L) = L_{S'}$.

Proposition 1 *Given a closure $L, 2NF(L)$ is a 2NF-approximation of L.*

Proof. Since $S \subseteq S', 2NF(L) \geq L$ in Cl_n. Moreover, since no new coatom appeared in S', by theorems -2 and -1 $\mathcal{U}_p(L) = \mathcal{U}_p(2NF(L))$, i.e. $2NF(L) \in Cl_n(\mathcal{U}_p(L))$. To prove that $2NF(L) \in 2NF_n$, consider an arbitrary prime X in S'. Since the families of antikeys of S and S' coincide and the antikeys unambiguously determine the keys, the keys of L and $2NF(L)$ are the same. Let $X = 2NF(L)(Y)$, where $Y \subset K$ and K is a minimal key. Then $X' = L(Y)$ is prime in S. Moreover, $X \subseteq X'$ since $L \leq 2NF(L)$. Since $X \in S', X$ is the intersection of all sets in S' that include X, X' among them. Let $X = X' \cap X_1 \cap \ldots \cap X_k$. Assume $A \in \mathcal{U}_n \cap X$. Thenm $A \in \mathcal{U}_n \cap X'$, and $X' - A \in S'$ as a set lying in $[X' \cap \mathcal{U}_p, X']$, which was added to S to get $\hat{S}$, and therefore lies in S'. Thus $X - A = (X' - A) \cap X_1 \cap \ldots \cap X_k \in S'$, which proves $[X - \mathcal{U}_n, X] = [X \cap \mathcal{U}_p, X] \subseteq S'$. Now, according to theorem 1, $2NF(L) \in 2NF_n$. □

2.2 Third Normal Form

In this section a lattice-theoretic characterization of the third normal form is given. Based on this characterization, the polynomiality of the 3NFTEST problem is proved for relations.[2] The approximation problem is solved in the case of 3NF for relations and relations schemes.

Theorem 2 *Let R be a relation over $\mathcal{U}$. Then R is in 3NF iff $[X \cap \mathcal{U}_p, X] \subseteq S_R$ for any $X \in S_R - \{\mathcal{U}\}$.*

Proof. Let R be in 3NF and $X \in S_R, X \neq \mathcal{U}$. Suppose $A \in \mathcal{U}_n$. As in the 2NF case, it is enough to show that $X - A \in S_R$. Assume $X - A$ is not closed; since X is, $L_R(X - A) = X$. Therefore $X - A \rightarrow A \in F_R$, and $X \rightarrow \mathcal{U} \in F_R$ for R is in 3NF. Closedness of X now implies $X = \mathcal{U}$, a contradiciton.

To prove the other direction, suppose $[X \cap \mathcal{U}_p, X] \subseteq S_R$ for any $X \in S_R - \{\mathcal{U}\}$; We must show that R is in 3NF. Let $X \rightarrow A \in F_R, A \in \mathcal{U}_n, A \notin X$. Let us prove that $X \rightarrow \mathcal{U} \in F_R$, or $L_R(X) = \mathcal{U}$. Suppose $L_R(X) = Y \neq \mathcal{U}$. Since $A \notin X, Y - A \in [X, Y] - \{Y\}$, and thus is not closed. But $Y - A \in [Y \cap \mathcal{U}_p, Y] \subseteq S_R$ and is therefore closed. This contradiction proves $L_R(Y) = \mathcal{U}$ and finishes the proof of the theorem. □

We denote the subset of Cl_n generated by 3NF relation schemes by $3NF_n$. Similarly to the 2NF case, this subset is not closed under the operations of Cl_n. One only has to observe that the closures L_{S_1}, L_{S_2} constructed in the previous section (for both $\vee-$ and $\wedge-$ cases) belong to $3NF_n$ while L_S does not since it is not in $2NF_n \subset 3NF_n$.

It is well-known that recognizing 3NF is $\mathcal{NP}$-complete in the case of relation schemes [14]. The situation is much better in the case of relations, where the problem has polynomial time complexity, as the following theorem shows.

Theorem 3 *There is an algorithm that, given a relation R over $\mathcal{U}$, decides whether R is in $3NF$ in a polynomial time in the number of rows and columns of R.*

Proof. Let us present an algorithm which, when given R as its input, produces a boolean variable x as the output:

1. Find the set $\mathcal{U}_p = \mathcal{U}_p(R)$.
2. Find the equality set E_R.

[2] The problem is known to be $\mathcal{NP}$-complete for relation schemes [14].

3. $x := 1$.

4. For all $X \in E_R$ and $A \in \mathcal{U}_n = \mathcal{U} - \mathcal{U}_p$ such that $A \notin X$ and $X \neq \mathcal{U}$ do the following: find the closure $L_R(X - A)$; if it equals $X - A$, go to the next pair (X, A), otherwise $x := 0$ and go to step 5.

5. Stop.

We claim that this algorithm works in polynomial time and that the output $x = 1$ iff R is in 3NF. Letus prove the first claim. Finding $\mathcal{U}_P$ can be done in polynomial time as shown in [8]. Constructing E_R is evidently polynomial in the size of R, and so is its own size. Finally, closure of any set can be found in $O(|E_R|)$ time, see [9]. Thus, the algorithm works in polynomial time.

Now, assume x produced by the algorithm is 0. Then for some $X \in E_R \subseteq S_R$ and $A \in \mathcal{U}_n$ we have $X - A \notin S_R$. Then R is not in 3NF by theorem 3. Let x be 1. Let $X \in S_R, X \neq \mathcal{U}$. Since $M(S_R) \subseteq E_R$ [9], $X = X_1 \cap \ldots \cap X_k$, where $X_1, \ldots, X_k$ are the elements of E_R which are supersets of X. Let $A \in X \cap \mathcal{U}_n$. Since $x = 1$, $X_i - A \in S_R$ for each i. Hence $X - A = (X_1 - A) \cap \ldots \cap (X_k - A) \in S_R$ and $[X \cap \mathcal{U}_p, X] \subseteq S_R$. Thus, R is in 3NF by theorem 3. □

Although $3NF_n$ is not closed under the operations of Cl_n, we nevertheless are able to find *the* 3NF approximation which is defined as follows:

Definition Let $L \in Cl_n$. Then $L' \in Cl_n$ is called the 3NF-*approximation* of L if the following holds:

1. $L' \geq L$;

2. $L' \in Cl_n(\mathcal{U}_p(L)) \cap 3NF_n$ ($i.e. \mathcal{U}_p(L) = \mathcal{U}_p(L')$ and L' is in $3NF$);

3. L' is the minimal such, i.e. if $L'' \in Cl_n(\mathcal{U}_p(L)) \cap 3NF_n$ and $L'' \geq L$, then $L'' \geq L'$.

Given a closure L, construct a closure denoted by $3NF(L)$ using the following procedure:

1. Add all intervals $[X \cap \mathcal{U}_p, X]$ for $X \in S_L$ to S_L. Denote S_L thus extended by $\hat{S}$.

2. Extend $\hat{S}$ to the semilattice, i.e. let S' be the minimal semilattice containing $\hat{S}$.

3. $3NF(L) = L_{S'}$.

Proposition 2 *Given a closure L, $3NF(L)$ is its 3NF-approximation.*

Proof. Since $S_L \subseteq S', 3NF(L) \geq L$. According to the procedure given above, no new coatom may appear in S' and since S' is an extension of S_L, $CA(S_L) = CA(S')$. Therefore $\mathcal{U}_p(L) = \mathcal{U}_p(3NF(L))$ by theorems -2 and -1.

To prove $3NF(L) \in 3NF_n$, consider $X \in S', X \neq \mathcal{U}$, and a nonprime $A \in X$. X can be represented as $X = X_1 \cap \ldots \cap X_k$, where $X_i \in [X_i^0 \cap \mathcal{U}_p, X_i^0]$ and $X_i^0 \in S_L$. Therefore, $X_i - A \in [X_i^0 \cap \mathcal{U}_p, X_i^0] \subseteq S'$ and $X - A = (X_1 - A) \cap \ldots (X_k - A) \in S'$. thus $3NF(L) \in 3NF_n$ by theorem 3.

If L'' is as in 3 of the definition of the 3NF-approximation, $S_L \subseteq S_{L''}$ and by theorem 3 $[X \cap \mathcal{U}_p, X] \subseteq S_{L''}$ for any $X \in S_L, X \neq \mathcal{U}$. Since $S_{L''}$ is a semilattice, this shows $S_{L'} \subseteq S_{L''}$ and $L'' \geq 3NF(L)$. The proposition is proved. □

Having described the 3NF-approximation, we have a natural question: how hard is it to find the approximation. Notice that the question asked is ambiguous - we did not specify what is given as an input: a relation or a relation scheme. That is, we have *two* different problems:

(3NF-APPROXIMATION FOR SCHEMES): *Given a relation scheme $\langle \mathcal{U}, F\rangle$, find a scheme $\langle \mathcal{U}, F'\rangle$ which is a 3NF-approximation of $\langle \mathcal{U}, F\rangle$ (to put it another way, $L_{F'} = 3NF(L_F)$).*

(3NF-APPROXIMATION FOR RELATIONS): *Given a relation R, find a relation R' which is R's 3NF-approximation (i.e. $L_{R'} = 3NF(L_R)$).*

The complexity result for these problems is very similar to the one for 3NFTEST - the problem is polynomial for relations and superpolynomial for schemes.

Theorem 4 *The problem 3NF-APPROXIMATION FOR RELATION can be solved in a polynomial time. The problem 3NF-APPROXIMATION FOR SCHEMES is superpolynomial provided that $\mathcal{P} \neq \mathcal{NP}$.*

Proof. Let R be a relation. Let $E'_R = E_R \bigcup \{X - A : X \in E_R, A \in \mathcal{U}_n(R)\}$. Since constructing both E_R and $\mathcal{U}_n(R)$ takes polynomial time in the size of R [8,9], E'_R can be found in polynomial time too. From theorem 3 we conclude that $M(S_L) \subseteq E'_R \subseteq S_L$, where $L = 3NF(L_R)$, and a relation R' satisfying $L_{R'} = L$ can be found by using the polynomial algorithms from [19]. This relation R' is a sought 3NF-approximation of R.

To prove the second part, show how we can use 3NF- APPROXIMATION FOR SCHEMES to solve 3NFTEST. Given a scheme $\langle \mathcal{U}, F\rangle$, let $\langle \mathcal{U}, F'\rangle$ be its 3NF-approximation. Notice that $\langle \mathcal{U}, F\rangle$ is in 3NF iff $F^+ = (F')^+$. Since checking the equality $F_1^+ = F_2^+$ for two arbitrary families of FDs takes polynomial time [18], knowing F' gives rise to a polynomial algorithm that tests 3NF. Since 3NFTEST is $\mathcal{NP}$-complete, $\mathcal{P} \neq \mathcal{NP}$ implies that approximation can not be found in a polynomial time. Note that an exponential time complexity algorithm was provided before proposition 2. The theorem is proved. □

2.3 Boyce-Codd Normal Form

In this section we discuss our main topics - characterization, testing, approximation - for BCNF. The characterization is the simplest one and corresponds to a well-known mathematical object: the order ideals. $BCNF_n$ turns out to be a sublattice of Cl_n, moreover, a distributive one. This ensures the existence of approximation, which, as in the 3NF case, can be found in polynomial time for relations and superpolynomial time for schemes.

Theorem 5 *Let R be a relation over $\mathcal{U}$. Then R is in BCNF iff $[\emptyset, X] \subseteq S_R$ for any $X \in S_R, X \neq \mathcal{U}$.*

Proof. Let R be in BCNF. Suppose $X - A \notin S_R$ for $X \in S_R - \{\mathcal{U}\}, A \in X$. Then $X - A \rightarrow A \in F_R$, implying $X \rightarrow \mathcal{U} \in F_R$. Thus $X - A \in S_R$ and $[\emptyset, X] \subseteq S_R$. Conversely, if the condition of the theorem holds, suppose $X \rightarrow A \in F_R, A \notin X$. If $X \rightarrow \mathcal{U} \notin F_R$, then $L_R(X) \neq \mathcal{U}$ and $X \in [\emptyset, L_R(X)] \subseteq S_R$, i.e. X is closed. This contradiction shows $L_R(X) = \mathcal{U}$, i.e. R is in BCNF. □

Some similar results for BCNF were established earlier, e.g. in [20]. The following corollary gives some alternative characterizations, all of them immediately derivable from theorem 5.

Corollary 1 *Given a relation scheme $\langle \mathcal{U}, F\rangle$, the following are equivalent:*

1. *$\langle U, F\rangle$ is in BCNF;*

2. *$[\emptyset, X] \subseteq S_F$ for every $X \in S_F - \{U\}$;*

3. *$S_F = (\bigcup_{i=1}^{t}[\emptyset, X_i]) \bigcup\{U\}$, where $X_1, \ldots, X_t$ are the antikeys;*

4. *$P(U) - S_F = (\bigcup_{i=1}^{r}[K_i, U]) - \{U\}$, where $K_1, \ldots, K_r$ are the minimal keys.*[3]

Let $BCNF_n$ stand for the subset of Cl_n generated y schemes in BCNF. Clearly, $BCNF_n \subseteq 3NF_n \subseteq 2NF_n$.

Proposition 3 *$BCNF_n$ is a distributive sublattice of Cl_n.*

Proof. Let $L_1, L_2 \in BCNF_n$. Since $S_{L_1} \cap S_{L_2}$ evidently satisfies the condition of theorem 5, $L_1 \wedge L_2 \in BCNF_n$. To prove $L_1 \vee L_2 \in BCNF_n$, represent S_{L_1} and S_{L_2} as $\bigcup_{i=1}^{r}[\emptyset, X_i] \bigcup\{U\}$ and $\bigcup_{i=1}^{t}[\emptyset, Y_i] \bigcup\{U\}$ respectively, see corollary 1. Let $Z_1, \ldots, Z_p$ be the maximal sets among $X_1, \ldots, X_r, Y_1, \ldots, Y_t$. Then $S_{L_1} \cup S_{L_2} \cup \{X \cap Y : X \in S_{L_1}, Y \in S_{L_2}\} = \bigcup_{i=1}^{p}[\emptyset, Z_i] \bigcup\{U\} = S_{L_1} \cup S_{L_2}$. Thus $L_1 \vee L_2 \in BCNF_n$. The sublattice $BCNF_n$ is distributive because the join and meet operations correspond to union and intersection of semilattices. □

BCNFTEST is known to be $\mathcal{NP}$-complete for relation schemes. BCNFTEST here is the problem that tests whether a subscheme of a relation scheme $\langle U, F\rangle$ generated by a proper subset $X \subset U$ is in BCNF. (Notice that checking whether the scheme itself is in BCNF takes polynomial time: one has to construct the *canonical* minimal cover [25] and check if it consists only of key dependencies). As in the 3NF case, the analogue of BCNFTEST problem for relations can be solved in polynomial time.

Proposition 4 *Given a relation R over U, BCNFTEST can be solved in polynomial time in the size of R.*

Proof. Let R be a relation and $X \subseteq U$. Let R_X denote the projection of R onto X. Denote the set of maximal elements of $E_{R_X} - \{X\}$ by E_X. Then, according to theorem 5, R_X is in BCNF iff for all $Y \in E_X$ and all $A \in Y : Y - A$ is closed, i.e. $L_{R_X}(Y - A) = Y - A$. Since constructing E_X takes polynomial time and closure can be computed in polynomial time too, the whole algorithm has polynomial time complexity. □

Similarly to th 3NF case, there exists unique approximation of a given closure by the one in BCNF. More precisely, we define *the BCNF-approximation* of a given closure L as the minimal closure L' such that $L' \geq L$ and $L' \in Cl_n(U_p(L)) \cap BCNF_n$. Let $BCNF(L)$ be the closure whose semilattice of closed sets is $\bigcup_{i=1}^{t}[\emptyset, X_i] \bigcup\{U\}$, where $X_1, \ldots, X_t$ are the antikeys of L. It follows immediately from corollary 1 and the definition of approximation:

Proposition 5 *Given a closure L, $BCNF(L)$ is its BCNF-approximation.* □

BCNF-approximation has clear interpretation in terms of FDs. If $L = L_F$ for a relation scheme $\langle U, F\rangle$, let $F' = \{K_1 \to U, \ldots, K_r \to U\}$, where $K_1, \ldots, K_r$ are the keys of $\langle U, F\rangle$. Then $L_{F'} = BCNF(L)$.

We finish this section by proving the complexity result for the approximation problem. As in the 3NF case, we have two problems:

[3] This result was proved by J. Biskup [3].

(BCNF-APPROXIMATION FOR SCHEMES): *Given a relation scheme* $\langle \mathcal{U}, F\rangle$, *find its BCNF-approximation, i.e. a relation scheme* $\langle \mathcal{U}, F'\rangle$ *such that* $L_{F'} = BCNF(L_F)$.

(BCNF-APPROXIMATION FOR RELATIONS): *Given a relation* R, *construct a relation* R' *which is* R*'s BCNF-approximation, i.e.* $L_{R'} = BCNF(L_R)$.

Theorem 6 *The problem BCNF-APPROXIMATION FOR RELATIONS can be solved in a polynomial time. The problem BCNF-APPROXIMATION FOR SCHEMES has exponential complexity.*

Proof. Let R be a relation and $X_1, \ldots, X_t$ its antikeys. Let $S = \bigcup_{i=1}^{t}[\emptyset, X_i] \bigcup \{\mathcal{U}\}$. Then $L_S \in BCNF_n$ by corollary 1 and $L_S = BCNF(L_R)$ because the family of antikeys unambiguously determines the family of keys. Let $\mathcal{M}_R = \{X_i - A : A \in \mathcal{U}, i = 1, \ldots, t\}$. Then $M(S) \subseteq \mathcal{M}_R \subseteq S$, and applying the polynomial algorithm of [19] we can construct a relation R' with $S_{R'} = S$. Since finding the antikeys takes polynomial time [8,9], R' can be constructed in a polynomial time if R is the input. Notice that $L_{R'} = L_S = BCNF(L_R)$. Thus R' is R's BCNF-approximation.

As it was mentioned before, an exponential complexity algorithm for BCNF-APPROXIMATION FOR SCHEMES exists: one has to find all minimal keys. On the other hand, it is clear that the size of the approximation F' is about the size of its canonical minimal cover [25] which consists of FDs $K_1 \rightarrow \mathcal{U}, \ldots, K_r \rightarrow \mathcal{U}$, where $K_1, \ldots, K_r$ are the minimal keys of $\langle \mathcal{U}, F\rangle$ and it can be exponential: Yu and Johnson [26] have given an example of a scheme consisting of k functional dependencies on k^2 attributes with $k!$ minimal keys. For a detailed discussion of schemes reaching this extremal number of minimal keys, see [2]. □

Note that the polynomial algorithm for approximation problem for relations, described in the proof of theorem 6, was used in [13] to construct an algorithm that, given a relation, finds its minimal keys. This problem has exponenetial time complexity, but it can be decomposed into two subproblems: BCNF-APPROXIMATION, which has polynomial time complexity, and *dependency inference problem* [21] for which good practical algorithms exist.

References

[1] C. Beeri, P.A. Bernstein, Computational problems related to the design of normal form relation schemes, *ACM TODS* **4** (1979), 30-59.

[2] A. Békéssy, J. Demetrovics, Contribution to the theory of data base relations, *Discrete Matheematics* **27** (1979), 1-10.

[3] J. Biskup, private letter.

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

[5] E.F. Codd, Further normalization of the data base relational model, in: *Data Base Systems*, Prentice Hall (1972), 33-64.

[6] A. Day, A lattice interpretation of database dependencies, Preprint, Lakehead University, 1989.

[7] J. Demetrovics, L. Libkin, I. Muchnik, Functional dependencies and the semilattice of closed classes, in: *Proc. of the 2nd Symp. on Math. Fund. of Database Syst.,* (J. Demetrovics, B. Thalheim eds.) *Springer LNCS* **364** (1989), 136-147.

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

[9] J. Demetrovics, V.D. Thi, Relations and minimal keys, *Acta Cybernetica* **8** (1988), 279-285.

[10] R. Fagin, Horn clauses and database dependencies, *Journal of ACM* **29** (1982), 678-698.

[11] R. Fagin, M. Vardi, The theory of data dependencies - a survey, in: *Mathematics of Information Processing* (M. Anshel, W. Gewirtz eds.), Amer. Math. Soc. **34** (1986), 19-71.

[12] M. Gary, D. Johnson, *Computers and Intractability: A Guide to NP-completness* (W.H. Freeman and Co., San Francisco, 1979).

[13] G. Gottlob, L. Libkin, Investigations on Armstrong relations, dependency inference, and excluded functional dependencies, *Acta Cybernetica* **9** (1990), 385-402.

[14] J.H. Jou, P.C. Fischer, The complexity of recognizing 3NF relation schemes, *Inform. Process. Letters* **14** (1982), 187-190.

[15] L. Libkin, I. Muchnik, The lattice of subsemilattices of a semilattice, *Algebra Universalis,* to appear.

[16] B.M. Litvakov, Approximation of choice functions, *Automation and Remote Control* **45** (1984), 1221-1229.

[17] C.L. Lucchesi, S.L. Osborn, Candidate keys for relations, *JCSS* **17** (1978), 210-279.

[18] D. Maier, *The Theory of Relational Databases* (Computer Science Press, Rockville, MD, 1983).

[19] H. Manilla, K.-J. Räihä, Design by example: an application of Armstrong relations, *JCSS* **33** (1986), 126-141.

[20] H. Manilla, K.-J. R äihä, Practical algorithms for finding prime attributes and testing normal forms, in: *Proc. of the Symp. on Principles of Database Systems* (1989), 128-133.

[21] H. Manilla, K.-J. Dependency inference, in: *Proc. of the Conf. on Very Large Databases* (1987), 155-158.

[22] J. Paredaens, P. De Bra, M. Gyssens and D. Van Gucht, *The Structure of the Relational Datamodel* (Springer-Verlag, Berlin, 1989).

[23] B. Thalheim, *Dependencies in Relational Databases* (Stuttgart-Leipzig, 1991).

[24] J.D. Ullman, *Principles of Database Systems* (Pittman, 2nd ed., 1982).

[25] M. Wild, Implicational bases for finite closure systems, Preprint 1210, University of Darmstadt, 1989.

[26] C.T. Yu, D.T. Johnson, On the complexity of finding the set of candidate keys for a given set of functional dependencies, *Information Pocessing Lett.* **5** (1976), 100-101.

Received April 5, 1991.
Revised version received November 5, 1991.