Combinatorial and Algebraic Results for Database Relations

G.O.H. Katona

Mathematical Institute of the Hungarian Academy of Sciences
Budapest, POBox 127, 1364, Hungary

Abstract. A database R has some obvious and less obvious parameters like the number of attributes, the size $|R|$, the maximum size of a domain, the number of some special functional dependencies (e.g. the minimal keys), and so on. The main aim of the paper is to survey some of the results giving connections, inequalities among these parameters. Results of this type give tools to guess the structure of the database having little *a priori* information. The methods are of combinatorial nature.

1 Introduction

The simplest model of a database is a matrix. The entries in one column are the datas of the same kind (name, date of birth, *etc.*), the entries of one row are the datas of one individual. Thus, actually, we are dealing with finite sets of homogeneous finite functions which can be illustrated by matrices.
Let us introduce, however, the names of these concepts in the form as they used in the literature. One kind of datas (*e.g.* name) is called an *attribute*. It can be identified with a column of the above matrix. The set of attributes will be denoted by $U = \{a_1, \ldots, a_n\}$. The set of possible entries in the ith column is the *domain* of a_i. It is denoted by $D(a_i)$. Thus, the data of one individual (row of the matrix) can be viewed as an element r of the direct product $D(a_1) \times D(a_2) \times \ldots \times D(a_n)$. Such an element is called a *tuple*. Therefore the whole database (or matrix) can be described by the *relation* $R \subseteq D(a_1) \times D(a_2) \times \ldots \times D(a_n)$, that is, by the set of tuples. If $r = (e_1, e_2, \ldots, e_n) \in R$ then $r(i)$ denotes the i component of r, that is, e_i. There might be some logical connections among the datas. For instance, the date of birth determines the age (in a given year). Let A and B be two sets of attributes $(A, B \subseteq U)$. The datas in A might uniquely determine the datas in B. Formally we say that $B \subseteq U$ *functionally depends* on $A \subseteq U$ if

$$r_1(i) = r_2(i) \quad \text{for all such} \quad i \quad \text{that} \quad a_i \in A$$

implies

$$r_1(i) = r_2(i) \quad \text{for all such} \quad i \quad \text{that} \quad a_i \in B.$$

* The work was supported by the Hungarian National Foundation for Scientific Research, grant number 2575.

It is denoted by $A \to B$ ([2],[12]). Less formally, $A \to B$ if any two elements of R having the same values in the attributes belonging to A must have the same values also in B. Functional dependencies have a very important role in practical applications, in most of the present paper we will consider them and some natural generalizations.

After this rough introduction of the concepts, let us be more precise. Two levels are distinguished in the database theory. The set U with the sets $D(a_i)$ and the set of logical connections (like functional dependencies) is called the *database scheme*. It determines the set R^* of possible tuples. The other level is the set R of actual tuples. This is called the *instance*. The instance obviously has to satisfy the conditions of the database scheme, that is, $R \subset R^*$. However, the inclusion is, in general, a proper one.

The traditional investigations of relational databases suppose that the database scheme is *a priori* given in some way. That is, e.g. some functional dependencies can be deduced either by the logic of the data or by the analysis of (one of) the instances. Then the determination of all functional dependencies is a question of computation. The scheme is fully determined. This information is used then to decompose and store the instances efficiently. Of course, this logical structure of the database scheme can be much more complex, several other (non-functional) dependencies might be known and used.

There is, however a basically different way of considering a database. Knowing some partial (sometimes very week) information on the instances, determine a scheme compatible with the given instances. Or, more modestly, determine some parameters of the scheme. The final goal is to obtain a simple scheme, since it is needed to decompose and store the instances reliably. The more complex the scheme is, the smaller storage space is needed. However it is more probable that the next instance will not be compatible. So actually we should find the simplest scheme determined by the informations on the instances where "simplest" can be defined in different ways. On the other hand, notice that the known information can be of very different nature from the description of the scheme. Since we want to use certain types of connections (which can be strongly used in the decomposition) like, for instance, functional dependencies, but the information could be entirely different from a dependency. Let us call this situation a *database relation with unknown structure*. (Neither the instance nor the scheme is known fully.)

Examples:

1) The number of attributes and the number of tuples are known. What can be said about the system of functional dependencies? In Section 3 some theorems of the following type are collected. Given the number of attributes and the system of functional dependencies, what is the minimum number of tuples? Obviously, if the given number of tuples is smaller than this minimum then the considered system of functional dependencies can be excluded.

2) The number of attributes and the number of tuples are known. Moreover, there is an assumption on the distribution on each domain. Having no more information it is natural to suppose that the tuples are chosen with random components following the given the distribution. What can be said about the system of functional dependencies? In Section 4 there are some modest results on the expected size of the functional dependencies.

The reality is (like almost always) between the two extremist models. Namely, it is impossible to put the data into the computer without a preliminary scheme. On the other hand, it migh turn out during the use of the database that there are counterexamples for our assumptions causing inconvenient chaos. Or we might observe new dependencies which escaped our attention before. (See, for instance, the "design by example" in Thalheim's book [41].) Thus the scheme might or should be improved in more steps untill it achieves its best performance. In Section 5 it will be supposed that the scheme contains only functional dependencies as logical constrains. This situation suggests the study of the system of possible schemes. Which systems of functional dependencies can be obtained by an elementary change from another given system of dependencies? How many such schemes are there at all? These and similar questions are treated in Section 5 using a partially oredered set whose elements are the relational schemes on the same U, modelled with their systems of fuctional dependencies.

The scheme might contain several other types of logical constrains, but we will consider, almost exclusively, only functional dependencies in the present paper. This fact justifies to study the functional dependencies in an introductory Section 2. Instead of the more common "system of functional dependencies" we use an equivalent form, the *closure operation*. This mathematical structure is easier. This fact made possible to characterize the *numerical dependencies* studied in [28]. (See Section 6.)

While the tools of the traditionals works in database theory were mathematical logic and similar mathematical disciplines, the works surveyed in the present paper needs some other mathematical areas, namely combinatorics (sometimes with a slight algebraic flavour, like in Sections 2 and 6) and probability theory.

I am indebted to J. Biskup and B. Thalheim for their help in writing this paper and to J. Demetrovics, my manifold co-author in the area. Actually, the present paper is based on the survey [20].

2 Different Characterizations of the Systems of Functional Dependencies

It is easy to see that the following four properties hold for the functional dependencies in any relation R. Let A, B, C and D be subsets of the set U of the attributes of R.

$$A \to A, \tag{2.1}$$

$$A \to B \quad \text{and} \quad B \to C \quad \text{imply} \quad A \to C, \tag{2.2}$$

$$A \subseteq C, \quad D \subseteq B \quad \text{and} \quad A \to B \quad \text{imply} \quad C \to D, \tag{2.3}$$

$$A \to B \quad \text{and} \quad C \to D \quad \text{imply} \quad A \cup C \to B \cup D. \tag{2.4}$$

A system $\mathcal{D}$ of pairs (A, B) of subsets of U satisfying (2.1)-(2.4) is called a *determination*. (To be precise, we should repeat here these conditions with some other kind of arrows, like $A \hookrightarrow B$ in place of $A \to B$ but we will use the same notation for functional dependencies in a relation and in a determination.)

The system of all functional dependencies in a relation R was called a *full family* by Armstrong [2]. He also found the characterization of the possible full families.

Theorem 2.1 [2] *A system of pairs $A \to B$ of sets is a full family for some relation R iff it is a determination.*

In order to illustrate the difference in use of the above and the forthcoming characterizations an example will be used. It is a theorem which otherwise fits better to Section 3. If $K \to U$, that is, the values in K determine all other values then K is called a *key*. If K is a key and contains no other key as a proper subset then it is a *minimal key*. The following theorem determines the maximum number of minimal keys in an instance of a database with n attributes.

Theorem 3.0 [13] *The number of minimal keys is at most*

$$\binom{n}{\lfloor \frac{n}{2} \rfloor}$$

and this estimate is sharp.

The inequality easily follows from the well known theorem of Sperner [38] which states that the size $|\mathcal{S}|$ of an *inclusion-free family* $\mathcal{S}$ ($A, B \in \mathcal{S}$ implies $A \not\subseteq B$) is at most $\binom{n}{\lfloor \frac{n}{2} \rfloor}$. To prove that the inequality cannot be improved one has to construct a relation R with n attributes and this many minimal keys. This can be done by using Theorem 2.1. Define the determination $\mathcal{D}$ to contain the pairs $A \to B$ for all $B \subseteq A \subseteq U$ and $A \to B$ for all $A, B \subseteq U$ such that $\lfloor \frac{n}{2} \rfloor \leq |A|$. It is easy (but somewhat tedious) to check that this is a determination, therefore there is a relation R in which the full family consists of these functional dependencies. The minimal keys in this relation are all the sets of size $\lfloor \frac{n}{2} \rfloor$. This proves the sharpness of the statement of Theorem 3.0.
Given a determination $\mathcal{D}$ on U one can define

$$\mathcal{L}(A) = \{a : \quad A \to a\} \quad \text{for all} \quad A \subseteq U. \tag{2.5}$$

The following properties can easily be proved for all $A, B \subseteq U$.

$$A \subseteq \mathcal{L}(A), \tag{2.6}$$

$$A \subseteq B \quad \text{implies} \quad \mathcal{L}(A) \subseteq \mathcal{L}(B), \tag{2.7}$$

$$\mathcal{L}(\mathcal{L}(A)) = \mathcal{L}(A). \tag{2.8}$$

A set-function satisfying these properties is called a *closure operation* or shortly a *closure*.

Proposition 2.2 *The correspondence $\mathcal{D} \to \mathcal{L}(\mathcal{D})$ defined by (2.5) gives a bijection between the set of determinations and the set of closures.*

Consider the following closure

$$\mathcal{L}^n_{\lfloor n/2 \rfloor}(A) = \begin{cases} A & \text{if } |A| < \lfloor \frac{n}{2} \rfloor, \\ U & \text{if } |A| \geq \lfloor \frac{n}{2} \rfloor, \end{cases} \quad A \subseteq U.$$

This closure defines the determination $\mathcal{D}^n_{\lfloor n/2 \rfloor}$ by Proposition 2.2. On the other hand, there is such a relation R that its full family is $\mathcal{D}^n_{\lfloor n/2 \rfloor}$. It is obvious that the minimal keys in this relation are the $\lfloor \frac{n}{2} \rfloor$-element subsets of U. This gives an easier proof of the sharpness of Theorem 3.0.
The reader could see that it was somewhat easier to prove Theorem 3.0 by using the closures. Moreover it is self-clear that a closure is an easier structure than a system of functional dependencies. To describe the closure, only some functional dependencies should be given, not all of them.
Given a closure $\mathcal{L}$ on U, define the *closed sets* by $B = \mathcal{L}(B)$. The family of closed sets is denoted by $\mathcal{Z} = \mathcal{Z}(\mathcal{L})$). It is easy to see that $\mathcal{Z}$ is closed under intersection, that is, $A, B \in \mathcal{Z}$ implies $A \cap B \in \mathcal{Z}$. Furthermore, $U \in \mathcal{Z}$. A family $\mathcal{Z}$ satisfying these properties is called an *intersection semi-lattice.*

Proposition 2.3 *The correspondence $\mathcal{L} \to \mathcal{Z}(\mathcal{L})$ is a bijection between the set of closures and the set of intersection semi-lattices.*

Denote by $\mathcal{D}^n_k$ the determination containing the pairs $A \to B$ for all $B \subseteq A \subseteq U$ and $A \to B$ for all $A, B \subseteq U$ such that $k \leq |A|$. The corresponding closure is

$$\mathcal{L}^n_k = \mathcal{L}(\mathcal{D}^n_k) = \begin{cases} A & \text{if } |A| < k, \\ U & \text{if } |A| \geq k, \end{cases} \quad A \subseteq U.$$

Note that $\mathcal{Z}^n_k = \mathcal{Z}(\mathcal{L}^n_k)$ consists of U and all sets of size at most $k-1$.
Given an intersection semi-lattice $\mathcal{Z}$ define

$$\mathcal{M} = \mathcal{M}(\mathcal{Z}) = \{M : \quad M \in \mathcal{Z} \quad \text{and there are no} \quad r \geq 2 \quad \text{sets in } \mathcal{Z}, \\ \text{all different from} \quad M \quad \text{such that their intersection is} \quad M\}. \tag{2.9}$$

It is easy to see that (i) no member M of $\mathcal{M}$ is an intersection of other members (all different from M) and (ii) $U \in \mathcal{M}$. Such families of subsets are called *intersection-free families.*

Proposition 2.4 *The correspondence $\mathcal{Z} \to \mathcal{M}(\mathcal{Z})$ is a bijection between the set of intersection semi-lattices and the set of intersection-free families.*

It is easy to see that $\mathcal{M}^n_k = \mathcal{M}(\mathcal{Z}^n_k)$ consists of U and the sets of size $k-1$.
Propositions 2.2, 2.3 and 2.4 give different equivalent notions describing the same thing in different ways. The given goal determines which one of them should be used. They more or less belong to the folklore but their proofs (and the inverse mappings) can be found in [8] and [18]. In the rest of this section we show some other equivalent notions which are/might be useful for some applications.
Let $\mathcal{Z}$ be an intersection semi-lattice on U and suppose that $H \subset U, \quad H \notin \mathcal{Z}$ hold and $\mathcal{Z} \cup \{H\}$ is also closed under intersection. Consider the sets A satisfying $A \in \mathcal{Z}, H \subset A$. The intersection of all of these sets is in $\mathcal{Z}$ therefore it is different from H. Denote it by $\mathcal{L}(H)$. (If $\mathcal{Z} = \mathcal{Z}(\mathcal{L})$ then $\mathcal{L}(H)$ is the closure of H according to $\mathcal{L}$.) $H \subset \mathcal{L}(H)$ is obvious. Let $\mathcal{H}(\mathcal{Z})$ denote the set of all pairs $(H, \mathcal{L}(H))$ where $H \subset U, H \notin \mathcal{Z}$ but $\mathcal{Z} \cup \{H\}$ is closed under intersection. The following theorem characterizes the posssible sets $\mathcal{H}(\mathcal{Z})$:

Theorem 2.5 [8] *The set* $\{(A_i, B_i)\}_{i=1}^m$ *is equal to* $\mathcal{H}(\mathcal{Z})$ *for some intersection semi-lattice* $\mathcal{Z}$ *iff the following conditions are satisfied:*

$$A_i \subset B_i \subseteq U, \quad A_i \neq B_i, \tag{2.10}$$

$$A_i \subseteq A_j \quad \text{implies either} \quad B_i \subseteq A_j \quad \text{or} \quad B_i \supseteq A_j, \tag{2.11}$$

$$A_i \subseteq B_j \quad \text{implies} \quad B_i \subseteq B_j, \tag{2.12}$$

$$\text{for any } i \text{ and } \quad C \subset U \quad \text{satisfying} \quad A_i \subset C \subset B_i (A_i \neq C \neq B_i)$$

$$\text{there is a } j \text{ such that either} \quad C = A_j \quad \text{or} \quad A_j \subset C, B_j \not\subset C, B_j \not\supset C \quad \text{all hold.} \tag{2.13}$$

The set of pairs (A_i, B_i) satisfying (2.10)-(2.13) is called an *extension*. Its definition is not really beautiful but it is needed in some applications (see Section 5). On the other hand it is also an equivalent notion to the closures:

Theorem 2.6 [8] $\mathcal{Z} \to \mathcal{H}(\mathcal{Z})$ *is a bijection between the set of intersection semi-lattices and the set of extensions.*

Let $\mathcal{L}$ be a closure on U. Define $\mathcal{S}_i$ as the family of minimal sets $A \subseteq U$ such that $a_i \in \mathcal{L}(A)$. It is clear that no member of $\mathcal{S}_i$ is a subset of another member of it. Such families are called *inclusion-free* or *Sperner families*. So it is obvious that

$$\mathcal{S}_i \quad (1 \leq i \leq n) \quad \text{is a Sperner family} \tag{2.14}$$

and

$$\text{either} \quad \mathcal{S}_i = \{\emptyset\} \quad \text{or} \quad \{a_i\} \in \mathcal{S}_i. \tag{2.15}$$

One more, essential property can be proved:

$$\begin{aligned} &\text{if} \quad A \subset U \quad \text{contains no subset belonging to} \quad \mathcal{S}_i \\ &\text{then} \quad \{j : A \quad \text{contains a subset belonging to} \quad \mathcal{S}_j\} \\ &\qquad \text{contains no subset belonging to} \quad \mathcal{S}_i. \end{aligned} \tag{2.16}$$

Proposition 2.7 [25] *The* $|U|$ *Sperner families satisfying (2.14)-(2.16) give an equivalent description of the closures.*

A function $\mathcal{C}$ satisfying

$$\mathcal{C}(A) \subseteq A \quad (A \subseteq U) \tag{2.17}$$

is a *choice function*. Given a closure $\mathcal{L}$,

$$\mathcal{C}(A) = U - \mathcal{L}(U - A) \tag{2.18}$$

is a choice function.

Theorem 2.8 [16] *The correspondence defined by (2.18) is a bijection between the set of closures and the set of choice functions satisfying*

$$\mathcal{C}(A) \subseteq B \subseteq A \quad \textit{implies} \quad \mathcal{C}(A) = \mathcal{C}(B) \quad \textit{for all} \quad A, B \subseteq U$$

and

$$A \subseteq B \quad \textit{implies} \quad \mathcal{C}(A) \subseteq \mathcal{C}(B) \quad \textit{for all} \quad A, B \subseteq U.$$

Given a determination $\mathcal{L}$ (or a closure $\mathcal{D}$, *etc.*) it determines the family $\mathcal{K} = \mathcal{K}(\mathcal{L})$ (or $\mathcal{K} = \mathcal{K}(\mathcal{D})$) of minimal keys. It is a non-empty Sperner family. Conversely, if a non-empty Sperner family $\mathcal{K}$ is given then

$$\mathcal{L}(A) = \begin{cases} A & \text{if there is no } K \in \mathcal{K} \text{ such that } K \subseteq A, \\ U & \text{if there is a } K \in \mathcal{K} \text{ such that } K \subseteq A \end{cases}$$

is a closure and the set of minimal keys in it is $\mathcal{K}$. This, Theorem 2.1 and Proposition 2.2 prove the following proposition.

Proposition 2.9 *For any non-empty Sperner family $\mathcal{K}$ there is a relation R in which the family of minimal keys is $\mathcal{K}$.*

Of course, $\mathcal{K}$ does not always determine $\mathcal{L}$ uniquely.

3 Inequalities for the Parameters of a Database

Let us go back to the first example in the introduction. A database (scheme or instance) has some obvious or less obvious parameters like the number n of attributes, the size $m = |R|$ of the relations, the maximum size of a domain, the number of some special functional dependencies (*e.g.* the number of minimal keys or the number of functional dependencies $A \to b$ where $|A| \leq k$ and b is an attribute), *etc.* If some theorems ensure the validity of certain inequalities are known among these parameters and we have information on the actual values of these parameter of the instance then some statement can be concluded for the other parameters of the scheme. So any inequalities of this kind may help in the prediction of the structure of the scheme, knowing a little about the instance.
We have shown an example of these kind of problems in Section 2. Theorem 3.0 determined the maximum number of minimal keys. Knowing the number of attributes we can upperestimate (somewhat less than 2^n) the number of minimum keys. Thalheim observed that this bound can be improved if the domains are bounded.

Theorem 3.1 [40] *Suppose that $D(a_i) \leq k \quad (1 \leq i \leq n)$ where $k^4 < 2n+1$. Then the number of minimal keys cannot exceed*

$$\binom{n}{\lfloor \frac{n}{2} \rfloor} - \lfloor \frac{n}{2} \rfloor.$$

Problem 3.2 *Improve this bound for small k-s.*

Another similar interesting question is the following. Theorem 3.0 determined the maximum number of minimum keys. How small can it be? There is one minimum key, always. It is obvous, that there are schemes with exactly one minimum key. [6] determined all the schemes described by functional dependencies having exactly one minimal key.

Let us show here an extension of Theorem 3.0. In many practical cases it is known that a certain set of attributes cannot uniquely determine a too large set of attributes. Formally, $|B - A| \leq k$ (suppose $k \leq n/2$) must hold for any functional dependency $A \rightarrow B$. It follows that the keys are of size at least $n - k$. As earlier, the minimal keys form a Sperner family. Thus, we have to find the largest Sperner family with members of size at least $n-k$. But this is an easy task knowing the YBLM-inequality (Yamamoto [42], Bollobás [7], Lubell [32], Meshalkin [34], it is often called LYM-inequality):

YBLM-inequality *If the number of i-element members in a Sperner family $\mathcal{S}$ of n elements is f_i then*

$$\sum_{i=0}^{n} \frac{f_i}{\binom{n}{i}} \leq 1. \tag{3.1}$$

In our case, if the Sperner family is the family of minimal keys then $f_0 = f_1 = \ldots = f_{n-k-1} = 0$ holds. Use the inequality

$$\binom{n}{i} \leq \binom{n}{n-k} \quad \text{if} \quad k \leq n/2, n-k \leq i \leq n$$

in (3.1):

$$1 \geq \sum_{i=n-k}^{n} \frac{f_i}{\binom{n}{i}} \geq \sum_{i=n-k}^{n} \frac{f_i}{\binom{n}{n-k}} =$$

$$\frac{\sum_{i=n-k}^{n} f_i}{\binom{n}{n-k}} = \frac{|\mathcal{S}|}{\binom{n}{n-k}}.$$

This proves the following statement.

Theorem 3.3 *Suppose that all functional dependencies $A \rightarrow B$ on an n-element set of attributes satisfy $|B - A| \leq k$ where $k \leq n/2$. Then the number of minimal keys is at most*

$$\binom{n}{k}.$$

Thalheim ([39] and [40]) obtained interesting results for the same problems for the case of null-values (some datas of some individuals are unknown).

The maximum number of functional dependencies is uninteresting, since the determination uniquely determined by the functional dependency $\emptyset \rightarrow U$ serves as the extremal one. (The number of functional dependencies is 2^{2^n} here.) The situation is rather different if we consider only those functional dependencies which are non-trivial and non-reducible. A functional dependency $A \rightarrow B$ is called *non-reducible* if

$$A \neq B,$$

$$\text{there is no} \quad A' \subset A (A' \neq A) \quad \text{such that} \quad A' \to B$$

$$\text{there is no} \quad B' \supset B (B' \neq B) \quad \text{such that} \quad A \to B'.$$

Let $N(n)$ denote the maximum number of non-reducible functional dependencies in a determination on n elements.

Theorem 3.4 ([3] and [31])

$$2^n \left(1 - \frac{4 \log_2 \log_2 n}{\log_2 e \log_2 n}\right)(1 + o(1)) \leq N(n) \leq 2^n \left(1 - \frac{\log_2^{3/2} n}{150\sqrt{n}}\right).$$

At first sight it might be surprising that this number is near to the obvious upper bound 2^n. But in this case the real question is to determine the deviation from this upper bound, that is, the second term. The above theorem gives only estimates.
A similar, but perhaps more natural parameter of a determination $\mathcal{D}$ is the following one. Let $\mathcal{E}$ be a set of functional dependencies on a set U, not necessarily satisfying the conditions (2.1)-(2.4). We say that $\mathcal{E}$ generates the determination $\mathcal{D}$ iff $\mathcal{E} \subseteq \mathcal{D}$ and $\mathcal{D}$ is the smallest such determination. The size $|\mathcal{E}|$ of the smallest $\mathcal{E}$ generating the determination $\mathcal{D}$ can be considered as the design complexity of $\mathcal{D}$. It is denoted by $C(\mathcal{D})$. Furthermore introduce the notation $C(n) = \max\{C(\mathcal{D}) : \quad \mathcal{D}$ is a determination on an n-element set $\}$ for the design complexity of the most complex determination in this sense. There is an obvious upper estimate by Theorem 3.4 and $C(n) \leq N(n)$. (It is not known how far these two parameters can be.) The lower estimate can be obtained proving that

$$C(\mathcal{D}^n_{\lfloor n/2 \rfloor}) = \binom{n}{\lfloor \frac{n}{2} \rfloor}.$$

(See Thalheim [41].)

Theorem 3.5

$$c\frac{2^n}{\sqrt{n}} \sim \binom{n}{\lfloor \frac{n}{2} \rfloor} \leq C(n) \leq 2^n \left(1 - \frac{\log_2^{3/2} n}{150\sqrt{n}}\right).$$

A relation R in which the full family is $\mathcal{D}^n_{\lfloor n/2 \rfloor}$) must have exponentially many rows (see Lemma 3.10), that is, $|R|$ must be very large. Mannila and Räihä [33] started to investigate the analogous question with bounded $|R|$. Let $C(n, m)$ denote $\max\{C(\mathcal{D}) : \quad \mathcal{D}$ is a relation R on an n-element set U with size $|R|$ at most $m\}$. The following result, surprisingly, states that the minimum number of functional dependencies generating the worst determination remains exponential even in the case of linearly many rows.

Theorem 3.6 [33]

$$C(2u + 1, 3u + 2) \geq 2^u.$$

Problem 3.7 *Find estimates for $C(n, m)$, in general.*

Mannila and Räihä also investigated the algorithms finding the smallest (that is, of size $|C(\mathcal{D})|$) $\mathcal{E}$ generating $\mathcal{D}$. They have shown in [33] that the number of steps is at least $cm \log m$ for fixed n, where m is the number of rows and c is a constant independent of m. The brute force algorithm needs $O(n^2 2^n m \log m)$ steps. On the other hand as a function of n, the number of steps must be exponential. In the proof they use the number of different $\mathcal{D}$s on an n-element set. (See Section 4.)

Problem 3.8 *Give estimates on $N_k(n)$ and $C_k(n)$ where these numbers are defined analogously to $N(n)$ and $C(n)$ under the restriction $|B - A| \leq k$ for all functional dependencies $A \to B$.*

It is known from the results surveyed in Section 2, that there is a relation R for any determination $\mathcal{D}$, closure $\mathcal{L}$ or set of minimal keys $\mathcal{K}$ such that R generates exactly this given $\mathcal{D}$, $\mathcal{L}$ or $\mathcal{K}$. It is not clear, however, what is the minimum of $|R|$ satisfying these goals. Let $s(\mathcal{D})$, $s(\mathcal{L})$ and $s(\mathcal{K})$ denote these minimums.

Theorem 3.9 [13], [15]

$$s(\mathcal{K}) \leq 1 + \binom{n}{\lfloor \frac{n}{2} \rfloor}.$$

holds for any non-empty Sperner family $\mathcal{K}$ on n elements. On the other hand, there is such a $\mathcal{K}$ satisfying

$$\frac{1}{n^2}\binom{n}{\lfloor \frac{n}{2} \rfloor} < s(\mathcal{K}).$$

The proof of the latter inequality is not constructive. We do not know the (nearly) worst Sperner families. A possible candidate is $\mathcal{D}^n_{\lfloor n/2 \rfloor}$. This is one of the motivations to study $s(\mathcal{K}^n_k)$ where $\mathcal{K}^n_k)$ denotes the family of all k-element sets of an n-element set. The following easy lemma is surprisingly strong.

Lemma 3.10 ([17] and [14])

$$\binom{s(\mathcal{K}^n_k)}{2} \geq \binom{n}{k-1} \quad (0 \leq k \leq n).$$

For $k = 1, 2$ or $n - 1$ the lower estimate obtained by this lemma is sharp. It can be shown by easy constructions. For $k = n$ this inequality is too weak, but the exact result can be obtained by a small trick.

Theorem 3.11 ([17] and [14])

$$s(\mathcal{K}^n_1) = 2, \quad s(\mathcal{K}^n_2) = \left\lceil \frac{1 + \sqrt{1 + 8n}}{2} \right\rceil,$$
$$s(\mathcal{K}^n_{n-1}) = n, \quad s(\mathcal{K}^n_n) = n + 1.$$

The case $k = 3$ is very interesting from the mathematical point of view. Lemma 3 leads to $s(\mathcal{K}^n_3) \geq n$. In [14] we proved the equality for ns of form $12r + 1$ and $12r + 4$ and conjectured that the equality holds for all $n \geq 7$. We also stated a conjecture for Steiner triple systems where n is of the form $3r + 1$. This conjecture would imply

the equality $s(\mathcal{K}_3^n) = n$. Andrea Rausche [35] found a counterexample for $n = 10$, but Ganter and Gronau [27] proved the second conjecture (therefore the first one, as well) for the integers $n = 3r + 1 \geq 13$. Bennett and Wu [4] indpendently proved the original conjecture for all $n \geq 7$ with the possible exeption $n = 8$. Somewhat later, but independently Gronau and Mullin [29] also settled the general case. (Very recently, Yeow Meng Chee [11] found a new proof for the second conjecture.)

Theorem 3.12

$$s(\mathcal{K}_3^n) = n \quad n \geq 7, n \neq 8.$$

For $k = 4, 5, \ldots, n-2, n-3, \ldots$, one cannot expect a nice formula for $s(\mathcal{K}_k^n)$. However, it is asymptotically determined for fixed k and large n. In fact, Lemma 3.10 gives an asymptotically correct lower estimate, the non-trivial construction given in [14] ensures the validity of

Theorem 3.13

$$c_1 n^{\frac{k-1}{2}} \leq s(\mathcal{K}_k^n) \leq c_2 n^{\frac{k-1}{2}}$$

where c_1 and c_2 do not depend on n.

There is a similar result for large ks.

Theorem 3.14 [26]

$$\frac{1}{12} n^2 \leq s(\mathcal{K}_{n-2}^n) \leq \frac{1}{2} n^2,$$

$$c_3 n^{\frac{2k+1}{3}} \leq s(\mathcal{K}_{n-k}^n) \leq c_4 n^k$$

where c_3 and c_4 do not depend on n.

Theorem 3.9 gives some information on the worst (in sense of minimum number of rows) key systems. It would be interesting to study smaller subclasses. We are able to offer only open problems.

Problem 3.15 *Determine* $\max s(\mathcal{K})$ *for Sperner families on n elements inducing a determination containing functional dependencies $A \to B$ satisfying $|B - A| \leq k$ where k is a fixed integer.*

Problem 3.16 *Determine* $\max s(\mathcal{K})$ *(and* $\min s(\mathcal{K})$*) for Sperner families on n elements, satisfying $|\mathcal{K}| = k$.*

Practically nothing is known about this problem. However it has a connection to another, perhaps easier problem. Let a subset $A \subseteq U$ be an *antikey* if it is not a key (= superset of a member of $\mathcal{K}$). The set of maximal antikeys is denoted by $\mathcal{K}^{-1}$. The following inequalities are known from [14]:

$$|\mathcal{K}^{-1}| \leq \binom{s(\mathcal{K})}{2}$$

and

$$s(\mathcal{K}) \leq 1 + |\mathcal{K}^{-1}|,$$

that is, there is a strong connection between $|\mathcal{K}^{-1}|$ and $s(\mathcal{K})$. This leads to another open problem.

Problem 3.17 *Determine* $\max|\mathcal{K}^{-1}|$ *and* $\min|\mathcal{K}^{-1}|$ *for Sperner families having exactly* $|\mathcal{K}| = k$ *members.*

We think that the minimum is attained for a family consisting of i and $i+1$-element subsets, where i is determined by

$$\binom{n}{i} \leq k < \binom{n}{i+1},$$

if k is not too large relative to n.

The reader probably has noticed that the last few theorems do not fit perfectly into the frames of the questions suggested at the beginning of this section. Indeed, here we considered a more general "parameter", namely the family $\mathcal{K}$ of minimal keys. However their use is similar. If our information tells us the number of attributes, as well as the size of the relation (instance) and it is smaller than the minimum in the given theorem then the family of keys in question is excluded. For instance, if the size of the instance is smaller than the number of attributes then we can be sure that the family of minimal keys cannot consist of all the $n-1$-element subsets (by Theorem 11).

Very little is known about $s(\mathcal{L})$ for closures (or equivalently determinations). Of course,

$$s(\mathcal{L}_k^n) = s(\mathcal{K}_k^n) \tag{3.2}$$

holds. Furthermore, there is a result on the s-function of direct products. It is not true for key systems.

Let $U = U_1 \cup U_2$ be a partititon of U and let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two closures defined on U_1 and U_2, resp. The *direct product* $\mathcal{L}_1 \times \mathcal{L}_2$ is defined by

$$(\mathcal{L}_1 \times \mathcal{L}_2)(A) = \mathcal{L}_1(A \cap U_1) \cup \mathcal{L}_2(A \cap U_2).$$

Theorem 3.18 [14]

$$s(\mathcal{L}_1 \times \mathcal{L}_2) = s(\mathcal{L}_1) + s(\mathcal{L}_2) - 1.$$

Theorems 3.11, 3.18 and (3.2) make us able to determine $s(\mathcal{L})$ for several closures.

4 Functional Dependencies in Random Instances

In this section we return to our second example in the introduction. Only the number of attributes and the number of tuples are known and a distribution on each domain. If such a distribution is not given we nay consider the uniform distribution. Only some probabilistic statements can be obtained for the functional dependencies of the instance, but they induce some other probabilistic consequences for the scheme, as well. Biskup [5] suggested to study random databases.

There is only a very modest result in this direction. Due to technical difficulties we were able to treat only the case when the elements of $D(a_i)$ are chosen with equal probabilities and independently. First suppose that all the domains contain exactly

two elements, that is, $|D(a_i)| = 2$ holds for all $1 \leq i \leq n$. We may also suppose that $D(a_i) = \{0, 1\}$. These values are chosen with equal $(\frac{1}{2} - \frac{1}{2})$ probabilities. All the entries (datas) are chosen totally independently. Therefore the probability of the choice of a given 0,1-sequence of length n as a tuple $r \in R$ is $\frac{1}{2^n}$.
The result is of asymtotic nature. It will be supposed that the number n of attributes (columns) tends to infinity and the number of individuals (rows) is a function of $n : m(n)$ where $m(n)$ tends to infinity with n. The investigated quantity $|A|$ is also expressed as a function of n.

Theorem 4.1 *Suppose that entries of the 0,1 matrix of $m = m(n)$ rows and n columns are chosen randomly: independently and with equal probabilities. Let $A \subset U$ be of size $r(n) = 2\log_2 m(n) + d(n)$ and suppose that $b \in U - A$. Then the probability of the event that b functionally depends on A satisfies*

$$P(A \to b) \to \begin{cases} 0 & \text{if } d(n) \text{ tends to } \infty \\ \exp(-\frac{1}{2^{d+2}}) & \text{if } d(n) \text{ tends to a finite } d \\ 1 & \text{if } d(n) \text{ tends to } -\infty. \end{cases} \quad (1)$$

Consequence 4.2 *If the number of rows is a polynomial of n, that is, $m(n) = n^h$ then (1) holds for $r(n) = 2h\log_2 n + d(n)$. On the other hand, if $m(n) = 2^{\frac{n}{2}+\log_2 n}$ then the probability of the event that there is any non-trivial functional dependency tends to 0.*

These results are generalized in [22] for the case when $D(a_i)$-s are larger and of distinct size.
The main moral of the above theorems is that there are functional dependencies $A \to b, b \notin A$ with small $|A|$ in a random relational instance if the number n of attributes is large and the size of the database is a polynomial of n. More precisely, the size of the smallest such A is constant times $\log_2 n$. The scheme can never have smaller functional dependencies than the instance has.
On the other hand, the size of R must be very large (more than $2^{\frac{n}{2}}$) to cease almost all functional dependencies. This size is impossible for large databases.

Problem 4.3 *Generalize the above results for non-uniform distributions.*

Problem 4.4 *Investigate the finer structure of the functional dependencies in the random instances. The above results determine the sizes of the typical dependencies $A \to b$. They are everywhere densely situated. However there are smaller dependencies, their number is smaller, they are placed sparsely. It seems to be much more difficult to say something about their mutual relation, or structure.*

5 Partially Ordered Set of Closures

In this section we will consider relations (databases) on a fixed attribute set U. More precisely the closures generated by them will serve as models. That is, we forget about other properties of the databases (like other types of dependencies) only the

functional dependencies are considered. A further very natural condition is added. Namely, only the closures satisfying

$$\mathcal{L}(\emptyset) = \emptyset \tag{5.1}$$

are considered. (For the determinations it means that $\emptyset \to A$ holds only for $A = \emptyset$.) A database is constantly changing during its life. This statement can be understood in two different ways. 1) The instance changes by adding a tuple or deleting one. Then the closure generated by this instance is also changed. Then the closure of the instances can be studied, knowing that it has an obvious consequence of the closure of the scheme, too. 2) The change what is observed seems so regular, so important that it forces the corresponding change in the scheme.
Both ways lead to the same mathematical model. The changes are in the matrix. A typical change is to delete the data of some individuals. If $A \to \{a\}$ $(A \subseteq U, a \in U)$ is true then it remains true after the change. This implies

$$\mathcal{L}_1(A) \subseteq \mathcal{L}_2(A) \quad (\text{for all} \quad A \subseteq U) \tag{5.2}$$

where $\mathcal{L}_1$ and $\mathcal{L}_2$ denote the closures before and after the change. We write $\mathcal{L}_1 \geq \mathcal{L}_2$ in this case. It is easy to see that this property is transitive, consequently the closures on a fixed n-element set U satisfying (5.1) form a partially ordered set (poset) for the ordering given in (5.2). The aim of the present section is to study this poset P. In Section 2 we saw that the family of closed sets is an equivalent form of a closure. A closure satisfies (5.1) iff

$$\emptyset \in \mathcal{Z}(\mathcal{L}) \tag{5.3}$$

. On the other hand it is easy to see ([8]) that

$$\mathcal{L}_1 \leq \mathcal{L}_2 \quad \text{iff} \quad \mathcal{Z}(\mathcal{L}_1) \subseteq \mathcal{Z}(\mathcal{L}_2).$$

Hence it follows that an equivalent form of P consists of the intersection semi-lattices containing $\emptyset$, ordered by inclusion as families.
It is easy to see that P has a rank function $r(\mathcal{Z}) = |\mathcal{Z}| - 2$, that is, r is zero for some element (namely, for $\mathcal{Z} = \{\emptyset, U\}$) and if $\mathcal{Z}_1 < \mathcal{Z}_2$ and there is no third element between them, then $r(\mathcal{Z}_2) = r(\mathcal{Z}_1) + 1$.
The first thing to study is the size of P. Consider the intersection semi-lattices consisting of U, some subsets of size $\lfloor \frac{n}{2} \rfloor$ and all of their intesections. They are distinct and their number is

$$2^{\binom{n}{\lfloor \frac{n}{2} \rfloor}}.$$

It was shown in [9] that the exponent in the upper estimate is at most

$$2\sqrt{2}\binom{n}{\lfloor \frac{n}{2} \rfloor}.$$

Recently Alekseyev [1] proved that $2\sqrt{2}$ can be omitted.

Theorem 5.1

$$2^{\binom{n}{\lfloor \frac{n}{2} \rfloor}} \leq |P| \leq 2^{\binom{n}{\lfloor \frac{n}{2} \rfloor}(1+o(1))}.$$

Problem 5.2 *Determine P asymptotically.*

As the number of Sperner families is determined by Korshunov [30], asymptotically (not only the asymptotics of the exponent!), there is some hope that the same can be done using Proposition 2.7 and Korshunov's theorem.
There are some initial results concerning the sizes of the lower levels of P.

Theorem 5.3 [10] *The number $\alpha(n,k)$ of the elements of rank k in P satisfies*

$$\alpha(n,k) \sim \delta(k)(k+1)^n$$

where k is fixed and n tends to infinity.

The next theorem deals with the levels near to the top. (The top rank is $2^n - 2$.)

Theorem 5.4 [10]

$$\alpha(n, 2^n - 2 - k) \sim \rho(k) n^k$$

where k is fixed and n tends to infinity.

Comparing Theorems 5.3 and 5.4 one can see that P is very asymmetric, the low levels are much wider than the top ones.

Problem 5.5 *Determine approximately the widest level of P.*

Theorems 5.3 and 5.4 suggest that the widest level is much below the middle.
To continue our investigations to understand the structure of P, the next question is to determine the minimum and maximum degrees at each level. Let $\deg_a(\mathcal{L})$ and $\deg_b(\mathcal{L})$ denote the number of edges going upward and downward, resp., from $\mathcal{L}$ in the Hasse-diagram of P. The folloing functions are defined:

$$f_1(n,k) = \max\{\deg_a(\mathcal{L}) : \quad r(\mathcal{Z}) = k\},$$
$$f_2(n,k) = \min\{\deg_a(\mathcal{L}) : \quad r(\mathcal{Z}) = k\},$$
$$f_3(n,k) = \max\{\deg_b(\mathcal{L}) : \quad r(\mathcal{Z}) = k\},$$
$$f_4(n,k) = \min\{\deg_b(\mathcal{L}) : \quad r(\mathcal{Z}) = k\}.$$
$$(1 \leq n, 0 \leq k \leq 2^n - 2).$$

$f_1(n,k)$ is fully determined, there are estimates on $f_2(n,k)$ and $f_4(n,k)$. However we know practically nothig about $f_3(n,k)$.

Theorem 5.6 [8]

$$f_1(n,k) = 2^n - 2 - k.$$

Theorem 5.7 [8]

$$f_2(n,k) = 0 \quad \textit{iff} \quad k = 2^n - 2,$$
$$f_2(n,k) = 1 \quad \textit{iff} \quad k = 2^n - 2^{n-a-1} - 2$$

for some $0 < a < n$. If $k > 2^{n-1} + 2$ then $f_2(n,k) \leq$ the number of bits 1 in the binary expansion of $2^n - k - 2$. This is at most $n - 1$.

Let us mention that the proof is based on the somewhat strange notion of $\mathcal{H}(\mathcal{Z})$, see Theorem 2.6.

Theorem 5.8 [8]

$$\lceil \log_2(k+1) \rceil \leq f_4(n,k) \leq \lfloor \log_2(k+2) \rfloor - 1 + \\ + (\textit{the number of non-zero digits in the binary form of } \quad k+2).$$

Problem 5.9 *Give estimates on* $f_3(n,k)$.

6 Branching and partial dependencies

Now we introduce a more general (weaker) dependency, than the functional dependency. We do it first in a very particular case to show the usefulness of the concept. Let $A \subseteq U$ and $b \in U$. We say that b *(1,2)-depends* on A if the values in A determine the values in a "two-valued" way. That is, there exist no three rows, same in A but having three different values in b. We denote it by $A \to (1,2) \to B$. Similarly, $A \to (1,q) \to b$ if there exist no $q+1$ rows having the same values in each column in A, but containing $q+1$ different values in the column b.

As an example, suppose that the database consists of the trips of an international transport truck, more precisely, the names of the countries the truck enters. For the sake of simplicity, let us suppose that truck goes through exactly four countries in each trip (counting the start and endpoints, too) and does not enter a country twice during one trip. Suppose furthermore, that there are 30 possible countries and one country has at most five neighbours. Let a_1, a_2, a_3, a_4 denote the countries as attributes. It is easy to see that $a_1 \to (1,5) \to a_2$, $\{a_1, a_2\} \to (1,4) \to a_3$ and $\{a_2, a_3\} \to (1,4) \to a_4$. Now, we cannot decrease the size of the stored matrix, as in the case of functional (that is, (1,1)-) dependencies, but we can decrease the range of the values in the new matrices. The domains $D(a_i)$ in the original database have 30 possible values, names of the countries or some codes of them (5 bits each, at least). Let us store a little table ($30 \times 5 \times 5 = 750$ bits) that contains a numbering of the neighbours of each country, which assigns to them the numbers 0,1,2,3,4 in some order. Now we can replace the attribute a_1 by these numbers (a_1^*), because the value in a_1 gives the starting country and the value in a_1^* determines the second country with the help of the little table. The same holds for the attribute a_3, but here the number of possible values can be even further decreased, if another table is given containing the numbering of possible third countries for each pair a_1, a_2. In this case the attribute a_3^* can take only 4 different values. The same holds for a_4, too. That is, while each value of the original relation could be encoded by 5 bits, now for the cost of two little auxiliary tables we could decrease the length of the values in the the second column to 3 bits, and that of the elements in the third and fourth columns to 2 bits.

It is easy to see, that the same idea can be applied in each case when the paths of a graph are stored, whose maximum degree is much less than the number of its vertices.

After this long motivation let us give the general definition. Fix a relation R on the set of attributes U. Let $A \subseteq U$, $b \in U$ and $1 \leq p \leq q$ integers. We say that

b (p,q)*–depends* on A if there are no $q+1$ rows of R such that they contain at most p different values in each attribute in A, but $q+1$ different values in b.
In [23] these dependencies were called *branching*. However, Grant and Minker [28] introduced the so called *numerical dependencies* which are identical to our $(1,q)$-dependencies. Their theorems are special cases of the forthcoming ones, supposing that no empty set $(1,q)$-depends on the empty set.
Define the mapping $\mathcal{I} = \mathcal{I}_{Rpq}: \quad 2^U \to 2^U$ by

$$\mathcal{I}(A) = \{b: \quad A \to (p,q) \to b\}.$$

Proposition 6.1 *$\mathcal{I}$ has the following properties:*

$$A \subseteq \mathcal{I}(A), \tag{6.1}$$

$$A \subseteq B \quad \textit{implies} \quad \mathcal{I}(A) \subseteq \mathcal{I}(B), \tag{6.2}$$

for any subsets $A, B \subseteq U$.

The set-functions satisfying conditions (6.1) and (6.2) are called *increasing-monotone functions*. Note that (6.1) is identical with (2.6) and (6.2) is identical with (2.7). An increasing-monotone function, however, does not satisfy the third property (2.8) of closures, in general.
Are these two conditions enough? We have only partial answers to this question. We say that an increasing-monotone function $\mathcal{N}$ is (p,q)-*representable* iff there is a relation R such that $\mathcal{N} = \mathcal{I}_{Rpq}$.

Theorem 6.2 [23] *Let $\mathcal{N}$ be an increasing-monotone function satisfying $\mathcal{N}(\emptyset) = \emptyset$. Then $\mathcal{N}$ is (p,q)-representable if one of the following conditions hold:*

$$p = 1 \quad \textit{and} \quad 1 < q,$$

$$p = 2 \quad \textit{and} \quad 3 < q,$$

$$2 < p \quad \textit{and} \quad p^2 - p - 1 < q.$$

Problem 6.3 *Is the statement of Theorem 6.2 true for any $p < q$? Is it possible to drop the condition $\mathcal{N}(\emptyset) = \emptyset$?*

The first undecided case is $p = 2, q = 3$. The situation is significantly different if $p = q$.

Proposition 6.4 [23] *$\mathcal{I}_{Rpp}$ is a closure for any $1 \leq p$.*

Thus, it is natural to ask if all closures are (p,p)-representable for any given p. If $p < q$ then we know that $\mathcal{I}$, in general, is not a closure. But is it true at least that all closures are (p,q)-representable? The answer, in general, is negative.

Theorem 6.5 [23] *If $p > 2, n > 6$ then $\mathcal{L}_2^n$ is not (p,p)-representable.*

The situation is better if $p = q = 2$ or $p < q$.

Theorem 6.6 [23] *Every closure is (p,q)-representable if one of the following conditions hold:*

$$p = 1 \quad \text{and} \quad 1 \leq q,$$

$$p = 2 \quad \text{and} \quad 2 \leq q,$$

$$2 < p \quad \text{and} \quad \frac{(p+1)^2}{2} < q.$$

Hence we can see that any closure is $(1,1)$-representable (we new it a lot earlier!) and $(2,2)$-representable. However it is not true for $(3,3)$-representation.

Problem 6.7 *Characterize the (3,3)-representable closures.*

One might think that this characterization, if found, is good for all (p,p). This is not true, as we will see using the following theorem.

Theorem 6.8 [37] *$\mathcal{L}_k^n$ is (p,p)-representable for $p = 1,2,3,4,2k-3,2k-2$, if $k > 2$ and is not (p,p)-representable for $\frac{3}{2}k - 1 \leq p \leq 2k-4$ and for $2k-1 \leq p$ if $n > n_0(k), k > 1$.*

For instance, $\mathcal{L}_4^n$ is (p,p)-representable for large n iff $p = 1,2,3,4,5,6$. This is a closure which is (6,6)-representable but not (7,7)-representable.
In the cases when a representation is found one can define $s_{pq}(\mathcal{N})$ as the minimum of $|R|$ for relations representing $\mathcal{N}$. In this part we do not pose open problems since they are obvious, the results are very modest.

Theorem 6.9 [24]

$$s_{1q}(\mathcal{N}) \leq 2qn2^n$$

holds for any integer $q > 1$ and increasing-monotone function $\mathcal{N}$.

Lemma 3.10 can be easily generalized for this case. This generalization helps to prove the following statement'

Theorem 6.10 [24]

$$s_{pq}(\mathcal{L}_1^n) = q + 1,$$

$$s_{22}(\mathcal{L}_2^n) = 2n \quad \text{if} \quad n > 3,$$

Theorem 6.11 [36]

$$s_{pp}(\mathcal{L}_n^n) = \min\{v : \binom{v-1}{p} \geq n\}.$$

Finally, let us only briefly mention the *partial dependencies*. The vector $\alpha = (a_{i_1}, \ldots, a_{i_k}; r_1, \ldots, r_k)$ is called a *partial function* where the a-s are elements of U and $r_h \in D(a_{i_h})$. We say that $\beta = (b_{j_1}, \ldots, b_{j_l}; s_1, \ldots, s_l)$ *depends* on α in R if each row containing r_h in the column of the attribute a_{i_h} (for all $1 \leq h \leq k$) it contains s_h in the attribute b_{j_h} (for all $1 \leq h \leq l$). The paper [21] contains investigations concerning this dependency.

References

1. V.B. Alekseyev: Diskret. Mat 1(2),129-136 (1989)
2. W.W. Armstrong: Dependency structures of data base relationship, Information Processing 74, North-Holland, Amsterdam, pp.580-583
3. A. Békéssy, J. Demetrovics, L. Hannák, P. Frankl and G.O.H. Katona: On the number of maximal dependencies in a data base relation of fixed order, Discrete Math. 30,83-88 (1980)
4. F.E. Bennett and Lisheng Wu: On minimum matrix representation of closure operations, Discrete Appl. Math. 26,25-40 (1990)
5. J. Biskup: personal communication
6. J. Biskup, J. Demetrovics, L.O. Libkin, M. Muchnik: On relational database schemes having unique minimal key, J. Information Process. Cybern. EIK, Berlin, 27,217-225 (1991)
7. B. Bollobás: On generalized graphs, Acta Math. Hungar. 16,447-452 (1965)
8. G. Burosch, J. Demetrovics , G.O.H. Katona: The poset of closures as a model of changing databases, Order 4,127-142 (1987)
9. G. Burosch, J. Demetrovics, G.O.H. Katona, D.J. Kleitman, A.A. Sapozhenko: On the number of databases closure operations, Theoret. Comput. Sci. 78,377-381 (1991)
10. G. Burosch, J. Demetrovics, G.O.H. Katona, D.J. Kleitman, A.A. Sapozhenko: On the number of databases closure operations, II, submitted to Discrete Appl. Math.
11. Yeow Meng Chee: Design-theoretic problems in perfectly (n-3)-error-correcting databases, submitted to SIAM J. Discrete Math.
12. E.F. Codd: A relational model of data for large shared data banks, Comm. ACM 13,377-387 (1970)
13. J. Demetrovics: On the equivalence of candidate keys with Sperner systems, Acta Cybernet. 4,247-252 (1979)
14. J. Demetrovics, Z. Füredi, G.O.H. Katona: Minimum matrix representation of closure operations, Discrete Appl. Math. 11,115-128 (1985)
15. J. Demetrovics, Gy. Gyepesi: A note on minimal matrix representation of closure operations, Combinatorica 3,177-180 (1983)
16. J. Demetrovics, G. Hencsey, L.O. Libkin, I.B. Muchnik: On the interaction between closure operations and choice functions with applications to relational databases, to appear in Acta Cybernet.
17. J. Demetrovics, G.O.H. Katona: Extremal combinatorial problems in relational database, In Fundamentals of Computation Theory 81, Proc. of the 1981 International FCT-Conference, Szeged, Hungary, 1981, Lecture Note In Computer Science, 117. Berlin: Springer 1981, pp.110-119
18. J. Demetrovics J., G.O.H. Katona: Combinatorial problems of database models, In Coll. Math. Soc. János Bolyai, 42. Algebra, Combinatorics and Logic in Computer Science, Győr, Hungary, 1983, 1986, pp. 331-353.
19. J. Demetrovics, G.O.H. Katona, Extremal combinatorial problems of databases, In: MFDBS'87, 1st Symposium on Mathematical Fundamentals of Database Systems, Dresden, GDR, January, 1987, Lecture Notes in Computer Science, Berlin: Springer 1987, pp.99-127.

References

1. V.B. Alekseyev: Diskret. Mat 1(2),129-136 (1989)
2. W.W. Armstrong: Dependency structures of data base relationship, Information Processing 74, North-Holland, Amsterdam, pp.580-583
3. A. Békéssy, J. Demetrovics, L. Hannák, P. Frankl and G.O.H. Katona: On the number of maximal dependencies in a data base relation of fixed order, Discrete Math. 30,83-88 (1980)
4. F.E. Bennett and Lisheng Wu: On minimum matrix representation of closure operations, Discrete Appl. Math. 26,25-40 (1990)
5. J. Biskup: personal communication
6. J. Biskup, J. Demetrovics, L.O. Libkin, M. Muchnik: On relational database schemes having unique minimal key, J. Information Process. Cybern. EIK, Berlin, 27,217-225 (1991)
7. B. Bollobás: On generalized graphs, Acta Math. Hungar. 16,447-452 (1965)
8. G. Burosch, J. Demetrovics , G.O.H. Katona: The poset of closures as a model of changing databases, Order 4,127-142 (1987)
9. G. Burosch, J. Demetrovics, G.O.H. Katona, D.J. Kleitman, A.A. Sapozhenko: On the number of databases closure operations, Theoret. Comput. Sci. 78,377-381 (1991)
10. G. Burosch, J. Demetrovics, G.O.H. Katona, D.J. Kleitman, A.A. Sapozhenko: On the number of databases closure operations, II, submitted to Discrete Appl. Math.
11. Yeow Meng Chee: Design-theoretic problems in perfectly (n-3)-error-correcting databases, submitted to SIAM J. Discrete Math.
12. E.F. Codd: A relational model of data for large shared data banks, Comm. ACM 13,377-387 (1970)
13. J. Demetrovics: On the equivalence of candidate keys with Sperner systems, Acta Cybernet. 4,247-252 (1979)
14. J. Demetrovics, Z. Füredi, G.O.H. Katona: Minimum matrix representation of closure operations, Discrete Appl. Math. 11,115-128 (1985)
15. J. Demetrovics, Gy. Gyepesi: A note on minimal matrix representation of closure operations, Combinatorica 3,177-180 (1983)
16. J. Demetrovics, G. Hencsey, L.O. Libkin, I.B. Muchnik: On the interaction between closure operations and choice functions with applications to relational databases, to appear in Acta Cybernet.
17. J. Demetrovics, G.O.H. Katona: Extremal combinatorial problems in relational database, In Fundamentals of Computation Theory 81, Proc. of the 1981 International FCT-Conference, Szeged, Hungary, 1981, Lecture Note In Computer Science, 117. Berlin: Springer 1981, pp.110-119
18. J. Demetrovics J., G.O.H. Katona: Combinatorial problems of database models, In Coll. Math. Soc. János Bolyai, 42. Algebra, Combinatorics and Logic in Computer Science, Győr, Hungary, 1983, 1986, pp. 331-353.
19. J. Demetrovics, G.O.H. Katona, Extremal combinatorial problems of databases, In: MFDBS'87, 1st Symposium on Mathematical Fundamentals of Database Systems, Dresden, GDR, January, 1987, Lecture Notes in Computer Science, Berlin: Springer 1987, pp.99-127.

20. J. Demetrovics, G.O.H. Katona, A survey of some combinatorial results concerning functional dependencies in database relations, to appear in Annals of Mathematics and Artificial Intelligence
21. J. Demetrovics, G.O.H. Katona, D. Miklós: Partial dependencies in relational databases and their realization, to appear in Discrete Appl. Math.
22. J. Demetrovics, G.O.H. Katona, D. Miklós: Functional dependencies in random relational databases, manuscript
23. J. Demetrovics, G.O.H. Katona, A. Sali: On the characterization of branching dependencies, to appear in Discrete Appl. Math.
24. J. Demetrovics, G.O.H. Katona, A. Sali: Branching dependencies in in relational databases (in Hungarian), Alkalmaz. Mat. Lapok. 15,181-196 (1990-91)
25. J. Demetrovics, Son Hua Nam: Closures and Sperner families, submitted to Coll. Math. Soc. János Bolyai, Extremal problems for families of subsets, Visegrád, Hungary, 1991
26. Z. Füredi: Perfect error-correcting databases, Discrete Appl. Math. 28,171-176 (1990)
27. B. Ganter, H.-O.O.F. Gronau: On two conjectures of Demetrovics, Füredi and Katona concerning partitions, Discrete Math. 88,149-155 (1991)
28. J. Grant, J. Minker: Normalization and Axiomatization for Numerical Dependencies, Inform. and Control, 65,1-17 (1985)
29. H.-O.O.F. Gronau and R.C. Mullin., preprint
30. A.D. Korshunov: On the number of monotone Boolean functions (in Russian), Problemy Kibernet. 38,5-108 (1981)
31. A.V. Kostochka: On the maximum size of a filter in the n-cube (in Russian), Metodi Diskretnovo Analiza 41,49-61 (1984)
32. D. Lubell: A short proof of Sperner's lemma, J. Combinat. Theory, 1,299 (1966)
33. H. Mannila, K.-J. Räihä: On the complexity of inferring functional dependencies, to appear in Discrete Appl. Math.
34. L.D. Meshalkin: A generalization of Sperner's theorem on the number of subsets of a finite set (in Russian), Teor. Veroyatnost. i Primenen. 8,219-220 (1963)
35. Andrea Rausche: On the existence of special block designs, Rostock Math. Kolloq. 35,13-20 (1985)
36. A. Sali: Extremal problems for finite partially ordered sets and matrices (in Hungarian), Thesis for "kandidátus" degree, Hungarian Academy of Sciences, Budapest, 1990
37. A. Sali: personal communication
38. E. Sperner: Ein Satz über Untermengen einer endlichen Menge, Math. Z. 27,544-548 (1928)
39. B. Thalheim: A review of reserch on Dependency Theory in Relational Databases I, II, preprint, Technische Universität Dresden, Section Mathematik, (1986).
40. B. Thalheim: On the number of keys in relational databases, to appear in Discrete Appl. Math.
41. B. Thalheim: Dependencies in Relational Databases, Leipzig: Teubner, 1991
42. K. Yamamoto: Logarithmic order of free distributive lattices, J. Math. Soc. Japan 6,347-357 (1954)