J. Indian Inst. Sci., 64 (B), Aug. 1983, Pp. 219-228

A generalization of functional dependencies in relational databases and use of boolean algebra

K. K. NAMBIAR AND V. G. TIKEKAR
Department of Applied Mathematics, Indian Institute of Science, Bangalore 560 012, India.

Received on December 7, 1982 ; Revised on July 2, 1983.

Abstract

Concept of *p*-dependency, a generalization of the notion of functional dependency, is introduced. It is established that every boolean function represents some *p*-dependency in relational database theory, in contrast to the situation with respect to functional dependency constraint. In the latter case, it is to be noted that a functional dependency is represented by a boolean term having only one uncomplemented variable. A one-to-one correspondence between the set of boolean functions and the set of *p*-dependency constraints is shown. Functional dependency naturally turns out to be a special case of *p*-dependency.

Key words: Functional dependencies, relational databases, entropy functions, boolem algebra.

1. Introduction

Many constraints like functional dependency, multivalued dependency, join dependency, and boolean dependency have been studied in relational database theory[1-4]. It is known that every functional dependency can be represented by a boolean function[5], but the boolean functions corresponding to functional dependencies form only a subclass of boolean functions. In other words, every boolean function does not necessarily correspond to a functional dependency.

In this paper, we introduce a generalization of the functional dependency constraint ; we call it *p*-dependency. The generalized notion of *p*-dependency has an important property that every boolean function will represent some *p*-dependency. Thus we have been able to establish here a one-to-one correspondence between the set of boolean functions and the set of *p*-dependencies. To be specific, a boolean function in the sum of products form, in which every term has only one variable uncomplemented, represents some functional dependency and *vice versa*[2]. On the other hand every boolean function (of whatsoever form) corresponds to some *p*-dependency.

Section 2 is devoted to preliminaries comprising the definitions and notations employed in this paper. In section 3, we review briefly the use of boolean algebra to denote functional dependencies with a view to explain our method of approach. We then establish how boolean algebra is used in the case of p-dependency constraints. This will justify our main claim in this paper that every boolean function represents some p-dependency constraint.

Some illustrative examples are included in section 4. From the examples, we notice the differences between a set of strict p-dependencies and a set containing at least one pure functional dependency. This leads us to the following result in boolean algebra. In a boolean function, expressed as a sum of products, if each product term is such that at least two of its variables are uncomplemented, then no prime implicant of this function can have only one variable uncomplemented. Section 5 summarizes some of the conclusions.

2. Preliminaries : Notations and definitions

As we are discussing some fundamental concepts, it is worthwhile that we formalize our notations and definitions, so that proper foundation for the understanding of the subject is built. A *relation* **R** on the collection $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n\}$ of attributes is a subset of the cartesian product $D_1 \times D_2 \times \ldots \times D_n$, where D_i is called the *domain* of the attribute X_i. The relation is denoted by $\mathbf{R}(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$. A, B, C, X, Y, Z, with or without subscripts are used to represent individual attributes, and U, V, W are used for representing subsets of attributes : *e.g.*, $\mathbf{U} = \{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3\}$. The elements of the relation are called *tuples* and specifically n-tuples if the relation is known to contain exactly n attributes. The n-tuples of $\mathbf{R}(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$ are designated as $(x_1, x_2, \ldots, x_n)$. If u represents a tuple in a relation $\mathbf{R}(\mathbf{U})$ and X is an attribute in **U**, then $u[\mathbf{X}]$ represents the element corresponding to X in the tuple. Similarly, if **V** is a subset of **U**, then $u[\mathbf{V}]$ is the tuple containing the elements corresponding to **V**. $u(\mathbf{V})$ is called the projection of u on **V**. $\mathbf{R}[\mathbf{V}]$, the *projection* of **R** on **V** is defined by :

$$\mathbf{R}[\mathbf{V}] \triangleq \{u[\mathbf{V}] : u \in \mathbf{R}\}.$$

If $\mathbf{R}(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$ is a relation, then **U** is said to *determine* **V** or **V** is *functionally dependent* on **U**, written as $\mathbf{U} \to \mathbf{V}$ (read as **U** determines **V** or **U** functionally determines **V**) if every pair of tuples of **R** which have the same projection on **U** also have the same projection on **V**. "**U** determines **V**" is referred to as a *functional dependency constraint*.

While investigating the basic notions of relational databases, it is irrelevant what the elements of the attribute domains are. Let us assume that they are integers and define a probability distribution corresponding to a given relation $\mathbf{R}(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$. We assign equal probabilities for all the tuples in **R** and designate the resulting distribution as $p(x_1, x_2, \ldots, x_n)$. If the number of elements in **R** is N, then the value of $p(x_1, x_2, \ldots, x_n)$ will be $1/N$ for all the tuples appearing in **R** and will be equal to zero for all the tuples not appearing in **R**. In essence, we have generated a set of n *random variables* corresponding to attributes $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$. Without causing confusion we shall designate these random variables also by the symbols $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$. Once we have constructed these random variables, it is possible to talk about the entropies associated with them. For the *entropy* of a random variable **X**, we make use of the usual definition of entropy[6] :

$$H(\mathbf{X}) \triangleq - \sum_{k=1}^{n} p_k \log_2 p_k \tag{2.1}$$

where p_k is the probability of the random variable **X** taking the k-th value. We shall need the distribution $p(x_1, x_2, \ldots, x_n)$ and its marginal distributions ; it should be clearly understood that these marginal distributions are not necessarily the distributions of the corresponding projections.

Making use of the entropy function $H(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$ it is possible to define an additive set function with its domain as boolean functions of the boolean variables $X_1, X_2, \ldots, X_n$. Assuming that it will not cause any confusion, we use the symbol H for the new *additive set function* also. Consider the set Ω containing the 2^n minterms formed out of the variables $X_1, X_2, \ldots, X_n$. Every subset of Ω corresponds to a boolean function of the variables $X_1, X_2, \ldots, X_n$, *viz.*, the sum of the minterms contained in the subset. The collection which has these subsets or equivalently these functions as elements is obviously an *additive class of sets* (alternatively called as σ-algebra, or σ-field). See Munroe[7] for the definition of additive class. We then define H by

$$H(X_1 + X_2 + \ldots + X_m) \triangleq H(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m) \tag{2.2}$$

and

$$\begin{aligned} H(Y_1 Y_2 \ldots Y_r \bar{Z}_1 \bar{Z}_2 \ldots \bar{Z}_{n-r}) \triangleq & \sum_{i=1}^{r} H(Y_i + C) \\ & - \sum_{\substack{i,j=1 \\ i<j}}^{r} H(Y_i + Y_j + C) \\ & + \sum_{\substack{i,j,k=1 \\ i<j<k}}^{r} H(Y_i + Y_j + Y_k + C) \\ & - \ldots \\ & + (-1)^{r+1} H(Y_1 + Y_2 + \ldots + Y_r + C) \\ & - H(C) \end{aligned} \tag{2.3}$$

where

$$C = Z_1 + Z_2 + \ldots Z_{n-r}$$

The value corresponding to the minterm $\bar{X}_1 \; \bar{X}_2 \ldots \bar{X}_n$ can be taken as zero if there are no more variables under consideration than $X_1, X_2, \ldots, X_n$ or if this is not the case, the value can be chosen as a non-negative number as required. Once this non-negative number is specified, the total entropy can be written as

$$H(X_1 + X_2 + \ldots) = H(X_1 + X_2 + \ldots + X_n) + H(\bar{X}_1 \; \bar{X}_2 \ldots \bar{X}_n). \quad (2.4)$$

In passing we note that a parallel for the formulae (2.3)–(2.4) exists in the theory of probability wherein we have exactly the same formulae if H is replaced by the probability P and X_i's, Y_i's and Z_i's are the events.

We now quote two important results which we shall invoke in establishing our main contribution in this paper. These results are proved elsewhere[1,6].

Result 1 : The functional dependency

$$\{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3\} \rightarrow \mathbf{X}_4$$

holds if and only if the entropy

$$H(\bar{X}_1\bar{X}_2\bar{X}_3X_4) = 0.$$

A similar result is true for any other functional dependency statement.

Result 2 : In a relation $\mathbf{R}(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$, $H(\bar{X}_1X_2) = 0$ if and only if the entropy of every term appearing in the minterm expansion of $\bar{X}_1X_2$ is zero, *i.e.*, any number of variables can be concatenated to the product $\bar{X}_1X_2$ without disturbing the equality. A similar result holds for any entropy equation corresponding to functional dependencies.

Now we state the definition of our generalized concept of *p-dependency* (*p* standing for 'partial'). If $\mathbf{R}(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$ is a relation, then $\mathbf{U}$ is said to *p*-determine $\mathbf{V}$ or $\mathbf{V}$ is *p*-dependent on $\mathbf{U}$, written as $\mathbf{U} \xrightarrow{p} \mathbf{V}$, if $\mathbf{U}$ functionally determines *at least one* of the attribtues of $\mathbf{V}$. For example,

$$\{\mathbf{X}_1, \mathbf{X}_2\} \xrightarrow{p} \{\mathbf{X}_3, \mathbf{X}_4, \mathbf{X}_5\}$$

means

$$\{\mathbf{X}_1, \mathbf{X}_2\} \rightarrow \mathbf{X}_3$$

or $\{\mathbf{X}_1, \mathbf{X}_2\} \rightarrow \mathbf{X}_4$

or $\{X_1, X_2\} \rightarrow X_5$

or $\{X_1, X_2\} \rightarrow \{X_3, X_4\}$

or $\{X_1, X_2\} \rightarrow \{X_3, X_5\}$

or $\{X_1, X_2\} \rightarrow \{X_4, X_5\}$

or $\{X_1, X_2\} \rightarrow \{X_3, X_4, X_5\}$.

In other words, $\{X_1, X_2\}$ functionally determines at least one of the non-null subsets of $\{X_3, X_4, X_5\}$.

3. Justification of boolean algebra to represent *p*-dependency constraints

In order to explain the basis of our use of boolean algebra to represent *p*-dependency constraints and also for the sake of completeness we explain first how boolean algebra is utilized in the case of functional dependencies.

The transition from functional dependencies to boolean algebra is accomplished as follows. First, given functional dependencies are transformed to corresponding entropy equations using Result 1 of section 2. Then, the entropy statements can be represented in a Venn diagram. Venn diagram representation is then converted to the corresponding Karnaugh map which in turn gives the boolean function for the functional dependencies with which we started. Take, for example, the functional dependencies

$$\mathbf{A} \rightarrow \mathbf{B}$$

and

$$\mathbf{B} \rightarrow \mathbf{C}$$

where **A**, **B**, **C** are three attributes. The corresponding entropy statements are given by

$$H(\bar{A}B) = 0$$

and

$$H(\bar{B}C) = 0.$$

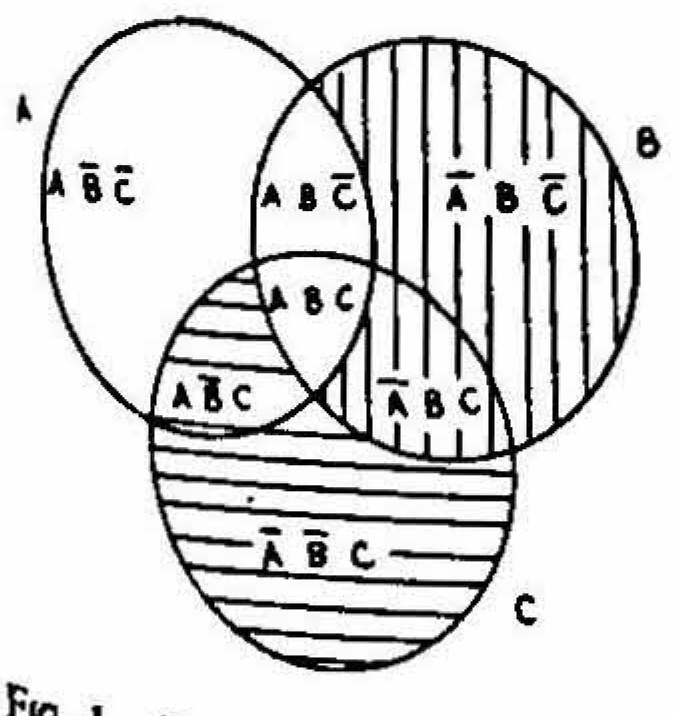

FIG. 1. Venn diagram corresponding to the entropy equations.

The Venn diagram that depicts these entropy equations is given in fig. 1; the corresponding portions of the diagram are hatched. In view of Result 2 of section 2, entropy of each hatched portion, $H(\bar{A}\bar{B}C)$, $H(\bar{A}BC)$, $H(\bar{A}B\bar{C})$, $H(A\bar{B}C)$, is separately zero. From the Venn diagram, one can see that $H(\bar{A}\ C) = 0$. This entropy equation yields the functional dependency constraint, *viz.*, $A \rightarrow C$ which has to be true in view of the transitivity law when applied to the given functional dependencies $A \rightarrow B$ and $B \rightarrow C$. Since there always exists a Karnaugh map corresponding to every Venn diagram, we can draw a Karnaugh map to represent whatever is depicted by the Venn diagram. Let us enter 1 or zero in the Karnaugh map according as the entropy of the corresponding portion of the Venn diagram is zero or non-zero. The Karnaugh map corresponding to fig. 1 is given in fig. 2.

It is easy to see that the prime implicants of this function are $\bar{A}B$, $\bar{B}C$, and $\bar{A}C$ as expected and the function itself can be written as

$$\bar{A}B + \bar{B}C + \bar{A}C.$$

Let us now proceed to establish how boolean algebra is utilized in the case of *p*-dependency constraints. For definiteness and simplicity we shall restrict ourselves to five attributes X_1, X_2, X_3, X_4, X_5. But our proof is quite general and therefore applies to any *p*-dependency constraint involving any number of attributes. Consider the *p*-dependency constraint :

$$\{X_1, X_2\} \xrightarrow{p} \{X_3, X_4, X_5\} \tag{3.1}$$

By definition of *p*-dependency given in section 2, (3·1) implies

$$\begin{aligned} & \{X_1, X_2\} \rightarrow X_3 \\ \text{or}\ & \{X_1, X_2\} \rightarrow X_4 \\ \text{or}\ & \{X_1, X_2\} \rightarrow X_5 \\ \text{or}\ & \{X_1, X_2\} \rightarrow \{X_3, X_4\} \\ \text{or}\ & \{X_1, X_2\} \rightarrow \{X_3, X_5\} \\ \text{or}\ & \{X_1, X_2\} \rightarrow \{X_4, X_5\} \\ \text{or}\ & \{X_1, X_2\} \rightarrow \{X_3, X_4, X_5\}. \end{aligned} \tag{3.2}$$

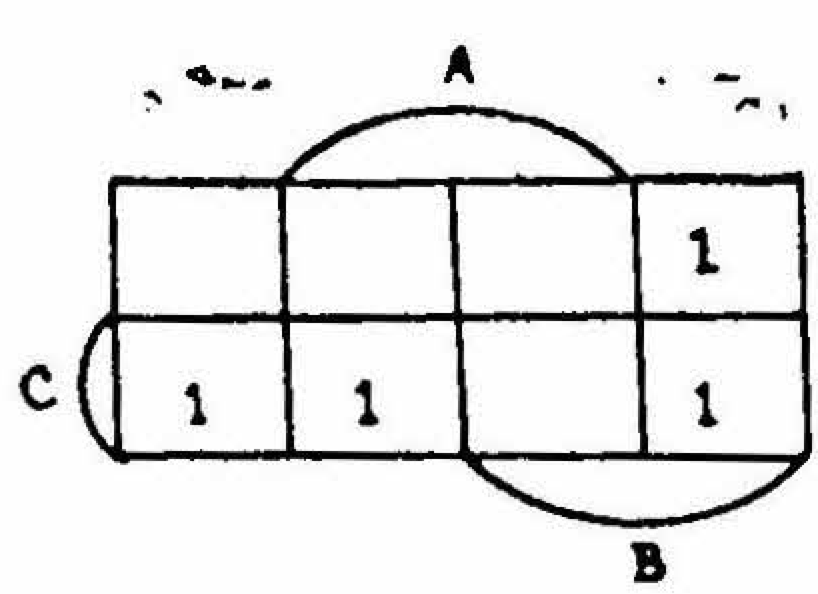

FIG. 2. Karnaugh map corresponding to fig. 1.

In view of Result 1 of section 2, we have from (3.2)

$$H(\bar{X}_1\bar{X}_2X_3) = 0$$

$$\text{or} \quad H(\bar{X}_1\bar{X}_2X_4) = 0$$

$$\text{or} \quad H(\bar{X}_1\bar{X}_2X_5) = 0$$

$$\text{or} \quad H(\bar{X}_1\bar{X}_2X_3) = 0 \text{ and } H(\bar{X}_1\bar{X}_2X_4) = 0$$

$$\text{or} \quad H(\bar{X}_1\bar{X}_2X_3) = 0 \text{ and } H(\bar{X}_1\bar{X}_2X_5) = 0$$

$$\text{or} \quad H(\bar{X}_1\bar{X}_2X_4) = 0 \text{ and } H(\bar{X}_1\bar{X}_2X_5) = 0$$

$$\text{or} \quad H(\bar{X}_1\bar{X}_2X_3) = 0 \text{ and } H(\bar{X}_1\bar{X}_2X_4) = 0 \text{ and } H(\bar{X}_1\bar{X}_2X_5) = 0. \tag{3.3}$$

Now in view of Result 2 of section 2, if (3.3) has to be true, it is compulsory that

$$H(\bar{X}_1\bar{X}_2X_3X_4X_5) = 0 \tag{3.4}$$

The boolean function associated with this entropy equation is

$$\bar{X}_1\bar{X}_2X_3X_4X_5. \tag{3.5}$$

Thus we have established that the p-dependency constraint (3.1) is represented by the boolean function (3.5). Thus, in general, the boolean function corresponding to the p-dependency constraint

$$\{X_1, X_2, \ldots, X_k\} \xrightarrow{p} \{Y_1, Y_2, \ldots, Y_m\}$$

is given by

$$\bar{X}_1\bar{X}_2 \ldots \bar{X}_k \; Y_1Y_2 \ldots Y_m.$$

A very special case of the p-dependency constraint occurs when a set of attributes determines every other attribute in the relation, the so-called total dependency defined by Nambiar[3]. Thus if X_1, X_2, X_3 determine every other attribute in a relation, the corresponding boolean function is $\bar{X}_1\bar{X}_2\bar{X}_3$. Note that the interpretation of the boolean function $X_1X_2X_3X_4$ is that at least one of these four columns in the relation is a constant.

4. Illustrative examples

In this section we work out two illustrative examples involving p-dependencies and as an aside point to an interesting result concerning prime implicants.

Example 4·1: Consider the p-dependencies

$$A \xrightarrow{p} \{B, C\}$$

and

$$\{\mathbf{A}, \mathbf{B}\} \overset{p}{\rightarrow} \mathbf{C}. \tag{4.1}$$

The boolean function representing these p-dependency constraints is given by

$$\bar{A}BC + \bar{A}\bar{B}C. \tag{4.2}$$

It can be easily seen by drawing the Karnaugh map for this function that its only prime implicant is

$$\bar{A}C$$

implying

$$A \rightarrow C. \tag{4.3}$$

Thus the given two p-dependencies (4.1) together imply the functional dependency (4.3). This fact can easily be deduced by straightforward arguments for this simple example.

Example 4.2 : Consider the strict p-dependencies

$$\mathbf{A} \overset{p}{\rightarrow} \{\mathbf{B}, \mathbf{C}\}$$

$$\mathbf{B} \overset{p}{\rightarrow} \{\mathbf{D}, \mathbf{E}\}$$

and

$$\mathbf{C} \overset{p}{\rightarrow} \{\mathbf{D}, \mathbf{E}\}. \tag{4.4}$$

All of these are strict p-dependencies in the sense that none of them is a functional dependency. The boolean function representing (4.4) is

$$\bar{A}BC + \bar{B}DE + \bar{C}DE. \tag{4.5}$$

The Karnaugh map of this function is shown in fig. 3. From the map we get

$$\bar{A}DE,\ \bar{A}BC,\ \bar{B}DE,\ \bar{C}DE$$

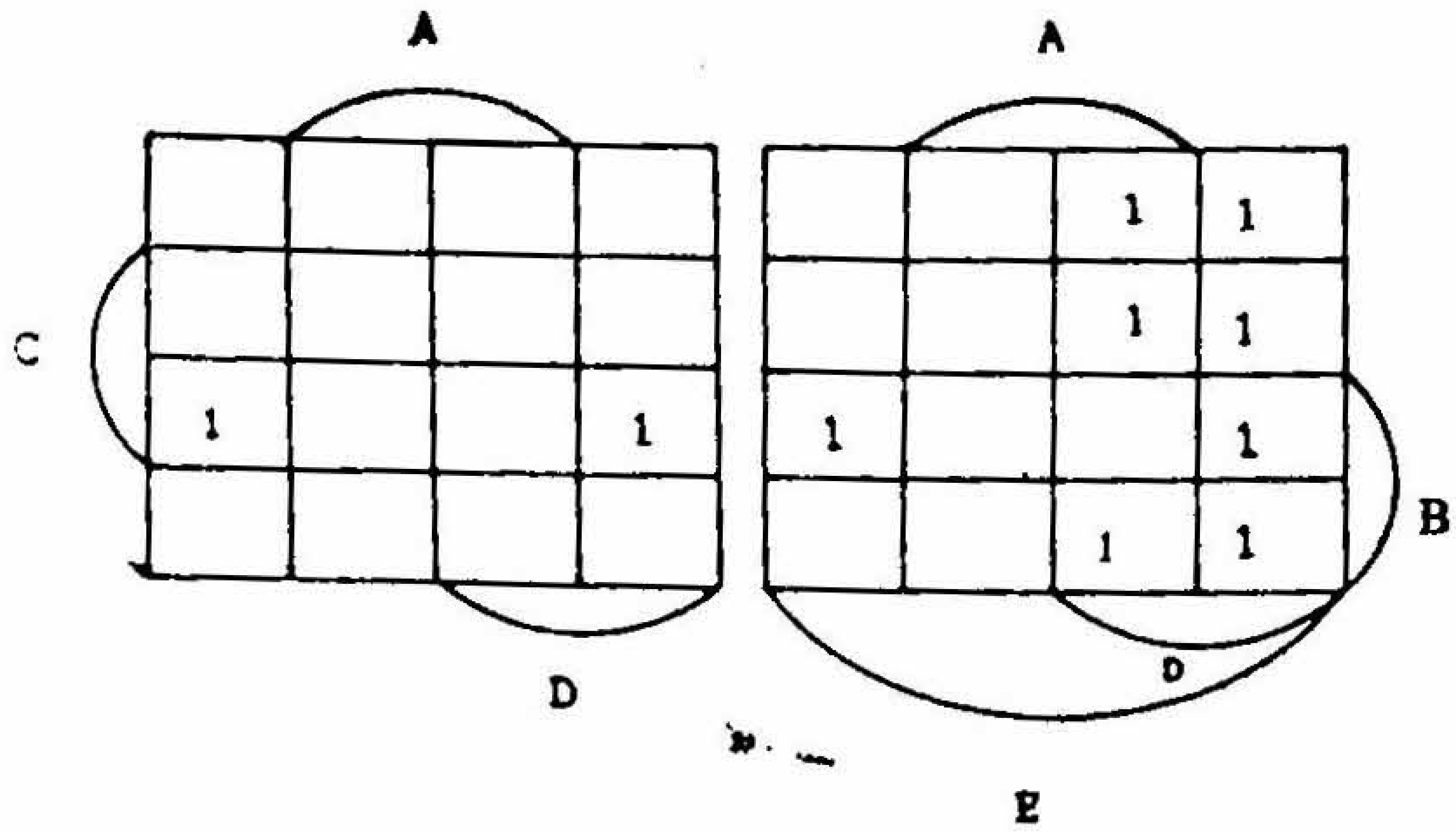

FIG. 3. Karnaugh map corresponding to (4.5).

as the prime implicants of (4.5). Thus the use of boolean algebra to represent p-dependencies gives us a simple way to identify an additional p-dependency, *viz.*,

$$\mathbf{A} \overset{p}{\rightarrow} \{\mathbf{D}, \mathbf{E}\}$$

implied by the given p-dependencies (4.4). If we do not employ boolean algebra we will have to advance a long winding qualitative argument to arrive at this result. One can easily appreciate that such an argument will be extremely cumbersome if we are dealing with a large number of attributes, as will be the case in any practical situation involving databases. This substantiates our claim that use of boolean algebra is a natural method of representing p-dependency constraints in relational databases.

Consideration of our examples leads us to an interesting result in boolean algebra which we wish to state now. We observe that in example 4.1, the given set of p-dependencies has in it one pure functional dependency, *viz.*, $\{\mathbf{A}, \mathbf{B}\} \rightarrow \mathbf{C}$. We observed that the given p-dependencies implied a pure functional dependency, *viz.*, $\mathbf{A} \rightarrow \mathbf{C}$. In example 4.2, on the other hand, we were given strict p-dependencies (4.4) to start with, and none of them was a functional dependency. (There were more than one attribute on the right hand side of the symbol $\overset{p}{\rightarrow}$). Here we note that the given strict p-dependencies implied another strict p-dependency only, *viz.*, $\mathbf{A} \overset{p}{\rightarrow} \{\mathbf{D}, \mathbf{E}\}$.

A little thought over similar sets of constraints shows that if all the given constraints are strict p-dependencies, it is impossible that they together imply a pure functional dependency. To convince ourselves that this is really so we observe that in any p-dependency constraint, the set of attributes on the left hand side of $\overset{p}{\rightarrow}$ functionally determines at least one of the non-null subsets of attributes appearing on the right hand side of $\overset{p}{\rightarrow}$. Such being the case, it is impossible that a number of strict p-dependencies together can capture one single attribute which will be functionally determined by a set of attributes. Thus unless there is at least one pure functional dependency constraint in a given set of p-dependency constraints, the given set cannot imply a pure functional dependency constraint.

The consequence of the result regarding strict p-dependencies stated above, when translated purely in terms of boolean algebra is as follows : If, in a boolean function expressed as a sum of products, each term is such that at least two of its variables are uncomplemented, then no prime implicant of this boolean function can have only one variable uncomplemented.

5. Conclusions

The basic aim of the investigations reported in this paper is to provide the mathematical foundations for some of the fundamental concepts of relational databases, The entropy functions and its generalisation to the additive set functions have been found useful for this purpose. We have shown that boolean algebra provides a natural approach to the study of p-dependency constraints in relational database theory. An important fact derived in this paper is that there is one-to-one correspondence between sets of p-dependencies and boolean functions.

References

1. NAMBIAR, K. K. Some fundamental concepts of relational database, *J. Comput. Soc. India*, June 1978, 8 (2), 41–48.

2. NAMBIAR, K. K. Some analytical tools for the design of relational database systems, *Proc. Sixth Conf. Very Large Data Bases*, Montreal, 1980, pp. 417–428.

3. SAGIV, Y., DELOBEL, C. PARKER, D. S. AND FAGIN, R. An equivalence between relational database dependencies and a subclass of propositional logic, *JACM*, July 1981, 28 (3), 435–453.

4. FAGIN, R. Functional dependencies in a relational database and propositional logic, *IBM J. Res. Dev.*, November 1977, 21 (6), 534–544.

5. DELOBEL, C. AND CASEY, R. G. Decomposition of a database and the theory of boolean switching functions, *IBM J. Res. Dev.*, September 1973, 17 (5), 374–386.

6. KHINCHIN, A. I. *Mathematical foundations of information theory*, Dover Publications, New York, 1957.

7. MUNROE, M. E. *Introduction to measure and integration*, Addison-Wesley, Reading, Mass., 1953.