GENERIC DEPENDENCIES AND DATABASE DESIGN

K. K. NAMBIAR AND VINOD KANNOTH

Abstract. The concept of functional dependencies in databases is generalized and called generic dependencies. Just as Karnaugh Map exhibits all the functional dependencies in a relation, Entropy Map represents all the generic dependencies. A generalized normal form useful in database design is defined.

Keywords—Generic dependency, Generalized normal form.

1. INTRODUCTION

The purpose of this paper is to introduce some mathematical concepts in the design of databases that can be useful to a data administrator. An important tool introduced is that of an entropy map, which gives a better measure of the dependencies in a relation than the karnaugh map. The generalized normal form given here is based on the values of the entropy function.

2. DEFINITIONS AND NOTATIONS

Some selected brief definitions are given below for two reasons. One is to make the paper reasonably self-contained, and the other is to give the definitions that are most suitable for our purposes.

Relation: A subset of a cartesian product $D_1 \times D_2 \times \ldots \times D_n$. A relation can be visualized as a table. The notation we use for the relation is $\{X_1, X_2, \ldots X_n\}$ or some times even $X_1X_2 \ldots X_n$, each X_k representing a column.

Tuples: Rows of the table.

Attributes: Columns of the table. We will use the names, attributes and columns, interchangeably.

Sifting Function: The function with the cartesian product $D_1 \times D_2 \times \ldots \times D_n$ as the domain and $\{0, 1\}$ as codomain, specifying the tuples of the relation with the value 1. The notation we use for the sifting function is $S(x_1, x_2, \ldots x_n)$.

Arity: The number of attributes in a relation.

$ABC \ldots K$: A string $ABC \ldots K$ can mean any of three things: a boolean term $ABC \ldots K$, a relation consisting of attributes $A, B, C, \ldots K$ or a set $\{A, B, C, \ldots K\}$. The meaning is to be taken from the context.

Join: The relation obtained from a product of sifting functions. Our symbol for the join operator is $\bowtie$, for example, $AB \bowtie AC \bowtie AD$ for the join of the relations AB, AC, and AD.

Date: June 12, 2000.

Boolean Projection: The table corresponding to a subset of the columns. Repetition of tuples are ignored in a boolean projection. In the literature, boolean projection is called, just projection.

Regular Projection: When the repetitions of tuples in a subset of columns are not ignored, we get the regular projection. With each tuple is associated a natural number, giving the number of repetitions of that tuple. If a tuple does not occur in the projection, that tuple gets the value zero.

Possibility Function: The function which specifies the regular projection, with the tuples as domain and natural numbers as codomain.

Hamming Weight: The total number of times the value 1 occurs in the sifting function of a boolean projection. Our notation for hamming weight is $\|S(x_1, x_2, \ldots x_n)\|$.

Real Weight: The sum of the values of the possibility function. Clearly, the real weight of any regular projection is equal to the hamming weight of the original relation.

Possibility Distribution: The function obtained by uniformly dividing the values of the possibility function by the hamming weight of the original relation. Our notation for the possibility distribution is $P(x_1, x_2, \ldots x_n)$. Since the values of the possibility disribution adds up to unity, it can be considered as a probability distribution.

Entropy: If the values of a possibility distribution $P(y_1, y_2, \ldots y_m)$ are listed as $p_1, p_2, p_3, \ldots$, then the entropy of the possibility function is defined as

$$H(Y_1 + Y_2 + \ldots Y_m) = -\sum_k p_k \log p_k,$$

where log is with respect to the base 2. Note that we have defined an entropy for a possibility function and not for a probability distribution, to make it clear that every partition of a number has an entropy associated with it.

Lossless Decomposition: A set of boolean projections of a relation, whose join gives the original relation.

Database: A set of relations.

Horn Function: A disjunctive boolean expression in which every term has all literals complemented except for exactly one.

Key: A minimal set of columns which determines all the other columns in a relation. Here, the meaning of the word, determines, is in the usual literary sense, the strict meaning is given later.

Determinant: A minimal set of columns which determines another column.

Saturated Set: A maximal set of columns which cannot determine any other column in the relation.

Boyce-Codd Normal Form: A relation is in Boyce-Codd Normal Form (BCNF), if every determinant in it is a key. A database is in BCNF, if all the relations in it are in BCNF.

Lattice: A class of subsets of a set, closed under intersection.

Partial Order: A class of subsets of a set, with no restrictions.

Inclusion-Exclusion Principle: The principle which allows us to write the probability of the event A or B or C in terms of the probability of simultaneous events:

$$\begin{aligned} Pr(A+B+C) &= Pr(A)+Pr(B)+Pr(C) \\ &- Pr(AB)-Pr(AC)-Pr(BC) \\ &+ Pr(ABC) \end{aligned}$$

More detailed definitions of some of the terms above and also the basic ideas of the theory used in the following discussion can be seen in [1, 2, 3, 4].

3. FUNCTIONAL DEPENDENCIES

A set of attributes, say $\{X_1, X_2, X_3\}$, is said to *determine* another attribute X_4, written as $X_1X_2X_3 \rightarrow X_4$, if

$$\|S(x_1, x_2, x_3)\| = \|S(x_1, x_2, x_3, x_4)\|.$$

In other words, $\{X_1, X_2, X_3\}$ determines X_4, or X_4 is functionally dependent on $\{X_1, X_2, X_3\}$, if $\{X_1, X_2, X_3\}$ is a superset of a determinant of X_4. An important factor that goes into the design of databases is the functional dependencies in the data.

Example 1.

Consider as an illustration, the relation given in Figure 1, in which there are two functional dependencies, $AB \rightarrow C$ and $C \rightarrow A$. In the discussion that follows, take $A, B, C, \ldots$ as the same as $X_1, X_2, X_3, \ldots$ respectively, if necessary.

A	B	C
a_0	b_0	c_0
a_0	b_1	c_1
a_1	b_0	c_2
a_1	b_1	c_2

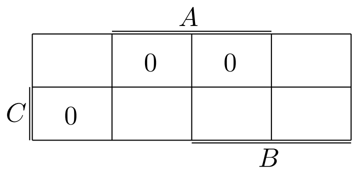
Figure 1

We will carry out the analysis in detail for this example, which should give some idea about the various tools available for the design of databases.

It is known that a horn function can represent the dependencies in a relation, in our case the function is

$$f = \overline{A}\,\overline{B}C + \overline{C}A.$$

It turns out that the complement of this function has interesting properties. The karnaugh map for $\overline{f}$ is as given in Figure 2, where the nonzero values have been left unmarked.



	0	0	
0			

Figure 2

As a sum of minterms,

$$\overline{f} = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}C + ABC.$$

If we collect those literals which are complemented in each term, we get the following class of sets

$$\{ABC, AC, A, B, \phi\}.$$

It is easy to verify from the given horn function that each element of this class is a saturated set [1, 2]. Since the class is closed under intersection, we can draw a lattice corresponding to it. If we inspect the lattice from the bottom looking for the appearance of new literals, and coalesce appropriate nodes, we arrive at a collection of relations which are in BCNF. This whole process is shown in the sequence of graphs given in Figure 3.

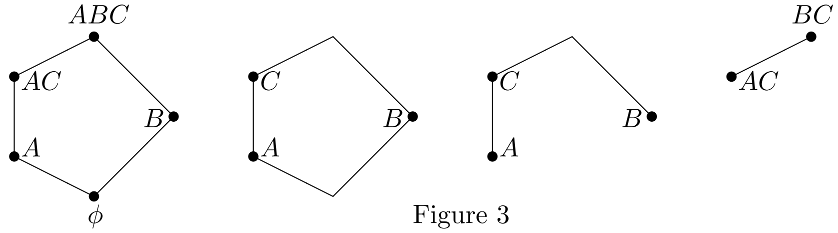


Figure 3

The BCNF decomposition is given by $\{AC, BC\}$. The join of these relations will give the original relation in Figure 1 back.

Example 2.

As another example, consider the relation given in Figure 4, with dependencies $AD \rightarrow C$, $BC \rightarrow D$, $C \rightarrow A$, and $D \rightarrow B$.

A	B	C	D
a_0	b_0	c_0	d_0
a_0	b_1	c_0	d_1
a_1	b_0	c_1	d_0
a_1	b_1	c_2	d_2

Figure 4

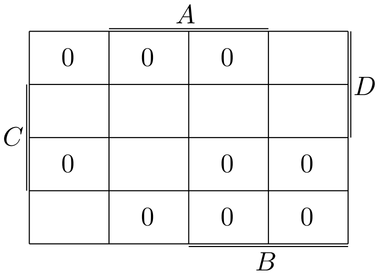
The corresponding horn function is

$$f = \overline{A}\,\overline{D}C + \overline{B}\,\overline{C}D + \overline{C}A + \overline{D}B$$

and

$$\overline{f} = \overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}\,\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BCD + A\overline{B}C\overline{D} + A\overline{B}CD + ABCD$$

The karnaugh map is as given in Figure 5, where the nonzero values have been left unmarked.



0	0	0	
0		0	0
	0	0	0

Figure 5

The class of saturated sets is given by

$$\{ABCD, AB, AC, BD, A, B, \phi\}.$$

The sequence of graphs which gives the BCNF is shown in Figure 6.

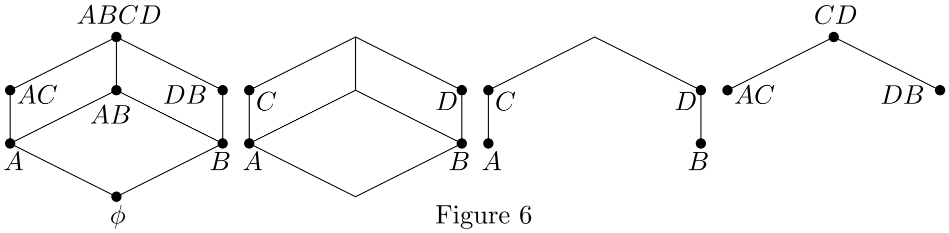


Figure 6

The BCNF decomposition is $\{AC, CD, DB\}$. The join of these relations will give the original relation in Figure 4 back.

4. GENERIC DEPENDENCIES

In a relation $\{X_1, X_2, \ldots X_n\}$, a set of attributes, say $\{X_1, X_2, X_3\}$, *generate* another attribute X_4, written as $X_1X_2X_3 \mapsto X_4$, if

$$P(x_1, x_2, \ldots x_n) = \frac{P(x_1, x_2, x_3, x_4)P(x_1, x_2, x_3, x_5, \ldots x_n)}{P(x_1, x_2, x_3)}.$$

Generic dependencies are not to be confused with the multivalued dependencies, extensively discussed in the literature. The definition of multivalued dependencies is in terms of sifting functions, whereas, our definition is in terms of possibility distributions.

Referring to Example 1, we note that the possibility distribution can be written as

$$P(x_1, x_2, x_3) = \frac{P(x_1, x_3)P(x_2, x_3)}{P(x_3)}.$$

From this we conclude that $C \mapsto B$.

Referring to Example 2, we note that the possibility distribution can be written as

$$P(x_1, x_2, x_3, x_4) = \frac{P(x_1, x_3, x_4)P(x_2, x_3, x_4)}{P(x_3, x_4)}.$$

From this we conclude that $CD \mapsto B$.

These examples should not give the impression that generic dependencies can occur only when functional dependencies are present.

Example 3.

Consider the relation in Figure 7, to take a closer look at generic dependencies. In this relation, there are no functional dependencies, in fact, the only dependencies are $A \mapsto B, A \mapsto C, A \mapsto D$.

A	B	C	D
a_0	b_0	c_0	d_0
a_0	b_0	c_0	d_1
a_0	b_0	c_1	d_0
a_0	b_0	c_1	d_1
a_0	b_1	c_0	d_0
a_0	b_1	c_0	d_1
a_0	b_1	c_1	d_0
a_0	b_1	c_1	d_1
a_1	b_1	c_1	d_1

Figure 7

The entropies of real projections of the relation are as follows.

$$\begin{aligned}
H(X_1) &= \log 9 - \frac{24}{9} \\
H(X_2) = H(X_3) = H(X_4) &= \log 9 - \frac{8}{9} \\
H(X_1 + X_2) = H(X_1 + X_3) = H(X_1 + X_4) &= \log 9 - \frac{16}{9} \\
H(X_2 + X_3) = H(X_2 + X_4) = H(X_3 + X_4) &= \log 9 - \frac{6}{9} \\
H(X_1 + X_2 + X_3) = H(X_1 + X_2 + X_4) = H(X_1 + X_3 + X_4) &= \log 9 - \frac{8}{9} \\
H(X_2 + X_3 + X_4) &= \log 9 - \frac{2}{9} \\
H(X_1 + X_2 + X_3 + X_4) &= \log 9.
\end{aligned}$$

A slight variation of the inclusion-exclusion principle (IEP) allows us to write the probabilities of simultaneous events in terms of other probabilities. For example, we can write,

$$\begin{aligned}
Pr(ABC) &= Pr(A) + Pr(B) + Pr(C) \\
&- Pr(A + B) - Pr(A + C) - Pr(B + C) \\
&+ Pr(A + B + C).
\end{aligned}$$

A slight generalization of the IEP allows us to write the probability of the conditional event BCD, when A is given as,

$$\begin{aligned}
Pr(\bar{A}BCD) &= Pr(A + B) + Pr(A + C) + Pr(A + D) \\
&- Pr(A + B + C) - Pr(A + B + D) - Pr(A + C + D) \\
&+ Pr(A + B + C + D) \\
&- Pr(A).
\end{aligned}$$

Making use of these identities, we can write the entropies of any minterm of the boolean expression $A + B + C + D$. For example,

$$\begin{aligned}
H(\overline{A}BCD) &= H(A+B)+H(A+C)+H(A+D) \\
&- H(A+B+C)-H(A+B+D)-H(A+C+D) \\
&+ H(A+B+C+D) \\
&- H(A) \\
&= 3(\log 9 - \frac{16}{9}) \\
&- 3(\log 9 - \frac{8}{9}) \\
&+ \log 9 \\
&- (\log 9 - \frac{24}{9}) \\
&= 0
\end{aligned}$$

If we calculate the entropies of the rest of the minterms, we get the entropy map as shown in Figure 8, where the nonzero entries have been left unmarked.

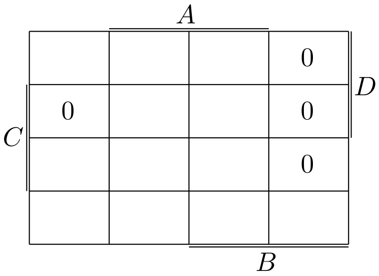


Figure 8

Corresponding to this entropy map we can write a boolean function

$$f = \overline{A}(BC+BD+CD)$$

and claim that the function represents the generic dependencies in the relation. The generic dependency $A \mapsto B$, present in the relation is exhibited by the expression $\overline{A}B(C+D)$ contained in the function. We can now generalize most of the notions connected with the functional dependencies. One obvious fact is that a functional dependency implies the corresponding generic dependency. The following definitions pertain to generic dependencies.

Generant: A minimal set of columns which generates another column.

Generic Key: A minimal set of columns which generates all the other columns.

Generic Normal Form: A relation is in generic normal form (GNF), if all the generants in it are generic keys. A database is in GNF, if all the relations in it are in GNF.

Closed Set: A maximal set of columns which cannot generate any other column in the relation. The class of closed sets is not just a partial order, but like the class of saturated sets, forms a lattice, as shown below.

To get the class of closed sets, keep all the functional dependencies in the relation, and consider the generic dependencies also as functional dependencies. The saturated sets we get, will be closed sets. For our example, the closed sets are $\{ABCD, BCD, BC, BD, CD, B, C, D, \phi\}$. Considering the lattice corresponding to this class of sets, we can proceed to decompose the relation into GNF as shown in Figure 9. However, in our simple example, A is the only generant present in the relation, and hence we can conclude that it is already in GNF.

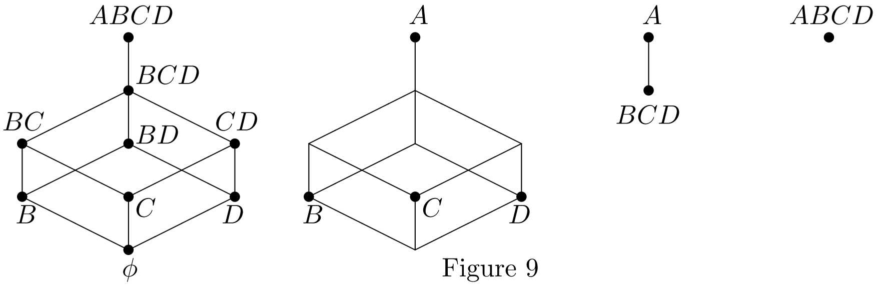


Figure 9

We have brought in entropies in our discussion only to arrive at the fact that any generic dependency has a boolean function associated with it. For example, as implied earlier, a generic dependency $A \mapsto B$ in a relation $\{A, B, C, D\}$ can be represented by the boolean function $\overline{A}B(C + D)$.

The analysis of the generic dependencies in our example would be as follows:

Given dependencies: $A \mapsto B, A \mapsto C, A \mapsto D$.
Boolean function: $\overline{A}B(C + D) + \overline{A}C(B + D) + \overline{A}D(B + C)$
$= \overline{A}(BC + BD + CD)$.
Closed set: $\{ABCD, BCD, BC, BD, CD, B, C, D, \phi\}$.
Generalized normal form: The relation $\{A, B, C, D\}$ is already in GNF.

Even though, this example is simple and contrived, it does illustrate that the generic dependency is a generalized form of a functional dependency.

5. CONCLUSION

While the boolean expression $\overline{A}\,\overline{B}C$ just tells us that AB determines C, the value of the entropy $H(\,\overline{A}\,\overline{B}C)$ gives us a much better measure of the uncertainty of the attribute C when A and B are known. If fact, it will not be unreasonable to say that any kind of relationship, whatsoever, between the attributes will get reflected in the entropy map. Thus, it is not very surprising that we have been able to carry out our analysis of generic dependencies using the entropy concept.

References

1. W. W. Armstrong, *Dependency Structures of Database Relationships*, Information Processing 74, North Holland, Amsterdam, (1974).
2. K. K. Nambiar, *Some Analytic Tools for the Design of Relational Database Systems*, Proceedings of the Sixth International Conference on Very Large Databases, Montreal, (1980).
3. K. K. Nambiar, et al, *Boyce-Codd Normal Form Decomposition*, Computers and Mathematics with Applications **33**, no. 4, 1–3, (1997).
4. C. J. Date, *An Introduction to Database Systems*, Narosa Publishing House, New Delhi, (1995).

Formerly, Jawaharlal Nehru University, New Delhi, 110067, India
Current address: 1812 Rockybranch Pass, Marietta, Georgia, 30066-8015
E-mail address: `nambiar@mediaone.net`

Prosoft Technologies Inc., 45 Swift Street, South Burlington, Vermont, 05403
E-mail address: `vinod.kannoth@prosoft-tech.com`