Journal of Mathematics and Statistics 10 (2): 231-243, 2014
ISSN: 1549-3644

doi:10.3844/jmssp.2014.231.243 Published Online 10 (2) 2014 (http://www.thescipub.com/jmss.toc)

SWITCHING-ALGEBRAIC ANALYSIS OF RELATIONAL DATABASES

[1]Ali Muhammad Ali Rushdi and [2]Omar Mohammed Ba-Rukab

[1]Department of Electrical and Computer Engineering, Faculty of Engineering,
King Abdulaziz University, P. O. Box 80204, Jeddah, 21589, Saudi Arabia
[2]Department of Information Technology, Faculty of Computing and Information Technology-Rabigh,
King Abdulaziz University, P.O. Box 344, Rabigh, 21911, Saudi Arabia

Received 2014-01-19; Revised 2014-02-01; Accepted 2014-04-18

ABSTRACT

There is an established equivalence between relational database Functional Dependencies (FDs) and a fragment of switching algebra that is built typically of Horn clauses. This equivalence pertains to both concepts and procedures of the FD relational database domain and the switching algebraic domain. This study is an exposition of the use of switching-algebraic tools in solving problems typically encountered in the analysis and design of relational databases. The switching-algebraic tools utilized include purely-algebraic techniques, purely-visual techniques employing the Karnaugh map and intermediary techniques employing the variable-entered Karnaugh map. The problems handled include; (a) the derivation of the closure of a Dependency Set (DS), (b) the derivation of the closure of a set of attributes, (c) the determination of all candidate keys and (d) the derivation of irredundant dependency sets equivalent to a given DS and consequently the determination of the minimal cover of such a set. A relatively large example illustrates the switching-algebraic approach and demonstrates its pedagogical and computational merits over the traditional approach.

Keywords: Switching Algebra, Relational Databases, Rules of Inference, Algebraic and Map Methods, Closure of a Set, Variable-Entered Karnaugh Map, Functional Dependency, Minimal Cover, Candidate Keys

1. INTRODUCTION

It has been known for decades that there is an equivalence between relational database Functional Dependencies (FDs) and a fragment of propositional logic (Delobel and Casey, 1973; Fagin, 1977; Sagiv *et al.*, 1981; Fagin, 1982; Russomano and Bonnell, 1999; Zhang, 2009a; 2009b; 2010; YiShun and ChunHua 2012). Typically, that fragment covers what is known as Horn clauses in switching theory (two-valued Boolean algebra). **Table 1** identifies related concepts and procedures in the domains of relational databases and propositional logic or switching theory. The traditional approach for the analysis and design of relational databases is based on the heuristic application of rules of inference. We demonstrate herein that such analysis and design can be facilitated, made more efficient, rendered algorithmic in nature, extended to problems of larger sizes and equipped with insightful visualization through the utilization of well established and readily-available tools of switching algebra.

2. MATERIALS AND METHODS

This study utilizes switching-algebraic tools in solving problems typically encountered in the analysis and design of relational databases.

Corresponding Author: Ali Muhammad Ali Rushdi, Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz, University, P. O. Box 80204, Jeddah 21589, Saudi Arabia

Table 1. Related concepts and procedures in the domains of relational databases and propositional logic (switching theory)

	Relational databases	Propositional logic (Switching theory)
Concepts	Functional Dependency (FD)	Horn clause (A subset of switching functions)
	Closure of a set	Complete Sum (CS), Blake Canonical Form (BCF)
	Irredundant Dependency Set (IDS)	Irredundant Disjunctive Form (IDF)
	Minimal cover	Minimal sum
Procedures	Application of armstrong's rules of inference	Iterative-consensus procedure
	Heuristic procedure for closure derivation	Algorithmic derivation for the complete sum

The switching algebraic tools utilized include purely-algebraic techniques (Muroga, 1979; Brown, 1990; Gregg, 1998), purely-visual techniques employing the Karnaugh map (Muroga, 1979; Brown, 1990; Gregg, 1998; Rushdi, 1997) and intermediary or mixed techniques employing the Variable-Entered Karnaugh Map (VEKM) (Muroga, 1979; Rushdi, 1985; 1987; 1997; 2001a; 2001b; Rushdi and Al-Yahya, 2000; 2001a; 2001b; 2002; Rushdi and Albarakati, 2012; 2014; Rushdi and Amashah, 2011). The problems handled include; (a) the derivation of the closure of a Dependency Set (DS), (b) the derivation of the closure of a set of attributes, (c) the determination of all candidate keys and (d) the derivation of all irredundant dependency sets equivalent to a given DS and consequently the determination of the minimal cover of such a set. A single relatively large example is used to apply the switching-algebraic approach to each of these problems and to demonstrate its pedagogical and computational merits over the traditional approach.

3. RESULTS

3.1. The Derivation of the Closure of a Dependency Set (DS)

Starting with a set of functional dependencies $A_i \rightarrow C_i, 1 \le i \le n$, that constitutes a set S, we view these dependencies as propositional implications that we denote by the same symbols ($A_i \rightarrow C_i$, $1 \le i \le$), taking liberty to allow a little abuse of notation. According to the Modern Syllogistic Method (MSM) (Blake, 1938; Brown, 1990; Rushdi and Al-Shehri, 2002; Rushdi and Ba-Rukab, 2007; 2008a; 2008b; 2009; Rushdi and Baz, 2007), these implications reduce to the switching Equation 1:

$$A_i \bar{C}_i = 0, \quad 1 \le i \le n \tag{1}$$

Which subsequently reduce to the single Equation 2:

$$g = \bigvee_{i=1}^{n} A_i \bar{C}_i = 0 \tag{2}$$

Which can be used to produce the equivalent result Equation 3 and 4:

$$CS(g) = 0 \tag{3}$$

where, CS(g) stands for the complete sum of the function g. We derive CS (g) via any appropriate algorithm such as Tison algorithm (Tison, 1967; Cutler *et al*., 1979; Brown, 1990; Rushdi and Al-Yahya, 2001; Rushdi and Albarakati, 2014), or the algorithm of VEKM folding (Rushdi and Al-Yahya, 2001). Each of the prime implicants (prime consequents) in CS(g) is interpreted as an equation of the form (1) and hence converted to a propositional implication or equivalently to a functional dependency that is a member of S^+.

Example 1

Consider the set of FD's described by (Dates, 2004):

$$S = \left\{ \begin{matrix} AB \rightarrow C,\ C \rightarrow A,\ BC \rightarrow D, \\ ACD \rightarrow B,\ BE \rightarrow C,\ CE \rightarrow \\ FA,\ CF \rightarrow BD,\ D \rightarrow EF \end{matrix} \right\} \tag{4}$$

We want to derive the closure S^+ of the set S. We first transform the set S from the relational domain to the Boolean domain as Equation 5:

$$\begin{matrix} g = AB\bar{C} \vee C\bar{A} \vee BC\bar{D} \vee \\ ACD\bar{B} \vee BE\bar{C} \vee CE\bar{F} \vee CE\bar{A} \\ \vee CF\bar{B} \vee CF\bar{D} \vee D\bar{E} \vee D\bar{F} \end{matrix} \tag{5}$$

Now, we derive CS(g) via the improved Tison algorithm (Rushdi and Al-Yahya, 2001) as detailed in **Fig. 1**. The final complete sum (after six iterations of consensi generation with respect to each biform variable, followed by absorptions of subsuming products) is Equation 6:

$$\begin{matrix} CS(g) = AB\bar{C} \vee AB\bar{D} \vee AB\bar{E} \vee AB\bar{F} \vee BC\bar{D} \vee BC\bar{E} \\ \vee BC\bar{F} \vee BD\bar{A} \vee BD\bar{C} \vee BE\bar{A} \vee BE\bar{A} \vee BE\bar{C} \vee BE\bar{D} \\ \vee BE\bar{F} \vee DE \vee D\bar{F} \vee C\bar{A} \vee CD\bar{B} \vee CE\bar{B} \vee CF\bar{B} \vee CE\bar{D} \\ \vee CE\bar{F} \vee CF\bar{D} \vee CF\bar{E} = 0 \end{matrix} \tag{6}$$

Note that CS(g) consists of 23 prime implicants, each of which has a single complemented literal.

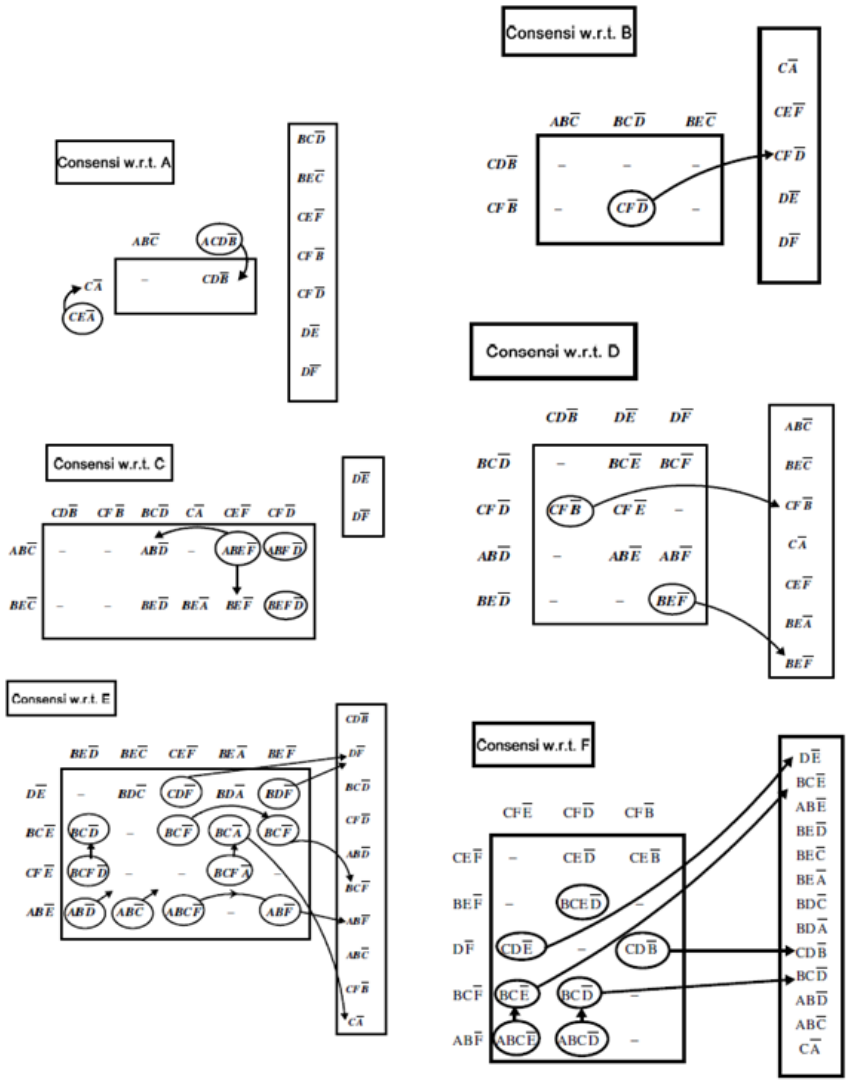


Fig. 1. Derivation of CS(g) via the improved Tison algorithm with a single passage through the biform variables A, B, C, D, E and F, respectively

Now, we make an inverse transform from the Boolean domain to the relational domain. The closure of the set S in (4) has 23 FDs and is given by Equation 7:

$$S^{+}=\left\{\begin{array}{l} AB\rightarrow C, AB\rightarrow D, AB\rightarrow E, AB\rightarrow F,\\ BC\rightarrow D, BC\rightarrow E, BC\rightarrow F, BD\rightarrow A,\\ BD\rightarrow C, BE\rightarrow A, BE\rightarrow C, BE\rightarrow A,\\ BE\rightarrow C, BE\rightarrow A, BE\rightarrow C, BE\rightarrow D, BE\rightarrow F,\\ D\rightarrow E, D\rightarrow F, C\rightarrow A, CD\rightarrow B, CE\rightarrow B,\\ CF\rightarrow B, CE\rightarrow D, CE\rightarrow F, CF\rightarrow D, CF\rightarrow E \end{array}\right\} \quad (7)$$

3.2. The Derivation of the Closure of a Set of Attributes

Given a set Z of attributes of relational variable (relvar) R and a set S of FDs that hold for R, the closure Z+ of Z under S is the set of all attributes A of R such that the FD {Z→A} is a member of S+ (i.e., such that the FD {Z → A) is implied by the FDs in S). Thanks to reflexivity or self-determination {A→A}, this definition agrees with the requirement that Z is a subset of Z^{+}. If we express the given set of FDs as a single equation (g = 0) and then convert its LHS g into the complete-sum form

CS(g), as we did earlier, then we can deduce the closure Z^+ of as follows. Initially, any attribute in Z is added to Z^+. Subsequently, a single pass is made through the set of prime implicants P_i whose disjunction constitutes CS(g). If the uncomplemented variables in P_i represent a subset of Z, then the attribute represented by the single complemented variable in P_i is added to Z^+ (if it is not already there). If the closure Z^+ of a set of attributes Z equals the total set of attributes of R, then Z constitutes a superkey. If such a set Z is irreducible, then this superkey is a candidate key.

Example 1 (Revisited)

Consider the relational variable (*relvar*) R with attributes A, B, C, D, E and F and the dependency set described by (4). Equation (6) can be utilized to deduce the closure of any subset of the set of attributes Z = {A, B, C, D, E, F}. We can ascertain that $\{C\}^+$ is given by Equation 8:

$$\{C\}^+ = \{A, C\} \neq Z \tag{8}$$

Thanks to the property of self-determination $\{C \rightarrow C\}$ and the existence of prime implicant $C\bar{A}$ in (6), which is equivalent to the dependency $\{C \rightarrow A\}$ Similarly Equation 9:

$$\{D\}^+ = \{D, E, F\} \neq Z \tag{9}$$

Since the existence of prime implicants $D\bar{E}$ and $D\bar{F}$ in (6) is equivalent to the dependencies $\{D \rightarrow E, D \rightarrow F\}$. Together with $\{D \rightarrow D\}$, these dependencies result in (9). Likewise, the closure $\{B, C\}^+$ equals {B, C, D, E, F} thanks to the appearance of prime implicants $BC\bar{D}$, $BC\bar{E}$ and $BC\bar{F}$ in (6). The result that Equation 10:

$$\{C, D\}^+ = \{A, B, C, D, E, F\} = Z \tag{10}$$

Can be accounted for similarly by noting that $\{C, D\}^+$ is a superset for the union of $\{C\}^+$ and $\{D\}^+$ in (8) and (9) and including the effect of the prime implicant $CD\bar{B}\{CD \rightarrow B\}$ in (6). This means that CD is an irreducible superkey (i.e., a candidate key) for the FD's in (4).

3.3. The Determination of All Candidate Keys

Since a natural (albeit typically redundant) superkey is the conjunction of all attributes X_i, we might supplement the set of FD's by an extra dependency of the form Equation 11:

$$\left(\wedge_{i=1}^{n} X_i\right) \rightarrow K \tag{11}$$

where, K is an additional attribute that stands for "Key". Now, the function g in (5) is replaced by a function f given by Equation 12:

$$f = g \vee \left(\wedge_{i=1}^{n} X_i\right)\bar{K} \tag{12}$$

The prime implicants in CS(f) that contain the literal $\bar{K}$, i.e., that are of the form $(\wedge_{j \in J} X_j)\bar{K}$ correspond to the implication Equation 13:

$$\left(\wedge_{j \in J} X_J\right) \rightarrow K \tag{13}$$

And hence the set J of attributes are superkeys. In fact, they are candidate keys since they are irredundant, because they correspond to prime implicants. Therefore, one can obtain all the candidate keys by obtaining the complete sum of the function f defined by (12). A candidate key corresponds to the uncomplemented literals in any prime implicant that contains the complemented literal $\bar{K}$. This scheme agrees with the procedure in Zhang (2009a).

Now, since K is a mono-form variable, it plays no role in the consensus generation used in complete-sum derivation. Hence, we can dispense with it altogether and rewrite (12) with $\bar{K}$ deleted, i.e., in the form Equation 14:

$$f = g \vee \left(\wedge_{i=1}^{n} X_i\right) \tag{14}$$

In this new scheme, the final result for CS (f) is Equation 15:

$$CS(f) = CS(g) \vee \vee_J \left(\wedge_{j \in J} X_j\right) \tag{15}$$

where, the prime implicants $(\wedge_{j \in J} X_j)$ of solely uncomplemented literals are the candidate keys. This scheme agrees with the procedure in (Russomano and Bonnell, 1999). It could be enhanced if CS (g) is already available, for then we redefine f in (14) by the equivalent formula:

$$f = CS(g) \vee \left(\wedge_{i=1}^{n} X_i\right) \tag{16}$$

Equation (16) suggests a method employing Tison algorithm incrementally (Kean and Tsiknis, 1990) for deriving the complete sum. In fact, we can restrict our attention to the consensi generated between terms without complemented literals (initially the single term $\left(\wedge_{i=1}^{n} X_i\right)$) and various terms of CS(g) w.r.t. bi-form variables traversed one by one. The final set of these consensi is the set of all candidate keys, i.e., $V_J \left(\wedge_{j\in J} X_j\right)$.

To implement the aforementioned method, let us use G_k to denote terms in G_k containing variable number k complemented and U_k to denote the disjunction of uncomplemented implicants of f at step k. At each step of the incremental Tison algorithm, U_k is updated via Equation 17:

$$U_{k+1} = \text{ABS}\left(U_k \vee \text{Consensi}\left(U_k, G_k\right)\right) \quad (17)$$

Here, ABS (F) is an absorptive formula for the sum-of-products formula F which is a formula obtained from F by successive deletion or absorption of terms subsuming other terms in F (Brown, 1990; Rushdi and Al-Yahya, 2001). The expression U_k can be interpreted as a disjunction of superkeys. Its initial value is $\left(\wedge_{i=1}^{n} X_i\right)$ and its final value is the disjunction of all prime implicants of CS(f) with solely un-complemented literals, which are the candidate keys.

Example 1 (Revisited)

We want to determine all candidate keys for the set of FD's given in (4). We construct a function f (A, B, C, D, E, F) which equals the function g in (5), (or its complete sum in (6)) disjunctioned with a term (A B C D E F) that equals the conjunction or ANDing of all pertinent variables, all un-complemented. **Table 2** shows an implementation of the incremental Tison algorithm via (17). The disjunction of the prime implicants of CS(f) with un-complemented literals is Equation 18:

$$CD \vee CE \vee CF \vee AB \vee BC \vee BD \vee BE \quad (18)$$

This result means that there are seven candidate keys for the given set of FDs, namely, CD, CE CF, AB, BC BD and BE. It is straightforward to verify that each of these keys is indeed a candidate key by inspecting CS(g) in (6). For example, BC is a superkey due to the existence of prime implicants $\bar{A}C, BC\bar{D}, BC\bar{E}$ and $BC\bar{F}$. Further, it is a candidate key since it is irreducible.

3.4. The Derivation of Irredundant Dependency Sets Equivalent to a Given DS

An Irredundant Disjunctive Form (IDF) for a switching function g is a disjunction of prime implicants such that removal of any of the prime implicants makes the remaining formula not express the original g (Muroga, 1979). This means that an IDF for g is a minimal sub-formula of CS(g) that covers g (Rushdi and Al-Yahya, 2002). The corresponding entity in the relational domain, namely, the Irredundant Dependency Set (IDS), is defined similarly (see, e.g., (Dates, 2004), with an additional requirement that a functional dependency of multiple consequents be decomposed into several FDs of single consequents. For example, the FD A $\rightarrow$ BCD must be replaced by the FDs A $\rightarrow$ B, A$\rightarrow$ C and A $\rightarrow$ D, that map into the terms or products $A\bar{B}$, $A\bar{C}$ and $A\bar{D}$, which can fit into a disjunctive form. An Irredundant disjunctive form (and correspondingly, an irredundant dependency set) is not necessarily unique.

There are many algebraic, tabular or mapping methods for obtaining all the IDFs of a switching function (Muroga, 1979). Most of these methods are algorithms that use the complete sum as a starting point when it is available, or more generally act in a 2-step fashion by finding the complete sum first before proceeding to derive the IDFs. There are other heuristic methods for obtaining the IDFs directly, such as those employing the conventional Karnaugh map (Muroga, 1979; Brown, 1990; Gregg, 1998; Rushdi, 1997) or the Variable-Entered Karnaugh Map (VEKM) (Muroga, 1979; Rushdi, 1985; 1987; 1997; 2001a; 2001b; Rushdi and Al-Yahya, 2000; 2001a; 2001b; 2002; Rushdi and Albarakati, 2012; 2014; Rushdi and Amashah, 2011). These heuristic methods are not guaranteed to find all the IDFs, but they typically find most of them, including the best or minimal among them. We now apply a VEKM minimization procedure to our running example.

Example 1 (Revisited)

For comparison purposes, we present herein the traditional heuristic for obtaining the IDSs that are equivalent to the DS in (4). The first step is to rewrite the given set in (4) such that every FD has a singleton consequent or right side. We denote the resulting set as set I.

Table 2. Derivation of CS (f) in (16) via the incremental tison method

	G_i	U_i	Consensi (U_i, G_i)
A	$BD\bar{A} \vee BE\bar{A} \vee C\bar{A}$	ABCDEF	BCDEF∨BCDEF∨BCDEF
B	$CD\bar{B} \vee CE\bar{B} \vee CF\bar{B}$	BCDEF	CDEF∨CDEF∨CDEF
C	$AB\bar{C} \vee BD\bar{C} \vee BE\bar{C}$	CDEF	ABDEF∨BDEF∨BDEF
D	$AB\bar{D} \vee BC\bar{D} \vee BE\bar{D} \vee CE\bar{D} \vee CF\bar{D}$	CDEF∨BDEF	ABCEF∨ABEF∨BCEF∨BCEF∨ BCEF∨BEF∨CEF∨BCEF ∨CEF∨BCEF
E	$AB\bar{E} \vee BC\bar{E} \vee D\bar{E} \vee CF\bar{E}$	CEF∨BEF	ABCF∨ABF∨BCF∨BCF∨ CDF∨BDF∨CF∨BCF
F	$AB\bar{F} \vee BC\bar{F} \vee BE\bar{F} \vee D\bar{F} \vee CE\bar{F}$	BEF∨ABF∨BDF∨CF	ABE∨AB∨ABD∨ABC ∨BCE∨ABC∨BCD∨BC∨ ∨BE∨ABE∨BDE∨BCE∨ BDE∨ABD∨BD∨CD ∨BCE∨ABCE∨BCDE∨CE
		AB∨BC∨BE∨BD∨CD∨CE∨CF	

$$\left.\begin{array}{l} 1.\ AB \rightarrow C \\ 2.\ C \rightarrow A \\ 3.\ BC \rightarrow D \\ 4.\ ACD \rightarrow B \\ 5.\ BE \rightarrow C \\ 6.\ CE \rightarrow A \\ 7.\ CE \rightarrow F \\ 8.\ CF \rightarrow B \\ 9.\ CF \rightarrow D \\ 10.\ D \rightarrow E \\ 11.\ D \rightarrow F \end{array}\right\} \text{Set I}$$

Henceforth, we keep the number of each Functional Dependency (FD) as assigned to it in set I. Now, FD2 implies FD6, so we can drop FD6. Functional Dependency FD8 implies {CF→BC} (by augmentation), which with FD3 implies {CF →D} (by transitivity), so we can drop FD9. Functional Dependency 8 implies {ACF→AB} (by augmentation) and FD11 implies {ACD→ACF} (by augmentation) and so {ACD→B} (by decomposition), so we can drop FD4. No further reductions are possible and so we are left with an irreducible set, that we call set II:

$$\left.\begin{array}{l} 1.\ AB \rightarrow C \\ 2.\ C \rightarrow A \\ 3.\ BC \rightarrow D \\ 5.\ BE \rightarrow C \\ 7.\ CE \rightarrow F \\ 8.\ CF \rightarrow B \\ 10.\ D \rightarrow E \\ 11.\ D \rightarrow F \end{array}\right\} \text{Set II}$$

Alternatively, starting from the original set (set I), FD2 implies {CD→ACD} (by composition), which with FD4 implies {CD→B} (by transitivity), so we can replace FD4 by {CD→B}, which we call FD12. Functional dependency 2 implies FD6, so we can drop FD6 (as before). Functional dependencies 2 and 9 imply {CF → AD} (by composition), which implies {CF→ACD} (by augmentation), which with (the original) FD4 implies FD8 {CF → B} (by transitivity), so we can drop FD8. No further reductions are possible and so we are left with an irreducible set, that we call set III. Set III is similar to set II with FD8 in set II replaced by FD9 and FD12 in set III.

Alternatively, starting again from set II, FD1 implies {AB →BC} (by augmentation), which with FD3 implies {AB→D} (by transitivity). Now if we add {AB→D} to set II it ceases to be irredundant anymore. Functional dependency 2 implies {BC→AB} (by augmentation), which with the new FD {AB→D} implies {BC→D} (by transitivity), so we can drop FD3 {BC→D} and we obtain a new IDF (set IV) in which we denote the new FD {AB→D} as FD3a since it replaces FD3:

$$\left.\begin{array}{l} 1.\ AB \rightarrow C \\ 2.\ C \rightarrow A \\ 3.\ BC \rightarrow D \\ 5.\ BE \rightarrow C \\ 7.\ CE \rightarrow F \\ 9.\ CF \rightarrow D \\ 10.\ D \rightarrow E \\ 11.\ D \rightarrow F \\ 12.\ CD \rightarrow B \end{array}\right\} \text{Set III}$$

$$\left.\begin{array}{l} 1.AB\rightarrow C \\ 2.C\rightarrow A \\ 3a.AB\rightarrow D \\ 5.BE\rightarrow C \\ 7.CE\rightarrow F \\ 8.CF\rightarrow B \\ 10.D\rightarrow F \\ 11.D\rightarrow F \end{array}\right\} \text{Set IV}$$

Alternatively, we can go back to the first IDS (Set II) where FD2 and FD5 imply {BE→A} (by transitivity). We add {BE→A} to the IDF so that it ceases to be irredundant. Now {BE→A} implies {BE→AB} (by augmentation), which with FD1 implies {BE→C} (by transitivity), so we drop FD5, label its replacement {BE→C} as FD5a and we end up with an IDS, that we call set V.

Alternatively, FD7 in set II implies {CE→CF} (by augmentation), which with FD8 implies {CE→B} (by transitivity). When we add {CE→B} to the original IDS it is no longer irredundant. But now {CE→B} implies {CE→BC} (by augmentation) which with FD3 implies {CE→BC} which by transitivity with FD11 implies {CE→F}, so we can drop FD7 and denote its replacement {CE→B} as FD7a and hence obtain an IDS, that we call set VI:

$$\left.\begin{array}{l} 1.AB\rightarrow C \\ 2.C\rightarrow A \\ 3.BC\rightarrow D \\ 5a.BE\rightarrow A \\ 7.CE\rightarrow F \\ 8.CF\rightarrow B \\ 10.D\rightarrow E \\ 11.D\rightarrow F \end{array}\right\} \text{Set V}$$

$$\left.\begin{array}{l} 1.AB\rightarrow C \\ 2.C\rightarrow A \\ 3.BC\rightarrow D \\ 5\ BE\rightarrow C \\ 7a.CE\rightarrow B \\ 8.\ CF\rightarrow B \\ 10.D\rightarrow E \\ 11.D\rightarrow F \end{array}\right\} \text{Set VI}$$

So far, we have identified 5 irredundant sets (Sets II to Set VI). However, the total number of irredundant sets is at least 16. As **Fig. 2** indicates, FDs 1, 2, 10 and 11 are included in every identified irredundant set. These are supplemented by (a) Either FD3 or FD3a, (b) Either FD5 or FD5a, (c) Either FD7 or FD7a and (d) FD8 or a combination of FD9 and FD12. Referring to the original set (Set I), we note that both FD4 {ACD→B} and FD6 {CE→A} have disappeared entirely being non-prime implicants (since they subsume the prime implicants $CD\bar{B}(CD\rightarrow B)$ and $\bar{A}C(C\rightarrow A)$, respectively).

The result obtained in **Fig. 2** can be restated to express the corresponding irredundant disjunctive forms as:

$$\begin{aligned} IDF = {} & P_1 \vee P_2 \vee P_3 \{P_{3a}\} \vee P_5 \{P_{5a}\} \vee p_7 \{p_{7a}\} \\ & \vee P_8 \{P_9 \vee P_{12}\} \vee P_{10} \vee P_{11,} \end{aligned} \tag{19}$$

Out of the 23 prime implicants of g that appear in CS(g) in (6), only 12 are needed in the IDFs in (19), namely:

$$\begin{aligned} & P_1 = AB\bar{C}, P_2 = C\bar{A}, P_3 = BC\bar{D}, P_{3\alpha} = AB\bar{D} \\ & P_5 = BE\bar{C}, P_{5\alpha} = BE\bar{A}, P_7 = CE\bar{F}, P_{7\alpha} = CE\bar{B}, \\ & P_8 = CF\bar{B}, P_9 = CF\bar{D}, P_{12} = CD\bar{B}, P_{10} = \\ & D\bar{E} \text{ and } P_{11} = D\bar{F} \end{aligned}$$

The notation P_3 {$P_{3\alpha}$}with curly brackets means that either prime implicant P_3 or prime implicant $P_{3\alpha}$ is included in the IDF. Since Equation (19) has 4 such binary alternatives, it represents 2^4 = 16 IDFs. Out of these, there are 8 minimal covers (the ones employing P_8 rather than its alternative (P_9 V P_{12}). However, due to the non-algorithmic nature of the current heuristic, we are not fully sure that we have exhausted all IDFs and hence all minimal covers.

The result of (19) is now recovered via VEKM minimization in the switching domain. We use Boole-Shannon expansion (Brown, 1990; Rushdi and Al-Yahya, 2001) to obtain the VEKM representation of the function g in (5) as shown in **Fig. 3**. An almost minimal s-o-p expression for the function g is given (Rushdi, 1987; Rushdi and Al-Yahya, 2000; 2001) by Equation 20:

$$g = V_r\ P_r\ Co(P_r) \tag{20}$$

where, P_r is a prime implicant of one or more of the subfunctions of the function g, i.e., it is a product that appears in at least one VEKM cell. Each P_r is ANDed with its minimal s-o-p contribution to g, namely Co (P_r). This contribution can be represented by a CKM directly deducible from the original VEKM according to heuristic rules stated in (Rushdi, 1987; Rushdi and Al-Yahya, 2000; 2001b).

Fig. 2. The functional dependencies included in various identified irredundant sets

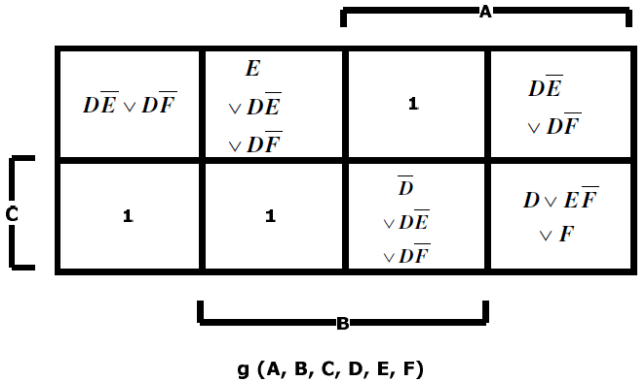


Fig. 3. One VEKM representation of the function g in (2)

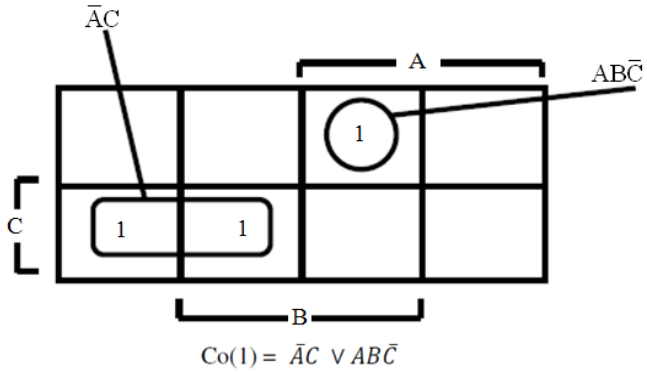


Fig. 4. Contribution of entered product 1 adds $P_1 = AB\bar{C}, P_2 = C\bar{A}$

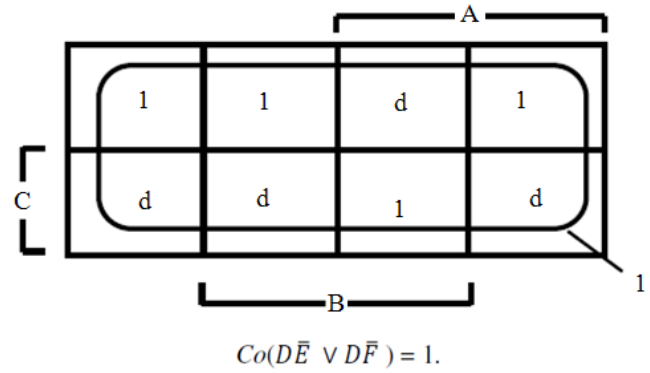
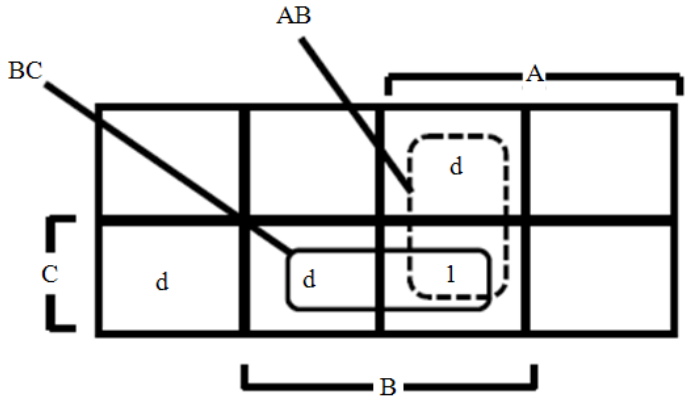
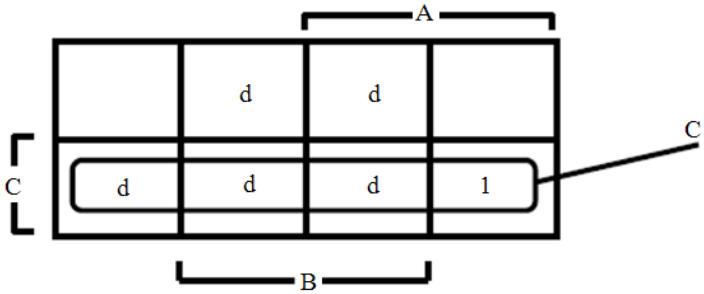


Fig. 5. Contribution of entered products $D\bar{E} \vee D\bar{F}$ adds $P_{10} = D\bar{E}$ and $P_{11} = D\bar{F}$



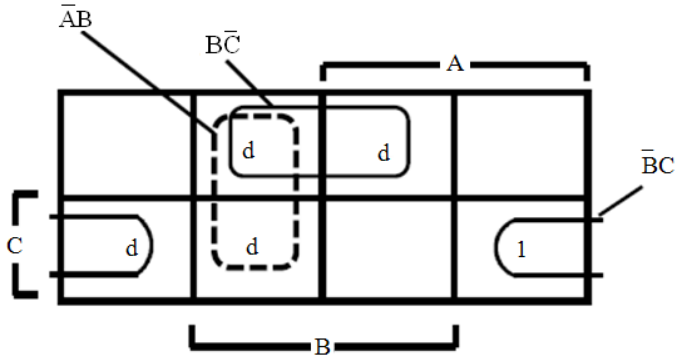
$Co(\bar{D}) = AB\ OR\ BC.$

Fig. 6. Contribution of entered product $\bar{D}$ adds $P_3 = BC\bar{D}$ or $P_{3\alpha} = AB\bar{D}$



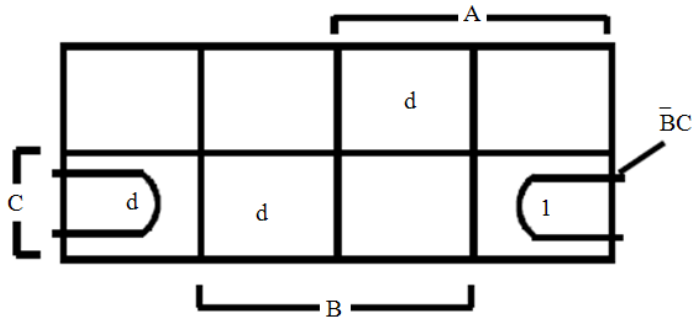
$Co(E\bar{F}) = C\ if\ A\bar{B}C$ is not covered in Co(E).

Fig. 7. Contribution of entered product $E\bar{F}$ adds $P_7 = CE\bar{F}$ if P_{7a} is not added



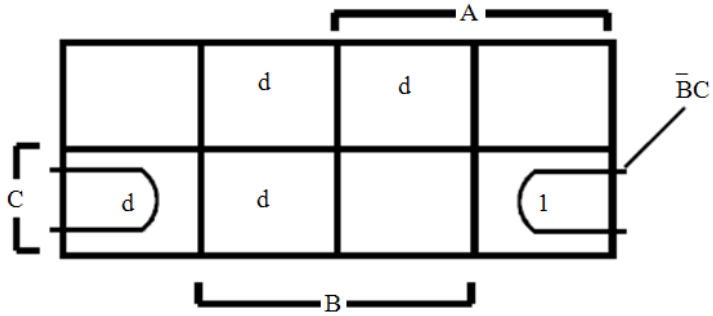
$Co(\mathrm{E}) = (B\bar{C}\ OR\ \bar{A}B) \vee (\bar{B}C\ \ if\ \ A\bar{B}C$ is not covered in $Co(\mathrm{E}\bar{F})$).

Fig. 8. Contribution of entered product E adds $P_5 = BE\bar{C}$ or $P_{5a} = BE\bar{A}$ and might add $P_7 = CE\bar{F}$ in place of P_{7a}



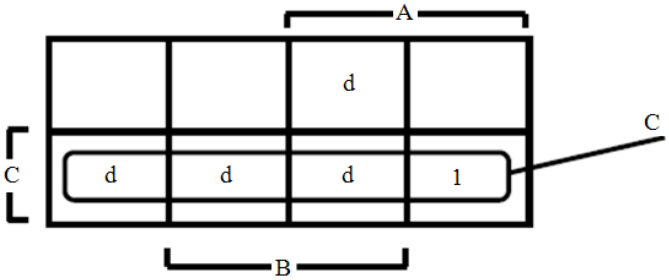
$Co(F) = \bar{B}C$
OR $Co(F) = 0$ *if* $A\bar{B}C$ is covered in both
$Co(D)$ and $Co(\bar{D}F)$.

Fig. 9. Contribution of entered product F might add $P_8 = CF\bar{B}$ if P_{12} and P_9 are not jointly added



$Co(D) = \bar{B}C$
OR $Co(D) = 0$ *if* Cell $A\bar{B}C$ is covered in
$Co(E)$ or in $Co(F)$.

Fig. 10. Contribution of entered product D adds $P_{12} = CD\bar{B}$ to join P_9 in replacing P_8



$Co(\bar{D}F) = C$ *if* $A\bar{B}C$ is not covered in $Co(F)$.

Fig. 11. Contribution of product $\bar{D}F$ adds $P_9 = CF\bar{D}$ to join P_{12} in replacing P_8

Note that Co (P_r) is a function of the map variables of the VEKM only while P_r itself is a function of the entered variables of the VEKM only.

Figures 4-11 detail how the various contributions of the entered products in **Fig. 3** are obtained and how the formula in (19) is recovered via the VEKM heuristics.

4. DISCUSSION

This study presented switching-algebraic analysis of four important and related problems of relational databases, namely, (a) the problem of deriving the closure of a set of functional dependencies, (b) the problem of deriving the closure for a set of attributes, (c) the problem of deriving all candidate keys and (d) the problem of deriving irredundant dependency sets equivalent to a set of functional dependencies and the determination of the minimal cover of such a set. Three of these four problems were shown to require the derivation of the complete sum (blake canonical form) of certain switching functions obtained via an appropriate transformation of the relational functional dependencies. These three problems were solved by two updated versions of Tison algorithm. The fourth problem could have been solved also by the derivation of a complete sum and the construction of a presence function (Petrick function) (Muroga, 1979; Rushdi and Al-Yahya, 2002), but it was solved herein via a heuristic utilizing the Variable-Entered Karnaugh Map (VEKM). All four problems were presented in ample detail and their methods of solution were demonstrated via the same relatively large example.

5. CONCLUSION

This study is a serious attempt to transform relational database concepts to the switching-algebraic domain and hence to utilize switching-algebraic procedures and concepts in relational databases. This attempt stresses the pedagogical advantages gained when one departs from the traditional database approach and utilizes pictorial tools of switching theory and digital design, such as the Variable-Entered Karnaugh Map (VEKM). The traditional database approach, adopted by almost all textbooks on database design is based on the heuristic application of axioms and lemmas for the manipulation of functional dependencies. This study replaces the traditional heuristics in the relational domain by more powerful and insightful heuristics and algorithms in the switching domain.

An interesting topic for further research stems from the fact that the function dealt with in deriving the closure for dependency sets is a disjunction of terms derived from particular Horn clauses. Each of these terms is a product (ANDing) of a single complemented literal with some other un-complemented literals. This feature should be studied with the hope of simplifying the algorithm that extracts all the prime implicants. A pertinent question in this respect is whether a linear representation of a switching function (e.g., Rushdi and Ghaleb, 2013; Rushdi and Alsogati, 2013) could provide anyadvantage over the current sum-of-products representation.

Another promising direction of potentially fruitful research is to utilize the switching-domain tools of this study in the implementation of the coneptual database designing model advocated by Hegazi (2014).

6. ACKNOWLEDGEMENT

This article was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah. The authors, therefore, acknowledge with thanks DSR technical and financial support.

7. REFERENCES

Blake, A., 1938. Canonical Expressions in Boolean Algebra. 1st Edn., University of Chicago, Chicago, pp: 60.

Brown, F.M., 1990. Boolean Reasoning: The Logic of Boolean Equations. 1st Edn., Springer, Boston, ISBN-10: 0792391217, pp: 273.

Cutler, R.B., K. Kinoshita and S. Muroga, 1979. Exposition of tison's method to derive all prime implicants and all irredundant disjunctive forms for a given switching function. Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana.

Dates, C.J., 2004. An Introduction to Database Systems. 8th Edn, Pearson Education/Addison-Wesley, Boston, ISBN-10: 0321189566, pp: 1005.

Delobel, C. and R.G. Casey, 1973. Decomposition of a data base and the theory of Boolean switching functions. IBM J. Res. Dev., 17: 374-386.

Fagin, R., 1977. Functional dependencies in a relational database and propositional logic. IBM J. Res. Dev., 21: 534-544. DOI: 10.1147/rd.216.0534

Fagin, R., 1982. Horn clauses and database dependencies. J. Assoc. Comput. Machinery, 29: 952-985. DOI: 10.1145/322344.322347

Gregg, J.R., 1998. Ones and Zeroes: Digital Circuits and the Logic of Sets. 1st Edn., Wiley, New York, NY, USA.

Hegazi, M.O.A., 2014. A coneptual foundation and integration database designing model. J. Comput. Sci., 10: 376-381. DOI: 10.3844/jcssp.2014.376.381

Kean, A. and G. Tsiknis, 1990. An incremental method for generating prime implicants/implicates. J. Symbolic Comput., 9: 185-206. DOI: 10.1016/S0747-7171(08)80029-6

Muroga, S., 1979. Logic Design and Switching Theory. 1st Edn., Wiley, New York, ISBN-10: 0471044180, pp: 617.

Rushdi, A.M., 1985. Map derivation of the minimal sum of a switching function from that of its complement. Microelectron. Reliability, 25: 1055- 1065. DOI: 10.1016/0026-2714(85)90481-0

Rushdi, A.M., 1987. Improved variable-entered Karnaugh map procedures. Comput. Electr. Eng., 13: 41-52. DOI: 10.1016/0045-7906(87)90021-8

Rushdi, A.M., 1997. Karnaugh Map, In: Encyclopaedia of Mathematics. Hazewinkel, S.M. (Edr), Kluwer Academic Publishers, Boston, pp: 327-328.

Rushdi, A.M., 2001a. Prime-implicant extraction with the aid of the variable-entered Karnaugh map. J. Sci. Med. Eng., 13: 53-74.

Rushdi, A.M., 2001b. Using variable-entered Karnaugh maps to solve Boolean equations. Int. J. Comput. Math., 78: 23-38. DOI: 10.1080/00207160108805094

Rushdi, A.M. and A.S. Al-Shehri, 2002. Logical reasoning and its supporting role in the service of security and justice. J. Security Stud., 11: 115-153.

Rushdi, A.M. and H.A. Al-Yahya, 2000. A Boolean minimization procedure using the variable-entered Karnaugh map and the generalized consensus concept. Int. J. Electron., 87: 769-794. DOI: 10.1080/00207210050028724

Rushdi, A.M. and H.A. Al-Yahya, 2001a. Derivation of the complete sum of a switching function with the aid of the variable entered karnaugh map. King Saud University J. Eng. Sci., 13: 239-269.

Rushdi, A.M. and H.A. Al-Yahya, 2001b. Further improved variable-entered Karnaugh map procedures for obtaining the irredundant forms of an incompletely-specified switching function. J. King Abdulaziz Univ. Eng. Sci., 13: 111-152. DOI: 10.4197/Eng.13-1.6

Rushdi, A.M. and Al-Yahya, 2002. Variable-entered Karnaugh map procedures for obtaining the irredundant disjunctive forms of a switching function from its complete sum. J. King Saud Univ. Eng. Sci., 14: 13-27.

Rushdi, A.M.A. and H. M. Albarakati, 2012. The inverse problem for Boolean equations. J. Comput. Sci., 8: 2098-2105. DOI: 10.3844/jcssp.2012.2098.2105

Rushdi, A.M.A. and H.M. Albarakati, 2014. Construction of general subsumptive solutions of Boolean equations via complete-sum derivation. J. Math. Stat., 10: 155-168. DOI: 10.3844/jmssp.2014.155.168

Rushdi, A.M.A. and A.A. Alsogati, 2013. On reduced scalar equations for synchronous Boolean networks. J. Math. Stat., 9: 262-276. DOI: 10.3844/jmssp.2013.262.276

Rushdi, A.M. and M.H. Amashah, 2011. Using variable-entered Karnaugh maps to produce compact parametric solutions of Boolean equations. Int. J. Comput. Mathem., 88: 3136-3149. DOI: 10.1080/00207160.2011.594505

Rushdi, A.M. and O.M. Ba-Rukab, 2007. Some engineering applications of the modern syllogistic method. Proceedings of the 7th Saudi Engineering Conference, (SEC ‘07), Riyadh, Saudi Arabia, pp: 389-401.

Rushdi, A.M. and A.O. Baz, 2007. Computer-assisted resolution of engineering ethical dilemmas. Proceedings of the 7th Saudi Engineering Conference, (SEC ‘ 07), Riyadh, Saudi Arabia, pp: 409-418.

Rushdi, A.M. and O.M. Ba-Rukab, 2008a. Powerful features of the modern syllogistic method of propositional logic. J. Math. Stat., 4: 186-193. DOI: 10.3844/jmssp.2008.186.193

Rushdi, A.M. and O.M. Ba-Rukab, 2008b. The modern syllogistic method as a tool for engineering problem solving. J. Qassim Univ. Eng. Comput. Sci., 1: 57-70.

Rushdi, A.M. and O.M. Ba-Rukab, 2009. An exposition of the modern syllogistic method of propositional logic. Umm Al-Qura Univ. J. Eng. Architecture, 1: 17-49.

Rushdi, A.M.A. and F.A.M. Ghaleb, 2013. On self-inverse binary matrices over the binary Galois field. J. Math. Stat., 9: 238-248. DOI: 10.3844/jmssp.2013.238.248

Russomano, D.J. and R.D. Bonnell, 1999. A pedagogical approach to database design via Karnaugh maps. IEEE Trans. Educ., 42: 261-270. DOI: 10.1109/13.804530

Sagiv, Y., C. Delobel, D.S. Parker and R. Fagin, 1981. An equivalence between relational database dependencies and a fragment of propositional logic. J. Assoc. Comput. Machinery, 28: 435-453. DOI: 10.1145/322261.322263

Tison, P., 1967. Generalization of consensus theory and application to the minimization of Boolean functions. IEEE Trans. Electr. Comput., 16: 446-456. DOI: 10.1109/PGEC.1967.264648

Zhang, Y.S., 2009a. Determining all candidate keys based on Karnaugh map. Proceedings of the International Conference on Information Management, Innovation Management and Industrial Engineering, Dec. 26-27, IEEE Xplore Press, Xi'an, pp: 226-229. DOI: 10.1109/ICIII.2009.515

Zhang, Y.S., 2009b. Determining closure of sets of attributes based on Karnaugh map. Proceedings of the 2nd International Symposium on Knowledge Acquisition and Modeling, Nov. 30-Dec. 1, IEEE Xplore Press, Wuhan, pp: 190-193. DOI: 10.1109/KAM.2009.120

Zhang, Y.S., 2010. A visual explanation for computing minimal cover. Proceedings of the International Conference on Information Management, Innovation Management and Industrial Engineering, Nov. 26-28, IEEE Xplore Press, Kunming, pp: 125-128. DOI: 10.1109/ICIII.2010.352

YiShun, Z. and J. ChunHua, 2011. Logic Algebra Method for Solving Theoretic Problems of Relational Database. In: Computer, Informatics, Cybernetics and Applications, He, X., E. Hua, X. Liu and Y. Lin (Eds.)., Springer, Dordrecht, ISBN-10: 940071839X, pp: 1706-1706.