An Extended Synthesis Algorithm for Relational Database Schema Design

Ashraf Ali Waseem
Ph.D. Candidate
Hamdard University
Karachi, Pakistan
E-mail:
ashrafaliwaseem@yahoo.com

Syed Jamal Hussain
Assistant Professor
Computer Science Department,
University of Karachi
Karachi, Pakistan
E-mail:
drsjhussain@yahoo.com

Zubair Ahmed Shaikh
Professor
Computer Science Department,
FAST-NU University
Karachi, Pakistan
E-mail:
zubair.shaikh@nu.edu.pk

ABSTRACT
Relational model is the most essential database model for decades and is examined in various ways. A very important concept in the relational modeling is the concept of dependencies. Particularly a set of functional dependencies and its equivalent minimal cover are discussed here for synthesizing third normal form relations. A traditional synthesis algorithm have been used to reduce the set of functional dependencies, F, into an equivalent minimal cover. This cover is then used to develop a relational schema; however, a given set of functional dependencies may have more than one minimal covers. In turn, different minimal covers may give rise to different relational schema. Database designers had already been agreed upon the use of popular inference rules of Armstrong to obtain minimal covers from the set F of functional dependencies. Despite the frequent use of the traditional synthesis algorithm, there is still a lack of formality and simplicity support in that algorithm to produce a minimum cover, especially; the step(s) of the algorithm in which the set of functional dependencies of F are used to make left-irreducible remain informal. In this paper, an extension is proposed to the synthesis algorithm that aids in to produce an optimal minimal cover more systematically from the given set of functional dependencies. The optimal minimal cover obtained by this extension would be appropriate for synthesizing third normal form relational schema from the universal relation scheme.

Categories and Subject Descriptors
H.2.1 **[Database Management]**: Logical Design – *schema and subschema, normal forms, data models;* H.2.4 **[Database Management]**: Systems – *Relational databases*.

General Terms
Algorithms, Design.

Keywords
Armstrong's axioms, Redundant attributes, Functional dependencies, Minimal cover of FDs, Relational Schema, 3NF relations

1. INTRODUCTION
The theory of dependencies plays an integral role in the theory of relational databases. The dependencies provide the formal view of relationships among data in a relational database model. A database problem can be easily transformed into logical schema using these dependencies. By using the concept of functional, multi-valued and join dependencies the theory of normalization is developed. In an early paper on relational model, Codd defined certain normal forms for a relational schema [1]. Codd's initial normal forms were based on the notion of functional dependencies. It is basically a functional relationship among database attributes. The concept of functional dependency was quickly popularized among researchers. The importance of functional dependencies cannot be ignored both for synthesizing relational database schemas and for the investigation of the properties of these schemas [2]. We restrict the class of data dependencies considered here to the functional dependencies.

Even though the concept of functional dependency is simple enough but applying it to schema design requires a fairly complex mathematical treatment.

The synthesis approach was pioneered by Bernstein [3]. In the synthesis approach, relations are created directly from the set of FDs. The core of the problem lies in determining the proper set of FDs that should be used.

The traditional Synthesis algorithm has appeared in [2], [3], [4], [5], [6], [7]; [8], [9], [10], and [11], which have covered most of the issues relating to synthesizing relational schema very effectively. But interestingly, in these algorithms, details of the procedure that how to obtain a minimal cover of functional dependencies in a systematic way for synthesizing a relational schema, were not clearly included.

Minimal cover of functional dependencies (FDs) is also not the new concept in database designing. Synthesizing relations into third normal form (3NF) depends on producing a minimal cover for a set of FDs [3]. Especially Maier [12] discussed various notions of minimality in early 80s. Despite the fact that the available algorithms are using the same system of inference rules defined by Armstrong [13] to find redundancies among the set of FDs, but still they require some enhancement in the procedure for complex handling of these FDs.

"Given some particular set F of FDs that needs to be enforced, it is sufficient for the system to find and enforce the FDs in an irreducible equivalent set I instead" [9]. Because the irreducible set I contains minimum number of FDs without any possible redundancy therefore it is cheaper to test I than it is to test F itself and hence we prefer it to solve the related problems more quickly instead of using original set F of FDs [3], [8] (further discussions are in sections 3.4, 3.5 and 3.6). By this method, a minimal cover of FDs can be enforced on a universal relation scheme of an application to synthesize its relational schema and can get directly the optimal higher normal forms not less than 3NF (Third normal form) [7], [10], [11].

The minimization of FDs hence makes the program or database more structured, modifiable, and less redundant.

In this paper, an extended algorithm is presented to obtain an optimal minimal cover of set of FDs in a systematic and simplified manner. The proposed algorithm, derived in section 4.3, is the extension and formalization of the existing ones [3], [7] [8], [9]. It describes how and by which axiom the redundancy from an FD can be removed effectively. We in this paper do not assume a familiarity with earlier work on minimal covers of functional dependencies.

It is generally agreed that to perform this synthesis only one minimal cover is required; however, a given set of FDs may have several minimal covers.

The proposed algorithm contains some basic propositions inferred from Armstrong's axioms [13] to show how redundant attributes can be removed from a set F of FDs and produces an appropriate optimal minimal cover of FDs from a given universal relation scheme R. This optimal minimal cover then can be used for synthesizing relations in 3NF. In this way, the database designers eliminate uncontrolled redundancies and possible anomalies from a relational schema more confidently.

2. OTHER RELATED APPROACHES

Sudha Ram in [7] suggested a modified synthesis algorithm in which he mainly focused to find all possible minimal covers of the set F by retaining all FDs that have been identified as potentially redundant. At the end, he applied a heuristic approach to select an optimal minimal cover. It is seldom useful especially when we are given equivalent classes of attributes but surely it is much time consuming as he claimed in his paper.

Some studies presented in [16], [17], [18] have focused on the methods for discovering FDs and Approximate FDs from large databases to minimize the number of FDs to be checked. They all have suggested an incremental minimal cover computation during each phase of discovering FDs/AFDs of set F from databases, by using the concept of equivalent classes, closure of set of attributes [9], and candidate set of attributes. While computing incremental minimal cover, they all have used the same traditional synthesis algorithm. Further discussions in these studies have focused on time complexities of the methods used to discover FDs/AFDs.

The next section clarifies the terminology and notations used in this paper and briefly describe the basic database design concepts. However, the reader is assumed to be familiar with the theory of functional dependencies and concept of normal forms up to the level of 3NF.

3. OVERVIEW

Most of the definitions illustrated here are from [2], [8], and [9].

3.1 Functional Dependencies

Let R be a relation scheme. Let A and B are sets of attributes in relation scheme R. We say that B functionally depends on A, or that A functionally determines B in R, and we write A → B if and only if in the relation r(R) for any two tuples t_1 and t_2, if

$t_1[A] = t_2[A]$ *implies* $t_1[B] = t_2[B]$.

Functional dependencies are constraints of the form A → B. These constraints express the potential one-to-one and many-to-one relationships among the attributes of R.

Formally, it is said that an FD A → B is embodied in R if A is a designated key of R and B is an attribute of R. The set of FDs that is embodied in a relational schema is the union of all the FDs embodied in the relation schemes of the schema.

3.2 Armstrong's Axioms

The most famous system of axioms (i.e. inference rules) for the functional dependencies was introduced by Armstrong in [13]. Every problem dealing with FDs requires a manipulation of FDs according to these axioms:

Reflexivity axiom 1. *If* $B \subseteq A$, *then* $A \rightarrow B$ *also* $A \rightarrow A$

Transitivity axiom 2. *If* $A \rightarrow B$ *and* $B \rightarrow C$, *then* $A \rightarrow C$

Augmentation axiom 3. *If* $A \rightarrow B$ *and* $C \in R$, *then* $AC \rightarrow B$

This set of axioms allows us to derive new dependencies that are logical consequence of the set F of functional dependencies. In [10] there is a proof that Armstrong's axioms are sound and complete system of axioms for functional dependencies.

Several further rules were derived from the above axioms. These rules are the supporting rules for computing F^+. The set of all FDs that are derivable from F by repeated applications of the rules (including the FDs in F) is called the closure of F and is denoted by F^+.

Union Rule 1. *If* $A \rightarrow B$ *holds and* $A \rightarrow C$ *holds, then* $A \rightarrow BC$ *holds*

Decomposition Rule 2. *If* $A \rightarrow BC$ *holds, then* $A \rightarrow B$ *holds and* $A \rightarrow C$ *holds*

Pseudo-transitivity Rule 3. *If* $A \rightarrow B$ *holds and* $BC \rightarrow D$ *holds, then* $AC \rightarrow D$ *holds*

Composition Rule 4. *If* $A \rightarrow B$ *holds and* $C \rightarrow D$ *holds, then* $AC \rightarrow BD$ *holds*

The above rules can be proved from Armstrong's axioms that they are correct.

General Unification Theorem. Darwen in [11], proved the extension of Composition rule, which he called the General Unification Theorem.

If $A \rightarrow B$ *holds and* $C \rightarrow D$ *holds, then* $A \cup (C - B) \rightarrow BD$ *holds, as if B is redundant attribute.*

The following is one of the special cases of Darwen's theorem:

If $A \rightarrow B$ *holds and* $AB \rightarrow C$ *holds, then* $A \rightarrow C$ *holds.* **(1)**

As B is redundant attribute in $AB \rightarrow C$.

3.3 Redundant Attributes

As in [9], an attribute of a FD is redundant if we can remove it without changing the closure of set of FDs. Formally redundant attributes are defined as follows:

Consider a set F of FDs and the FD $\alpha \rightarrow \beta$ in F.

- *Attribute A is redundant in* α *if* $A \in \alpha$, *and F logically implies* $(F - \{\alpha \rightarrow \beta\}) \cup \{(\alpha - A) \rightarrow \beta\}$

- *Attribute A is redundant in β if A ∈ β, and the set of FDs (F – {α → β}) U {α → (β – A)} logically implies F.*

3.4 Irreducible set I of FDs

The set F of FDs of a relation scheme R is reduced to an extent that further it can't be reduced without changing the closure F^+, such that F logically implies all dependencies in I, and I logically implies all dependencies in F. Hence, I and F are equivalent if their closures are equivalent that is $I^+ = F^+$. Then, testing whether I is satisfied is equivalent to testing whether F is satisfied. This irreducible set I is also called minimal cover of set F of FDs in the relational model.

3.5 Notion of Minimality

I is minimal cover in a certain sense –

- It does not contain redundant attributes
- Right side of each FD in F has a single attribute, and
- It is left irreducible set of determinants

"If A → B holds in r(R) and there is no proper subset Á of A such that Á → B holds in r(R), then A → B is a "left-irreducible" FD in r(R). In this case, B is sometimes said to be fully dependent on A. Conversely, if there is such a subset Á, then A → B is a "left-reducible" FD in r(R), and B is therefore not fully dependent on A" [11].

In this paper, the minimal cover of F means LR- minimum cover of F as described by Maier in [12].

3.6 Optimal Minimal Cover

A set of FDs I is optimal if there is no set of FDs G with fewer attribute symbols such that G ≡ I. That is:

- It is LR- minimum cover, and
- FDs with the same left side have been combined.

Repeated symbols are counted as many times as they occur [12].

3.7 Keys

The concept of key is one of the most important concepts in the relational model. Every record or tuple in a relation has to be defined in such way that it could be uniquely specified and queried. As in [11]

- *If A → B is an FD in r and AUB constitute the entire heading of r, then A is a Super key of r.*
- *If A → B is a left-irreducible FD in r and A U B constitute the entire heading of r, then A is a key (Candidate key) of r.*

3.8 Normalization

Normalization is the act of decomposing a given relation scheme that fails to satisfy a certain normal form, such that the result of decomposition is an equivalent set of two or more "smaller" relation schemes that do satisfy that normal form [1], [2], [6], [10].

The analyses of Codd's normal forms are out of the scope of this paper therefore not defined here.

3.9 Bijection

A bijection (or equivalent class or one-to-one correspondence) is a function giving an exact pairing of the elements of two sets. Every element of one set is paired with exactly one element of the other set, and every element of the other set is paired with exactly one element of the first set. There are no unpaired elements. In formal mathematical terms, a bijective function $f: X \rightarrow Y$ is a one to one and onto mapping of a set X to a set Y [3], [12].

A bijection from the set X to the set Y has an inverse function from Y to X $(Y \rightarrow X)$. If X and Y are finite sets, then the existence of a bijection means they have the same number of elements.

4. METHODOLOGY

As earlier explained, the algorithm formulated here is not a new procedure as a whole. The authors in fact extend the existing algorithm of traditional synthesis approach used initially by [3] (discussed here in section 4.1). By considering it as the foundation algorithm, an extension is suggested here by including some basic set of propositions that is again implied from the Armstrong's axioms and the Darwen's theorem illustrated in (1). The set of propositions and their preferred sequence may be added in step-2 of the synthesis algorithm to provide users a simplified and systematic method that how to make an FD left irreducible.

One of the applications of Armstrong's axioms is to make new FDs inferred from the given set F of FDs to find F^+. The database designers are using them as a reverse tool to make the given set of FDs reduced to an irreducible form by removing redundancies among them. Some designers are using the concept of augmentation rule for this purpose. No doubt the augmentation rule is a powerful tool to expand and generate all possible set of FDs implied from the given set F, but for the case of minimal cover, the use of augmentation rule is a tricky one and should carefully be handled. That is one should know exactly to which FD, for which attribute, and for what purpose in the given set F, the augmentation rule is going to be applied. Otherwise there may be a higher chance of errors to remove redundancies among FDs and therefore this may lead to obtain various irrelevant covers.

In the proposed algorithm, there is no need of augmentation rule, the designers may use pseudo transitivity rule instead. Because, using pseudo transitivity rule for removing redundancies among the set of FDs requires two basic conditions. First, we compare only that pair of FDs of F who's RHS (dependent side) is same. Second, one FD of the pair should be left-irreducible and the other is left-reducible on the basis of the given source FD of F (for further detail see proposition 5). The optimal minimal cover obtained in this way will be more appropriate and as a result, the designers can directly synthesize the given relation scheme to the higher normal forms with more confidence and surety.

4.1 The Classical Synthesis Approach

A brief summary of the traditional synthesis approach is presented in this section before author suggests an extension. This algorithm is available in [2], [3], [7], [8], [9] and in most of the literatures. The synthesis algorithm is summarized below:

Algorithm 4.1.1:

Step 1: By Armstrong's Decomposition rule, we can assume without loss of generality that every FD in F has a singleton RHS.

Step 2: For each FD in F, we examine each attribute A in the LHS; if deleting A has no effect on the closure F^+; we delete A from the LHS of FD because A was a redundant attribute such that no FD in F contains a redundant attribute on LHS. Such an FD is called Left-irreducible

Step 3: For each FD remaining in F, if deleting from F has no effect on the closure F^+, we delete FD from F

Step 4: The left side of each FD in F should be unique, that is, using union rule to combine such FDs to one so that all of the FDs in each group have identical left hand side.

Step 5: For each group, a relation consisting of all the attributes appearing in that group can be constructed. The set of relations constructed constitutes a schema for the given set of FDs.

4.2 Important Propositions that are used to Extend Algorithm 4.1.1

Most synthesis approaches produce just one minimal cover and use it to generate the relations in 3NF [3]; however, the order in which FDs are submitted to Algorithm 4.1.1 may affect the minimal cover produced. Distinct covers may be produced by reordering the FDs. In order to find all minimal covers, the FDs have to be reordered in all possible distinct ways. This is extremely inefficient [7].

The debate in this paper mainly focuses on the step-2 of the Algorithm 4.1.1. This is the major step of the algorithm in which redundant attributes from the left side of the set F of FDs are found to be removed. Here we consider five simple propositions that will play a very important role to formulate this extended algorithm to discover an appropriate and optimal minimal cover of FDs. Later, the proposed algorithm is verified by using some popular examples.

Proposition 1.

Let F = {A → B, and AC → B} in r(R) then: I = {A → B}

Proof:

As FD AC → B is generated by augmentation rule from A → B [2], this implies that A can determine B independently therefore C is extra here and hence whole FD is redundant here, this implies that

I = {A → B}.

Proposition 2.

Let F = {A → B and AB → C} in r(R) then:I = {A → BC}

Proof:

Using "(1)", B is redundant in AB → C. Also, in [2] and [9], "A stronger FD always implies a weaker one therefore; implication in the opposite direction is trivial in the case of redundant attribute present in weaker FD". In other words B is *non-implied* extraneous attribute on the LHS. This implies that

I = {A → BC}.

Proposition 3.

Let F = {A → B, B → C and A → C} in r(R) then:
I = {A → B, B → C}

Proof:

A → C is redundant FD in F and can be removed because this FD is generated by transitivity rule [13] from A → B, and B → C. This implies that

I = {A → B, B → C}.

Proposition 4. *(Extension of Prop.2 & Prop.3)*

Let F = {A → B, B → C, and AC → D} in r(R) then:

I = {A → BD and B → C}

Proof:

C is redundant in AC → D because by transitivity rule A → C hold but this FD is redundant. This implies, as in Prop.2, that

I = {A → BD and B → C}.

We may further extend this proposition to any level as:

Let F = {A → B, B → C, C →D, and AD → E} in r(R) then:

I = {A → BE, B → C and C →D}.

Proposition 5.

Let F = {A → B, BC → D, and AC → D} in r(R) then:

I = {A → B and BC → D}

Proof:

A is redundant in AC → D and can be replaced by B as this FD is generated by Pseudo-transitivity rule from BC → D using the source FD A → B. Hence whole FD AC → D is redundant here and can be removed. It is noticed that the FD BC → D should hold in F otherwise we can't apply this proposition. Another condition to apply this proposition is that we can compare that pair of FDs who's RHS (dependent side) is same. This implies that

I = {A → B and BC → D}.

This proposition can be further extended as:

Let F = {AC → D, CD → B and AC →B} in r(R) then:

I = {AC → D, CD → B}.

4.3 An Extended Algorithm

All the extensions are made only to step 2 of the Algorithm 4.1.1.

Algorithm 4.3.1:

Step 1: The right-hand-side (the dependent) of every FD in F should contain just one attribute. (Using decomposition rule)

Step 2: The left-hand-side (the determinant) of every FD in F is irreducible in turn; meaning that no attribute can be discarded from the determinant without changing the closure F^+.

In authors' point of view this can easily be done by applying the above discussed propositions in the following sequence:

Step2 (a): Consider first those FDs to which Prop.1 applied if any

Step2 (b): Consider second those FDs to which Prop.2, Prop.3 and/or Prop.4 applied if any

Step2 (c): Consider in the last to which Prop.5 applied if any

Step 3: Repeat step-2. till no FD in F can be discarded from F without changing the closure F^+.

Step 4: The left side of each FD in F should be unique, that is, using union rule to combine such FDs to one so that all of the FDs in each group have identical left hand side.

Step 5: For each group, construct a relation consisting of all the attributes appearing in that group.

Propositions 2, 3, and 4 are combined in step2 (b) as they all are related to transitivity rule but they should be examined separately.

This algorithm can be implemented in the same $O(n^2)$ time complexity as discussed by Bernstein in [3] and Maier in [12]. As the authors in this extended algorithm do nothing with the number of steps but here they are trying to explain how an optimal minimal cover of FDs from the traditional synthesis algorithm can be achieved efficiently.

An illustration of the proposed algorithm is given in the following example.

4.4 Example

Consider scheme R= {A, B, C, D, E, F}, which satisfies the following set of FDs [7], [8], [14], and [15].

F = {AB → C, C → A, BC → D, ACD→ B, BE→ C, CE → FA, CF → BD, D → EF}

Solution: (Using Algorithm 4.3.1)

Step 1.

AB → C

C → A

BC → D

ACD→ B

BE → C

CE → A

CE → F

CF → B

CF → D

D → E

D → F

Step 2. (a)

AB→ C

C → A

BC → D

ACD→ B

BE → C

CE → A (by Prop. 1, reduced to C→ A; deleted)

CE → F

CF → B

CF → D

D → E

D → F

Step 2. (b)

AB→ C

C → A

BC → D

ACD→ B (by Prop. 2, reduced to CD→ B)

BE → C

CE → F

CF → B

CF → D

D → E

D → F

Step 2. (c)

AB → C

C → A

BC → D

CD→ B (reduced to CF→ B using D → F & Prop. 5; redundant; deleted)

BE → C

CE → F

CF → B

CF → D (as BC → D expands to CFC → D and reduced to CF → D using CF → B & Prop. 5; redundant; deleted)

D → E

D → F

Step 3.

AB → C

C → A

BC → D

BE → C

CE → F

CF → B

D → E

D → F

Step 4. I = {AB → C, C → A, BC → D, BE→C, CE → F, CF → B, D → EF}

According to authors' point of view, this is the optimal minimal cover of this problem. Another cover found in [14] and [15] about this problem would be further reducible to the above set I, as follows.

2nd minimal cover of the same problem [14], [15]:

I_{Ulm} = {1. AB → C

2. C → A
3. BC → D
4. CD → B
5. BE → C
6. CE → F
7. CF → D
8. D → E

9. $D \rightarrow F\}$

Solution: (Using Algorithm 4.3.1)

- Observe 4: Using augmentation and composition rule, $CD \rightarrow B$ is included here and $CF \rightarrow B$ is removed from above solution, where as $CD \rightarrow B$ can be further reduced to $CF \rightarrow B$ by using $D \rightarrow F$ and prop 5. therefore $CD \rightarrow B$ should be replaced by $CF \rightarrow B$ so that

I_{Ulm} = {1. $AB \rightarrow C$
2. $C \rightarrow A$
3. $BC \rightarrow D$
4. $CF \rightarrow B$
5. $BE \rightarrow C$
6. $CE \rightarrow F$
7. $CF \rightarrow D$
8. $D \rightarrow E$
9. $D \rightarrow F\}$

- Observe 7: $CF \rightarrow D$ can also be deleted because if $BC \rightarrow D$ of 3 expands to $CFC \rightarrow D$ and then reduced to $CF \rightarrow D$ again by using $CF \rightarrow B$ of 4 & Prop. 5 then it shows that it is redundant FD; therefore deleted.
- Hence $I_{Ulm} = I$

This optimal minimal cover I has also produced by Sudha Ram from the same problem but after applying a lengthy modified synthesis algorithm [7] to find all possible minimal covers and then applying a heuristic approach to select the optimal one, which was same set I, illustrated above.

It is observed that when we try to find all possible minimal covers of a set F of FDs then we often missed some of the FDs to fully reduce them into an appropriate optimal minimal cover mainly because of using augmentation rule; and in fact prior to this we did not have proper procedure to reduce all FDs of F systematically.

4.5 Bijection Cases

It is believed by the researchers that analyzing FDs in different order might result more than one minimal covers whereas, on the other hand, if there are bijection cases in F like $A \rightarrow B$ and $B \rightarrow A$, then there must be more than one optimal minimal covers obtained by changing the order of FDs of F in the algorithm.

The bijection FDs occur when two or more attributes determine each other. They are alternate keys. This was discussed in detail by Bernstein [3] and Maier [12] in which they also proved that any two minimum coverings of F will produce the equal as well as minimal number of relations. Therefore a synthesized schema will contain a minimal number of relations.

Following is an example of more than one minimal covers of FDs in case of bijections:

4.6 Example:

Scheme R= {A, B, C} satisfies the following FDs:
$F = \{A \rightarrow BC, B \rightarrow AC, C \rightarrow AB\}$

Minimum Cover I: *(Using Algorithm 4.3.1)*

Step 1. $A \rightarrow B$
$A \rightarrow C$
$B \rightarrow C$
$B \rightarrow A$
$C \rightarrow A$
$C \rightarrow B$

Step 2. (a) N/A

Step 2. (b) $A \rightarrow B$
$A \rightarrow C$ (transitive FD, therefore deleted)
$B \rightarrow C$
$B \rightarrow A$ (transitive FD, therefore deleted)
$C \rightarrow A$
$C \rightarrow B$ (transitive FD, therefore deleted)

Step 3. $A \rightarrow B$
$B \rightarrow C$
$C \rightarrow A$

Step 4. $I_1 = \{A \rightarrow B, B \rightarrow C, C \rightarrow A\}$

Minimum Cover II: *(Using Algorithm 4.3.1 and order of FDs in Step 2. is changed)*

Step 1. $A \rightarrow B$
$A \rightarrow C$
$B \rightarrow C$
$B \rightarrow A$
$C \rightarrow A$
$C \rightarrow B$

Step 2. (a) N/A

Step 2. (b) $A \rightarrow B$ (transitive FD, therefore deleted)
$A \rightarrow C$
$B \rightarrow C$ (transitive FD, therefore deleted)
$B \rightarrow A$
$C \rightarrow A$ (transitive FD, therefore deleted)
$C \rightarrow B$

Step 3. $A \rightarrow C$
$B \rightarrow A$
$C \rightarrow B$

Step 4. $I_2 = \{A \rightarrow C, B \rightarrow A, C \rightarrow B\}$

Here two optimal minimal covers of the same problem are found after changing the order of FDs, where two or more attributes of R are bijection sets. It is noticed that there would be no further optimal minimal covers possible of this problem even if we try it after changing the order of propositions in Step 2(b) in the proposed algorithm.

It is further noticed that the two optimal minimal covers I_1 and I_2 generate exactly equal number of FDS. This result shows the soundness of the proposed algorithm as it is claimed by Bernstein

[3] and Maier [12] that all minimum coverings of F will produce the equal number of relations.

5. RESULTS

Following results are concluded:

1. By using the proposed algorithm we can obtain an optimal minimal cover I of FDs from any set F of FDs in a systematic and effective manner.
2. However, we can generate more than one optimal minimal covers but all such sets produce equal as well as minimum number of relations for synthesizing third normal form relations of a relational schema.
3. The time complexity of this algorithm remains the same as $O(n^2)$ as analyzed in earlier papers on minimum covers of F of FDs.

6. CONCLUSION

We have suggested here an extension to the traditional synthesis algorithm to discover an optimal minimal cover of FDs more efficiently, which utilizes the accepted and popular inference axioms of Armstrong's and Darwen's theorem. This extension is suggested in step 2 of the traditional synthesis algorithm to produce set of FDs left irreducible. The aim of this algorithm is to produce an optimal minimal cover in a simplified and systematic manner and as a whole synthesize relational schema successfully and effectively.

7. REFERENCES

[1] Codd, E.F. 1972. Further Normalization of the Database Relational Model. In *Database Systems, Courant Inst. CS Symposium. 6, R. Rustin, ed.*, Prentice-Hall, Englewood Cliffs, N.J., 33-64, 1972.

[2] Beeri, C. and Bernstein, P.A. 1979. Computational Problems Related to the Design of Normal Form Relational Schemas. *ACM Trans. Database Systems*, 1979.

[3] Bernstein, P.A. 1976. Synthesizing Third Normal Form Relations from Functional Dependencies. *ACM Trans. Database Syst.* 1(4), 277-298, Dec. 1976.

[4] Zaniolo, C. and Melkanoff, M. 1982. A Formal Approach to the Definition and the Design of Conceptual Schemata for Database Schemata. *ACM Transactions on Database Systems,* Vol. 7, No. 1, 24-59, March 1982.

[5] Beeri, C. and Kifer, M. 1986. An Integrated Approach to Logical Design of Relational Database Schemes. *ACM Transactions on Database Systems,* Vol. 11, No. 2, 134-158, 1986.

[6] Fagin, R. 1977. The Decomposition versus Synthetic Approach to Relational Database Design. In *Proceedings of the third International Conference on Very Large Databases*, (Tokyo, Oct. 1977) 441-446.

[7] Sudha Ram, and Curran, S. M. 1988. The Synthesis Approach for Relational Database Design: An Expanded Perspective. In *Proceedings of the Twenty-First IEEE Annual Hawaii International Conference on System Sciences, Software Track,*, Vol. II, 571-580, 1988, ISBN: 0-8186-0842-0

[8] Date, C.J., Kannan, A., and Swamynathan, S. 2006. *An Introduction to Database Systems.* 8th ed., India: Dorling Kindersley, 2006.

[9] Silberschatz, A., Korth, H.F., and Sudarshan, S. 2006. *Database System Concepts.* 5th ed., Boston, MA: McGraw-Hill, ISBN: 007124476X, 2006.

[10] Maier, D. 1983. *The Theory of Relational Databases*. Rockville, USA: Computer Science Press, 1983.

[11] Darwen, H. and Date C. J. 1992. The Role of Functional Dependence in Query Decomposition. In *Relational Database Writings 1989-1991*, Reading, Mass.: Addison-Wesley 1992.

[12] Maier, D. 1980. Minimum Covers in the Relational Database Model. *J. ACM* 27(4), 664-674, 1980.

[13] Armstrong, W.W. 1974. Dependency Structures of Database Relationships. In *Proceedings of IFIP Congress* (Stockholm, Sweden, 1974).

[14] Date, C.J. 2003. *An Introduction to Database Systems.* 8th ed., Instructor's Manual, 11.8 - 11.9, 2003.

[15] Ullman, J.D. 1982. *Principles of Database Systems*. Second Edition, 1982, Computer Science Press, Rockville, MD 20850.

[16] Atoum, J., Bader, D., and Awajan, A. 2008. Mining Functional Dependency from Relational Databases using Equivalent Classes and Minimal Cover. *Journal of Computer Science* 4 (6), 421-426, 2008.

[17] Yao, H., Hamilton, H. J. 2008. Mining Functional Dependencies from Data. *Journal of Data Mining and Knowledge Discovery*, v.16 n.2, 197-219, April 2008. [doi:10.1007/s10618-007-0083-9]

[18] Atoum, J. 2009. Mining Approximate Functional Dependencies from Databases Based on Minimal Cover and Equivalent Classes. *European Journal of Scientific Research,* Vol. 33 Issue 2, 338, Jul 2009.