

TOWARDS A SOUND VIEW INTEGRATION

METHODOLOGY

M A Casanova

Centro Científico - IBM do Brasil
Caixa Postal 853
70 000, Brasília, DF
Brasil

V M P Vidal

Departamento de Informática
Pontifícia Universidade Católica do RJ
22 453, Rio de Janeiro, RJ
Brasil

ABSTRACT

View integration is investigated with the help of three classes of interrelational dependencies, inclusion dependencies, exclusion dependencies and union functional dependencies. The process of view integration is divided into two steps, combination and optimization. View combination consists in defining new interrelational dependencies that capture similarities between different views. The optimization step tries to reduce redundancy and the size of the schema. Finally, general results about interrelational dependencies are presented that lead to an optimization procedure for a restricted class of schemas.

1 INTRODUCTION

We investigate in this paper how certain classes of dependencies can be used to secure sound foundations for relational view integration.

View integration is a database design method that suggests to synthesize an integrated conceptual schema by combining previously obtained schemas that reflect, for each class of users, their view of the enterprise [NG,TF].

We assume that all views have already suffered a preliminary integration process to detect when entity/relationship sets [Ch] of different views are of the same type. The question then is how to integrate views that have entity/relationship types in common, but which may differ on the entity/relationship sets represented and on the attributes

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

considered.

We propose to divide relational view integration into two steps, combination and optimization. To combine a set of views simply means to define a new schema R that contains all original views, plus a new set of interrelational dependencies that express how data in distinct views is interrelated. The optimization step then tries to modify R to reduce redundancy and the size of the schema. The optimization step is of interest by itself as it is similar in purpose to normalization [BBG,U1,Da2].

The classes of dependencies considered are the inclusion dependencies (INDs) [CFP], the exclusion dependencies (EXDs) and the union functional dependencies (UFDs), besides the usual functional dependencies (FDs). INDs are quite useful for view integration since they can express that two entity/relationship sets, S_1 and S_2 , of the same type, but from different views, are equal or one is a subset of the other; EXDs can express, on the other hand, that S_1 and S_2 are disjoint, UFDs may be used to indicate that attributes of different entity/relationship sets are synonyms. Hence, these constraints avoid integrating views based only on overlapping structures or on naming similarities, as suggested in [NG,WM].

When using these dependencies for view integration we must keep in mind that they interact in unexpected ways (see [CFP]). To overcome this difficulty we state separability results that indicate when sets of dependencies from different classes do not interact or, at least, interact in predicted ways. We also define a subclass of INDs for which logical implication can be decided in polynomial time.

This paper is divided as follows. Section 2 contains basic definitions. Section 3 discusses view integration. Section 5 states results about the dependencies considered that justify the optimization procedure discussed in Section 4. Finally, Section 6 contains conclusions and directions for future research.

2. PRELIMINARY DEFINITIONS

A relational schema is a pair $R = (S, C)$ where $S = \{R_1[U_1], \dots, R_n[U_n]\}$ is a set of relation schemes and C is a set of dependencies over S . A database or state of S is a set $D = \{r_1, \dots, r_n\}$ of relations, one for each scheme, such that r_1 is n_1 -ary, if $|U_1| = n_1$.

If X and Y are sequences, we use $Y \subseteq X$ to indicate that Y is a permutation of a subsequence of X .

In the rest of this section, let $R = (S, C)$ be a schema and D be a state of S .

A functional dependency (FD) $[Ar, Da, U1]$ over S is a statement γ of the form $R_1 \cdot X \rightarrow Y$, where $1 \leq i \leq n$ and X, Y are sequences of attributes of R_1 . We say that γ is valid in D iff, for any $t, u \in r_1$, if $t[X] = u[X]$ then $t[Y] = u[Y]$.

An inclusion dependency (IND) $[CFP, Dal, FV, L1]$ over S is a statement σ of the form $R_1[X] \subseteq R_j[Y]$, where $1 \leq i, j \leq n$ and where X and Y are sequences of attributes of R_1 and R_j , respectively, such that $|X| = |Y|$. We say that σ is valid in D iff $r_1[X]$ is a subset of $r_j[Y]$.

INDs are quite useful for view integration since they can express that two entity/relationship sets of the same type, but from different views, are equal or one is a subset of the other. For example, if user A sees $EMP[EN, DN]$ and user B sees $SECRETARY[EN, TYPING_SPEED]$, then $SECRETARY[EN] \subseteq EMP[EN]$ indicates that B sees a subset of the entities perceived by A.

An exclusion dependency (EXD) over S is a statement δ of the form $R_1[X] \mid R_j[Y]$ where X and Y are sequences of attributes of R_1 and R_j , respectively, such that $|X| = |Y|$. We say that δ is valid in D iff $r_1[X]$ and $r_j[Y]$ are disjoint. A vacuous EXD over R_1 is any EXD of the form $R_1[X] \mid R_1[Y]$ that is valid in D iff $r_1 = \emptyset$. EXDs have not been considered in the literature, except implicitly

in connection with a database abstraction called partitioning.

EXDs can express that two entity/relationship sets of the same type, but from different views, are disjoint. For example, if user A sees $UNDERGRAD[SN, DEPT]$ and user B sees $GRAD[SN, DEPT]$ then $UNDERGRAD[SN] \mid GRAD[SN]$ indicates that A and B see disjoint sets of students.

A union functional dependency (UFD) over S is a statement ψ of the form $\langle R_{i_1} X_1 \rightarrow Y_1, \dots, R_{i_m} X_m \rightarrow Y_m \rangle$ where X_j, Y_j are sequences of attributes of R_{i_j} such that $|X_j| = |X_k|$ and $|Y_j| = |Y_k|$, for any j, k in $[1, m]$. We say that ψ is valid in D iff, for any j, k in $[1, m]$ (j may be equal to k), for any $t \in r_{i_j}$ and any $u \in r_{i_k}$, if $t[X_j] = u[X_k]$ then $t[Y_j] = u[Y_k]$. Note then that ψ implies the FD $R_{i_j} X_j \rightarrow Y_j$, for each $j \in [1, m]$. Moreover, if $m=1$, then ψ reduces to the FD $R_{i_1} X_1 \rightarrow Y_1$. UFDs are a special case of the dependencies considered in $[Ca, K1]$.

UFDs will be used to indicate that attributes from different relation schemes are synonyms. For example, consider the set of relation schemes $S = \{STUDENT[ID, DEPT], TEACHING_ASSISTANT[ID, DEPT]\}$. Let $D = \{s, s'\}$ be a database for S . Suppose that ID is a key of each of the above schemes. This does not imply that if $(1, d) \in s$ and $(1, d') \in s'$ then $d=d'$. Indeed, $DEPT$ in $STUDENT$ may be the department where the student is enrolled, whereas $DEPT$ in $TEACHING_ASSISTANT$ may be the department where the student is working, which is not necessarily the same where he is enrolled. If otherwise $DEPT$ means the same in both schemas, then ID must be the common key. That is, if $(1, d) \in s$ and $(1, d') \in s'$ then $d=d'$, which is expressed by $\langle STUDENT ID \rightarrow DEPT, TEACHING_ASSISTANT ID \rightarrow DEPT \rangle$.

Given a set of dependencies Σ and a single dependency σ over S , we say that Σ logically implies σ with respect to finite databases, or that σ is a logical consequence of Σ , iff for every finite database D of S , if all dependencies in Σ are valid in D , then σ is also valid in D . If this is the case, we write $\Sigma \models_f \sigma$. We say that a dependency α is trivial iff $\emptyset \models_f \alpha$.

3 BASIC VIEW INTEGRATION

The first step of any design methodology based on view integration would naturally be to obtain a schema (or view) for each group of users. The size of the schema should be such that it can be obtained directly by a monolithic design method. Then, the individual views would be integrated into a single conceptual schema. This step involves detecting when structures from different views represent the same information.

There are at least three very serious problems connected with view integration. First, one should expect to start with a large number of different views. This problem can in part be solved by a hierarchical view integration strategy. That is, groups of closely related views are integrated separately, obtaining a new set of views, and so on until the final schema is constructed. For example, views related to the marketing department are integrated to obtain the view of the whole department, which may be significant by itself, and so on.

The second problem refers to the fact that widely different structures may actually represent the same information. This problem is discussed in [Ke] and lies outside the scope of this paper.

Finally, the third problem is that even after resolving structural differences and classifying entity/relationship sets into types, we must still interrelate and integrate entity/relationship sets of different views, that may also differ on the attributes considered. It is to this problem that we restrict ourselves from now on.

Suppose that each view V_i is described by a relational schema $R_i = (S_i, C_i)$ where C_i contains only INDs, EXDs, UFDs and FDs, and where S_i and S_j do not have relation names in common, for each i and j in $[1, k]$, with $i \neq j$. The combination step consists in defining new INDs and EXDs to indicate how entity/relationship sets are interrelated, and new UFDs to indicate which attributes are regarded as synonyms (see the examples in Appendix A). Thus, the result of combining views $R_i = (S_i, C_i)$, $i=1, \dots, k$, is a schema $R = (S, C)$ with $S = \bigcup_{i=1}^k S_i$ and $C = \bigcup_{i=1}^k C_i \cup C'$, where C' is a set of INDs, EXDs and UFDs over S . The optimization step will then try to minimize any redundancy impli-

cit in $R = (S, C)$, regardless of whether or not the original views R_1, \dots, R_n already contained redundancies (although ideally we may assume that the views were optimized before the combination step). The optimization step will also try to reduce the size of S and C by combining relation schemas when appropriate.

The goal of the optimization step, as stated above, is somewhat vague, specially because INDs, EXDs and UFDs interact in a complex way (see Section 5). Thus, from now on, we will concentrate on a restricted class of schemas for which we can precisely state the goals of the optimization step.

Let $R = (S, C)$ be a schema where C contains only FDs, UFDs, INDs and EXDs. We say that R is restricted iff there is a function $f: S \rightarrow K$ associating a sequence $K_i \in K$ to each $R_i \in S$ such that

- (i) C implies that K_i is the only key of R_i , and the only FDs over R_i in C are those implying that K_i is a key of R_i ,
- (ii) if $R_i[X] \subseteq R_j[Y]$ is in C , then $X=Y=K_j$,
- (iii) if $\langle R_{i_1} X_{i_1} \rightarrow Y_{i_1}, \dots, R_{i_m} X_{i_m} \rightarrow Y_{i_m} \rangle$ is in C , then $X_{i_1} = \dots = X_{i_m} = K_{i_1} = \dots = K_{i_m}$ and $|Y_j|=1$, $j \in [1, m]$,
- (iv) for any $R_i \in S$ and any attribute A of R_i , A occurs in at most one UFD in C ,
- (v) if $R_i[X] \mid R_j[Y]$ is in C , then $X=Y=K_i=K_j$,

To reinforce the importance of K , we denote R by the triple (S, C, K) (the function f is defined by underlying the attributes of $R_i \in S$ that form K_i , as usual). An example of a restricted schema appears in Appendix A.

Intuitively, in a restricted schema $R = (S, C, K)$, S represents a collection of entity/relationship sets, identified by their keys. An IND $R_i[K_i] \subseteq R_j[K_j]$ expresses that tuples of R_i refer to tuples of R_j via their keys, if K_j is not the key of R_i , or that the set denoted by R_i is a subset of the set denoted by R_j , if K_j is the key of R_i . An EXD $R_i[K_i] \mid R_j[K_j]$ indicates that R_i and R_j denote disjoint sets. An UFD $\langle R_{i_1} K_{i_1} \rightarrow A_{i_1}, \dots, R_{i_m} K_{i_m} \rightarrow A_{i_m} \rangle$ indicates that A_{i_1}, \dots, A_{i_m} must be regarded as synonyms, condition (iv) indirectly captures the fact that the notion of synonym is transitive.

We note that relational schemas generated from

entity-relationship diagrams [Ch] or based on the structural model of [WM] are special cases of restricted schemas. Moreover, we note that each relation scheme in S is obviously in BCNF (by condition (1)), which avoids the redundancies causing the well-known anomalies of non-BCNF schemes [BBG,U1].

Returning to view integration, assume that the result of the combination step is a restricted schema $R = (S, C, K)$. We propose an optimization heuristics that will achieve the following four goals: (i) R should be transformed so that C implies no INDs of the form $R_j[K_1A] \subseteq R_1[K_1A]$, which indicates that A is redundantly stored in R_j , and where K_1 is the common key of relation schemes $R_1[U_1]$ and $R_j[U_j]$. INDs of this form arise as a consequence of $R_j[K_1] \subseteq R_1[K_1]$ and $\langle R_j, K_1 \rightarrow A, R_1, K_1 \rightarrow A \rangle$ (see Appendix A for an example of how these inclusion dependencies may be created), (ii) R should be transformed so that C implies no INDs of the form $R_1[K_j] \subseteq R_j[K_j]$ and $R_j[K_j] \subseteq R_1[K_j]$, which indicate that R_1 and R_j actually represent the same set of objects, (iii) R should be transformed so that C implies no UFD, since a UFD indicates a potential source of redundancy. The only exceptions are UFDs of the form $\langle R_{1_1}, X_1 \rightarrow Y_1, \dots, R_{1_m}, X_m \rightarrow Y_m \rangle$ in the presence of $R_{1_j}[X_j] \mid R_{1_k}[X_k]$, $1 \leq j, k \leq m$ with $j \neq k$, since a UFD of this form degenerates into a set of FDs $R_{1_j}, X_j \rightarrow Y_j$, $1 \leq j \leq m$, (iv) if C implies a vacuous EXD over R , then R should be eliminated from S , since we do not want to retain relations that must always be empty. (In fact, the presence of vacuous EXDs is a sign of a defective project.)

We observe that goals (ii) and (iv) try to reduce the size of R , whereas goals (i) and (iii) try to minimize redundancy. The transformations mentioned above should be such that the final

schema R' is equivalent to the initial one, R , in the sense that any consistent state of R can be mapped into a consistent state of R' and vice-versa. This point is discussed further at the end of Section 4.

Section 4 presents an optimization procedure for restricted schemas, whose correctness and implementation is discussed in Section 5.

4 AN OPTIMIZATION PROCEDURE

We present in this section an optimization procedure for restricted schemas, whose correctness and implementation is discussed in Section 5.

Before presenting the optimization procedure, we need some extra definitions and conventions. Let $R = (S, C, K)$ be a restricted schema, where $S = \{R_1[U_1], \dots, R_n[U_n]\}$.

We define an equivalence relation $\equiv_R \subseteq [1, n]^2$ such that $i \equiv_R j$ iff $R_i[K_j] \subseteq R_j[K_j]$ and

$R_j[K_j] \subseteq R_i[K_j]$ are both logical consequences of the INDs in C . We use $[1, n] / \equiv_R$ to indicate the partition of $[1, n]$ into \equiv_R -equivalence classes.

To simplify the description of the optimization procedure, we assume that if $K_1 = K_j$ then $U_1 \cap U_j = K_1$. Since the keys are independently given, and by conditions imposed on C , we may simplify the notation of the dependencies in C . An IND $R_i[K_j] \subseteq R_j[K_j]$ is denoted by $R_i \subseteq R_j$, an EXD $R_i[K_j] \mid R_j[K_j]$ is denoted by $R_i \mid R_j$, and an UFD $\langle R_{1_1}, K_{1_1} \rightarrow A_1, \dots, R_{1_m}, K_{1_m} \rightarrow A_m \rangle$ is denoted by $\{R_{1_1}[A_1], \dots, R_{1_m}[A_m]\}$. Moreover, we may assume that $R_1[A]$ occurs in at most one UFD, by definition of restricted schema (condition (iv)).

The optimization procedure is shown in Figure 4.1

FIGURE 4 1

OPTIMIZATION PROCEDURE

```

proc OPTIMIZE(R,R')
/* input   R = (S,C,K) - a restricted schema, where
           S = {R1[U1], ..., Rn[Un]}
           output R' = (S',C',K') - a restricted schema which is
           optimized and equivalent to R
*/

1. begin set S' and C' to ∅ ,
2.   set π to [1,n]/≡R ,
3.   set Σ to be the set of INDs in C ,
   /* goal (i) modify UFDs that imply a dependency
     of the form R1[K1A] ⊆ Rj[KjB]
   */
4.   for each i,j in [1,n] such that Σi =f R1[K1] ⊆ Rj[Kj] do
5.     for each UFD F in C such that there is R1[A], Rj[B] in F do
6.       begin for each A in U1 such that R1[A] ∈ F do
           /* by condition (iv) on C, R1[A] occurs in no other UFD
             by condition (ii) on C and condition (v), A occurs in
             no IND or EXD in C
           */
7.         begin delete A from U1 ,
8.           delete R1[A] from F ,
           end
9.         if F is reduced to a singleton
           then delete F from C ,
           end
   /* goal (ii) create a new relation scheme in S' by combining
     all relation schemes in S that are forced to be equal
   */
10.  for each πk in π do
11.    begin add R'k[U'k] to S' with U'k = ∪i ∈ πk Ui and K'k = K1, for some i ∈ πk,
12.      for each F in C do
13.        add F' to C' where F' is obtained from F by replacing
           R1 by R'k, if i ∈ πk ,
           end
   /* goal (iii) create new relation schemes in S' to eliminate
     any remaining UFD in C'
   */
14.  for each F = {R'1[A1], ..., R'1[Am]} in C' do
15.    begin add a new scheme R'p[U'p] to S' with U'p = K1 ∪ {A1} and K'p = K1 ,
16.      delete F from C' ,
17.      for each UFD F' in C' do
18.        if F' is of the form {R'1[B1], ..., R'1[Bm]}
19.          then begin add B1 to U'p ,
20.            delete F' from C' ,
           end
21.      for each j in [1,m] do
22.        add R'1[K'j] ⊆ R'p[K'j] to C' ,
           end
   /* goal (iv) delete any relation scheme in S' that is forced to be
     empty by C' and create key FDs for each remaining scheme
   */
23.  for each R'1 in S' do
24.    if C' implies a vacuous EXD over R'1
25.      then delete R'1[U'1] from S' and all constraints in C' over R'1 ,
26.      else add R'1 K'1 → U'1 to C' ,
           end
end

```

The optimization procedure can be easily implemented, except for lines 2 (see the definition of Ξ_R), 4 and 24, which involve testing logical implication. Moreover, we must also prove that it achieves the four goals proposed at the end of Section 3. This will be discussed in Section 5.

We note at this point that lines 14 to 22 can be modified to eliminate all UFDs from arbitrary schemas. The resulting procedure is quite similar to the algorithm that creates a schema in 3NF from an arbitrary schema whose dependencies are just FDs (see [U1]). The only sensible difference lies in the creation of INDs on lines 21 and 22, which we believe should exist in the algorithm of [U1].

We conclude this section with a brief discussion about the relationship that exists between $R = (S, C, K)$ and the optimized schema $R' = (S', C', K')$. We can easily modify the optimization procedure so that, for each $R_1[U_1] \in S$, it produces a relational expression e_1 over S' involving only natural joins and projections and, for each $R'_1[U'_1] \in S'$, it produces a relational expression e'_1 over S involving only natural joins, projections and unions. Moreover, these collections of expressions have the following properties. Let $D = \{r_1, \dots, r_n\}$ be a database for S that satisfies C . Let $D(e'_1)$ be the value of e'_1 in D . Then, $D' = \{D(e'_1), \dots, D(e'_m)\}$ is a database for S' that satisfies C' . Now, let $D'(e_1)$ be the value of e_1 in D' . Then, $D = \{D'(e'_1), \dots, D'(e'_n)\}$. That is, e'_1, \dots, e'_n induces a mapping from a consistent state of S to a consistent state of S' , whose inverse is exactly the mapping induced by e_1, \dots, e_n . It is in this sense that we say that R and R' are equivalent. (This notion of equivalence is discussed in [Co]). Finally, observe that e_1 indicates that $R_1[U_1]$ can be treated as a view of S' .

5. SOME RESULTS ABOUT INDs, EXDs, UFDs and FDs

When attempting to use INDs, EXDs, FDs and UFDs in any (automated) database design method, we have to face two difficulties. First, the decision problem for INDs alone is PSPACE-complete [CFP], which implies that most likely a design algorithm that accepts any set of INDs will be computationally hard. Second, we do not even know if the inference problem for INDs and FDs is decidable or not, let

alone when EXDs and UFDs are also considered.

We show in this section how to overcome these difficulties in the context of restricted schemas. Our tactic is to present general results about dependencies and then derive corollaries that help understand the optimization procedure. Our results are somewhat more general than it is actually needed in the hope that they can be reused in other contexts or in connection with other classes of schemas.

Section 5.1 indicates how to construct the equivalence relation Ξ_R used in line 2 of the procedure. Section 5.2 shows how to detect vacuous EXDs in restricted schemas. Finally, Section 5.3 contains results that indicate that the optimization procedure achieves each of the four goals proposed at the end of Section 3.

5.1 RESULTS ABOUT INDs

We describe in this section an easily solvable case of the decision problem for INDs. Recall from Section 2 that, if X and W are sequences, then $X \subseteq W$ indicates that X is a permutation of a subsequence of W . We also recall that $\Sigma \models_f \sigma$ indicates that Σ logically implies σ for finite databases.

THEOREM 5.1 Let $S = \{R_1[U_1], \dots, R_n[U_n]\}$ be a set of relation schemes and Σ be a set of INDs over S . Suppose that if $R_p[W] \subseteq R_q[Z]$ is in Σ then $W = Z$. Then, for any IND $\sigma = R_1[X] \subseteq R_j[Y]$ over S , we have that $\Sigma \models_f \sigma$ iff σ is trivial, or

- (1) $X = Y$, and
- (11) there is a path from R_1 to R_j in the digraph $G_X = (V, E)$, where $V = \{R_1, \dots, R_n\}$ and $(R_p, R_q) \in E$ iff there is $R_p[W] \subseteq R_q[W]$ in Σ such that $X \subseteq W$.

(All proofs appear in Appendix B)

Let $R = (S, C, K)$ be a restricted schema. Recall that $1 \Xi_R j$ iff Σ implies both $R_1[K_j] \subseteq R_j[K_j]$ and $R_j[K_j] \subseteq R_1[K_j]$, where Σ is the set of INDs in C . Now, since Σ satisfies the conditions of Theorem 5.1, by definition of restricted schema, we can use G_K to detect if $R_1[K_j] \subseteq R_j[K_j]$ and $R_j[K_j] \subseteq R_1[K_j]$ are implied by Σ and, consequently, if $1 \Xi_R j$ holds. We may do better, though. Recall from Section 4

that in a restricted schema $R_1[K_j] \subseteq R_j[K_j]$ is abbreviated as $R_1 \subseteq R_j$. Then, we directly obtain the following corollary

COROLLARY 5 1 Let $R = (S, C, K)$ be a restricted schema. Suppose that $S = \{R_1[U_1], \dots, R_n[U_n]\}$. Let Σ be the set of INDs in C . Define $1 \equiv_R j$ iff Σ implies both $R_1 \subseteq R_j$ and $R_j \subseteq R_1$. Then, $1 \equiv_R j$ iff 1 and j lie in the same strongly connected component of $G = (V, E)$, where $V = \{1, \dots, n\}$ and $(i, j) \in E$ iff $R_i \subseteq R_j \in C$. \square

Corollary 5 1 then solves one of the problems raised by an implementation of the optimization procedure (line 2)

5 2 RESULTS ABOUT INDS AND EXDS

We investigate in this section the interaction between INDS and EXDS. Our ultimate goal is to detect when the INDS and EXDS of a restricted schema imply a vacuous EXD, since this is needed to implement our optimization procedure.

We begin our investigation by presenting in Figure 5 1 an axiom system for INDS and EXDS.

THEOREM 5 2 The axiom system for INDS and EXDS is sound and complete (with respect to finite and unrestricted implication).

AN AXIOM SYSTEM FOR INDS AND EXDS

- I1 $\frac{}{R[X] \subseteq R[X]}$
- I2 $\frac{R[A_1 \dots A_n] \subseteq S[A_1 \dots A_n]}{R[A_{1_1} \dots A_{1_m}] \subseteq S[B_{1_1} \dots B_{1_m}]}$
- I3 $\frac{R[X] \subseteq S[Y], S[Y] \subseteq T[Z]}{R[X] \subseteq T[Z]}$
- E1 $\frac{R[X] \mid S[Y]}{S[Y] \mid R[X]}$
- E2 $\frac{R[A_{1_1} \dots A_{1_m}] \mid S[A_{1_1} \dots A_{1_m}]}{R[A_1 \dots A_n] \mid S[B_1 \dots B_n]}$
- E3 $\frac{R[X] \mid R[W] \text{ if } R[X] \mid R[W] \text{ is vacuous}}{R[Y] \mid S[Z]}$

IE1 $\frac{R[X] \mid R[W] \text{ if } R[X] \mid R[W] \text{ is vacuous}}{R[Y] \subseteq S[Z]}$

IE2 $\frac{R[X] \subseteq S[Y], T[W] \subseteq U[Z], S[Y] \mid U[Z]}{R[X] \mid T[W]}$

note in rules I2 and E2, $1_1, \dots, 1_m$ is any permutation of any subsequence of $1, \dots, n$.

FIGURE 5 1

COROLLARY 5 2 Let Σ be a set of INDS and Δ be a set of EXDS. Suppose that if $R_p[W] \subseteq R_q[Z]$ is in Σ then $W = Z$ and that if $R_k[U] \mid R_m[V]$ is in Δ then $U = V$.

Then, $\Sigma \cup \Delta$ implies a vacuous EXD iff there is an EXD $R_1[X] \mid R_j[X]$ in Δ such that there is a node R_a of G_X such that there is a path from R_a to R_1 and a path from R_a to R_j in G_X . \square

Given a restricted schema $R = (S, C, K)$, since all INDS and EXDS in C satisfy the conditions of Corollary 5 2, we can efficiently detect when they imply a vacuous EXD. By a result of Section 5 3 (Theorem 5 4 - Part (1)), we can then use Corollary 5 2 to implement line 24 of the optimization procedure.

5 3 SOME RESULTS ABOUT THE INTERACTION BETWEEN FDS, UFDs, INDS AND EXDS

We now turn to the question of showing that the optimization procedure achieves the four goals set at the end of Section 3. We will again state results about dependencies that have as a corollary the desired conclusions.

Let C be a class of dependencies and $C \subseteq C$. We denote by C^+ the set of all dependencies in C that are logical consequences of C with respect to finite databases. Thus, if Σ is a set of INDS, Σ^+ indicates the set of all INDS that are logical consequences of Σ with respect to finite databases. Given mutually disjoint classes of dependencies, C_1, \dots, C_n , a separability condition states when sets $C_1 \subseteq C_1^+, \dots, C_n \subseteq C_n^+$, $1 \leq i \leq n$, are such that $(C_1 \cup \dots \cup C_n)^+ = \bigcup_{i=1}^n C_i^+$. We first state a separability condition for INDS and FDS (Appendix B contains an additional definition and a lemma that we need to prove the results in this section).

THEOREM 5 3 Let Σ be a set of INDs and Γ be a set of FDs. Suppose that, if $R[X] \subseteq S[Y] \in \Sigma$ and if $S \xrightarrow{W} Z \in \Gamma$, then we have that $Y = W$. Then, $(\Sigma \cup \Gamma)^+ = \Sigma^+ \cup \Gamma^+$ \square

When we also consider EXDs, we can extend the results in Theorem 5 3 to a near separability condition

THEOREM 5 4 Let Σ be a set of INDs, Δ be a set of FDs and Γ be a set of EXDs. Suppose that, if $R[X] \subseteq S[Y] \in \Sigma$ and if $S \xrightarrow{W} Z \in \Gamma$, then $Y = W$. Then, we have

- (i) for any EXD δ , $\Sigma \cup \Gamma \cup \Delta \models_f \delta$ iff $\Sigma \cup \Delta \models_f \delta$
- (ii) for any IND σ , $\Sigma \cup \Gamma \cup \Delta \models_f \sigma$ iff either $\Sigma \models_f \sigma$ or there is a vacuous EXD δ such that $\Sigma \cup \Delta \models_f \delta$ and $\delta \models_f \sigma$
- (iii) for any FD γ , $\Sigma \cup \Gamma \cup \Delta \models_f \gamma$ iff either $\Gamma \models_f \gamma$ or there is a vacuous EXD δ such that $\Sigma \cup \Delta \models_f \delta$ and $\delta \models_f \gamma$
- (iv) for any UFD ψ of the form $\langle R_{1_1} X_1 \rightarrow Y_1, \dots, R_{1_m} X_m \rightarrow Y_m \rangle$ we have that $\Sigma \cup \Gamma \cup \Delta \models_f \psi$ iff
 - (a) $\Gamma \models_f R_{1_j} X_j \rightarrow Y_j$ or $\Sigma \cup \Delta$ implies a vacuous EXD over R_{1_j} , for any $j \in [1, m]$,
 - (b) $\Sigma \cup \Delta \models_f R_{1_j} [X_j] | R_{1_k} [X_k]$, for any $j, k \in [1, m]$ with $j \neq k$ \square

This last theorem gives us enough understanding about FDs, UFDs, INDs and EXDs to prove that the optimization procedure achieves the goals set at the end of Section 3

THEOREM 5 5 Let $R' = (S', C', K')$ be the schema produced by the optimization procedure when the input is a restricted schema $R = (S, C, K)$. Then, we have

- (i) C' implies no vacuous EXDs,
- (ii) all INDs implied by C' are of the form $R'_1[X] \subseteq R'_j[X]$, where $X \subseteq K'_j$,
- (iii) C' implies no INDs of the form $R'_1[K'_j] \subseteq R'_j[K'_j]$ and $R'_j[K'_j] \subseteq R'_1[K'_j]$ at the same time,
- (iv) if C' implies a UFD $\langle R_{1_1} X_1 \rightarrow Y_1, \dots, R_{1_m} X_m \rightarrow Y_m \rangle$ then C' also implies $R_{1_j}[X_j] | R_{1_k}[X_k]$ for any $j, k \in [1, m]$ with $j \neq k$ \square

This last theorem completes our study of the optimization procedure and the results we wanted to present about dependencies

6 CONCLUSIONS AND DIRECTIONS FOR FUTURE RESEARCH

We have shown how certain classes of dependencies can be used to place view integration on solid foundations. This was accomplished by dividing view integration into view combination and optimization. View combination uses dependencies to capture associations between different views. The optimization step tries to reduce redundancy and the size of the schema. We have also shown that, under certain circumstances, the optimization step can be automated.

Our ability to construct optimization procedures is largely limited by two problems: the complexity of the inference problem for INDs and the tight interaction between INDs, EXDs, FDs and UFDs. Both problems have to be overcome in the context of the design methodology in question. That is, one should seek easily solvable special cases of the decision problem for these dependencies that help the design method. Of special interest are separability conditions since all logical consequences of the constraints defined in a schema should be easy to anticipate.

Further work should be directed towards extending the optimization procedure given in Section 4 to more general schemas, perhaps using other types of inter-relational constraints. The software engineering aspects of view integration should also be stressed, since the combination step depends entirely on dictionary facilities to help locate redundancies.

ACKNOWLEDGEMENTS

Research support from FINEP and from CNPq, grant 40 1963/81, is gratefully acknowledged.

APPENDIX A

AN EXAMPLE

(a) Suppose that different users observe sets of students that, after uniformization of nomenclature can be represented by four relation schemes

- (1) ST[SN,NAME] - students
- (2) UST[SN,UNAME,MAJOR] - undergraduates
- (3) GST[SN,GNAME,ADVISOR,GDEPT] - graduate stud
- (4) TA[SN,TNAME,TDEPT] - teaching assist

In the combination step, it is detected that UST, GST and TA define subsets of ST and that UST and GST define disjoint sets This can be expressed as

- (5) $UST[SN] \subseteq ST[SN]$, $GST[SN] \subseteq ST[SN]$, $TA[SN] \subseteq ST[SN]$
- (6) $UST[SN] \mid GST[SN]$

It is also detected that NAME, UNAME, GNAME and TNAME are synonyms, as well as GDEPT and TDEPT. So, we must have

- (7) $\langle ST \text{ SN} \rightarrow \text{NAME}, UST \text{ SN} \rightarrow \text{UNAME}, GST \text{ SN} \rightarrow \text{GNAME}, TA \text{ SN} \rightarrow \text{TNAME} \rangle$
- (8) $\langle GST \text{ SN} \rightarrow \text{GDEPT}, TA \text{ SN} \rightarrow \text{TDEPT} \rangle$

The result of the combination step is a schema $\sigma = (S,C)$, where S consists of the relation schemes (1) through (4) and C consists of the dependencies in (5) to (8), plus FDs defining that SN is the key of each of the four schemes

(b) We now observe that the INDs in (5) and the UFD in (7) imply that

- (9) $UST[SN,UNAME] \subseteq ST[SN,NAME]$
- (10) $GST[SN,GNAME] \subseteq ST[SN,NAME]$
- (11) $TA[SN,TNAME] \subseteq ST[SN,NAME]$

Since the above dependencies indicate that redundancies exist in σ , the optimization procedure deletes the UFD in (7) and transforms UST, GST and TA to

- (12) UST'[SN,MAJOR], GST'[SN,ADVISOR,GDEPT], TA'[SN,TDEPT]

Note that the attributes deleted can be recovered from ST The optimization procedure eliminates the UFD in (8) by creating a new relation scheme

- (13) SD'[SN,DEPT]

by deleting GDEPT from GST' and TDEPT from TA'

- (14) GST''[SN,ADVISOR], TA''[SN]

and by adding two new INDs

- (15) $GST''[SN] \subseteq SD'[SN]$, $TA''[SN] \subseteq SD'[SN]$

The initial schema $\sigma = (S,C)$ is then transformed into $\sigma' = (S',C')$ where

- (16) $S' = \{ST'[SN,NAME], SD'[SN,DEPT], UST'[SN,MAJOR], GST''[SN,ADVISOR], TA''[SN]\}$
- (17) $C' = \{UST'[SN] \subseteq ST'[SN], GST''[SN] \subseteq ST'[SN], TA''[SN] \subseteq ST'[SN], TA''[SN] \subseteq SD'[SN], GST''[SN] \subseteq SD'[SN], UST'[SN] \mid GST''[SN], GST'' \text{ SN} \rightarrow \text{ADVISOR}, UST' \text{ SN} \rightarrow \text{MAJOR}, ST' \text{ SN} \rightarrow \text{NAME}, SD' \text{ SN} \rightarrow \text{DEPT}\}$

(c) Finally, we observe that the relation schemes in σ can be mapped into those of σ' , and vice-versa by the mapping below

- (18) ST = ST'
- TA = TA''*ST'*ST'
- GST = GST''*ST'*SD'
- UST = UST'*ST'
- (19) ST' = ST
- TA'' = TA[SN]
- GST'' = GST[SN,ADVISOR]
- UST' = UST[SN,MAJOR]
- SD' = TA[SN,TDEPT] \cup GST[SN,GDEPT]

APPENDIX B

PROOFS OF SELECTED THEOREMS

Proof of Theorem 5.1

Let $S = \{R_1[U_1], \dots, R_n[U_n]\}$ be a set of relation schemes Let Σ be a set of INDs satisfying the conditions of the theorem Let σ be an IND. If σ is trivial, the result follows directly So assume that σ is not trivial Let A be the axiom system of [CFP] (rules I1,I2,I3) Since A is sound and complete, it suffices to prove that σ is a theorem of Σ in A iff (i) and (ii) hold

(\Rightarrow) We prove, by induction on n , that if there is a proof of length n of σ from Σ , then (i) and (ii) hold

basis assume that there is a proof of length 1 of σ from Σ . Then, $\sigma \in \Sigma$ or σ is a trivial IND. The result then follows trivially

induction step assume that if there is a proof of length m , $m < k$, of γ from Σ , then (i) and (ii) hold. Let σ be such that there is a proof of length k of σ from Σ . Let $P = (P_1, \dots, P_k)$ be such a proof. Assume that σ is $R_1[X] \subseteq R_j[Y]$

case 1 P_k was obtained by transitivity (rule I3) from P_r and P_s , for some $r, s < k$

Then, there are proofs of length less than k of P_r from Σ and of P_s from Σ . By the induction hypothesis, and since P_k is $R_1[X] \subseteq R_j[Y]$, P_r and P_s must be of the form $R_1[X] \subseteq R_m[Z]$ and $R_m[Z] \subseteq R_j[Y]$, respectively, with $X = Z$ and $Z = Y$. Hence, $X = Y$. Moreover, there must be a path from R_1 to R_m in G_X and from R_m to R_j in G_X . Hence, there is a path from R_1 to R_j in G_X . This concludes the analysis of this case

case 2 P_k was obtained from P_r by permutation and projection (rule I2), for some $r < k$

Then, there is a proof of length less than k of P_r from Σ . By the assumptions of this case and since P_k is $R_1[X] \subseteq R_j[Y]$, P_r must be of the form $R_1[Z] \subseteq R_j[W]$, where X, Y are obtained by the same permutation and projection from Z and W , respectively. By the induction hypothesis, we must have that $Z = W$ and that there is a path from R_1 to R_j in G_Z . Therefore, $Z = W$ implies $X = Y$. Moreover, since $X \subseteq Z$, we have that G_Z is a subgraph of G_X . Hence, there is a path from R_1 to R_j in G_X . This concludes the case analysis and the induction

Therefore, if σ is a theorem of Σ then (i) and (ii) hold

(\Leftarrow) Let $\sigma = R_1[X] \subseteq R_j[X]$ and assume that there is a path from R_1 to R_j in G_X . Let $(R_{1_0}, \dots, R_{1_k})$ be such a path (with $1_0 = 1$ and $1_k = j$). Then, by construction of G_X , for each $m \in [0, k-1]$, there must be $R_{1_m}[X_m] \subseteq R_{1_{m+1}}[X_{m+1}] \in \Sigma$ such that $X \subseteq X_m$. Then, we can easily construct a proof in A of $R_1[X] \subseteq R_j[X]$ from Σ using these formulas to obtain $R_{1_m}[X] \subseteq R_{1_{m+1}}[X]$ by permutation and projection and then repeatedly

applying transitivity

This concludes the proof \square

DEFINITION 5.1 Let $S = \{R_1[U_1], \dots, R_n[U_n]\}$ be a set of relation schemes and Σ be a finite set of INDs over S . Let $B' = \{s_1, \dots, s_n\}$ be a finite database over S . Let $V = \{v_1/v_1, \dots, v_n/v_n\}$ be a tuple over $R_1[U_1]$, for $i=1, \dots, n$

The completion or chase of B' with respect to Σ and V is the finite database $B = \{r_1, \dots, r_n\}$ over S defined as follows

- initially let $r_i = s_i$, for $i = 1, \dots, n$
 - add tuples to B by repeatedly applying the following rule until no new tuples can be added
- rule (*) if $t \in r_i$ and there is an IND in Σ of the form $R_1[X] \subseteq R_j[Y]$ then if $t[X] \notin r_j[Y]$, add a new tuple u to r_j constructed as follows: $u[Y] = t[X]$ and $u[A] = v_j[A]$, for all other attributes A of R_j not occurring in X \square

LEMMA 5.1 Let B be the completion of B' with respect to Σ and V

(a) B satisfies Σ ,

(b) let Γ be a set of FDs and suppose that

if $R_1[X] \subseteq R_j[Y] \in \Sigma$ and $R_j W \rightarrow V \in \Gamma$ then $Y = W$

Then, B satisfies Γ iff B' also satisfies Γ

Proof

(a) Follows directly by construction of B

(b) Suppose that $B' = \{s_1, \dots, s_n\}$ and $B = \{r_1, \dots, r_n\}$. Then, by construction of B , $s_i \subseteq r_i$, for $i = 1, \dots, n$. Hence, we immediately have that if B satisfies Γ , B' also satisfies Γ . To prove the converse, suppose that B' satisfies Γ , but B does not satisfy Γ . Then, there is an FD $R_j W \rightarrow V$ which is not valid in B . That is, there are two tuples $t, u \in r_j$ such that $t[W] = u[W]$ but $t[V] \neq u[V]$. Since B' satisfies Γ , either t or u are not in s_j . Suppose that t is not in s_j . Therefore, t was introduced in B by rule (*). Let $R_1[X] \subseteq R_j[Y]$ be the IND in Σ used to introduce t . By the assumption on Σ and Γ , we then have $W = Y$. Therefore, $t[Y] = u[Y]$. But this contradicts the construction of B , since t was unnecessarily introduced in r_j . Hence, we may conclude that B satisfies all FDs in Γ , which was to be shown \square

Proof of Theorem 5 3

Let Σ be a set of INDs and Γ be a set of FDs. Let $S = \{R_1[U_1], \dots, R_n[U_n]\}$ be the set of schemes in question. Suppose that Σ and Γ satisfy the condition of the theorem. Clearly, $\Sigma^+ \cup \Gamma^+ \subseteq (\Sigma \cup \Gamma)^+$. So, we only prove the converse. Let $\sigma \in (\Sigma \cup \Gamma)^+$.

case 1 assume that σ is an IND

We show that $\sigma \in \Sigma^+$ or, equivalently, that there is a proof of σ from Σ in A , where A is the axiom system for INDs of [CFP] (rules I1, I2, I3 of Figure 5 1). Suppose that σ is $R_a[A_1 \dots A_m] \subseteq R_b[B_1 \dots B_m]$. Let $V = \{v_1, \dots, v_n\}$ be a set of tuples such that v_1 is over $R_1[U_1]$ and $v_1 = (0, \dots, 0)$ for $i = 1, \dots, n$. Let $B' = \{s_1, \dots, s_n\}$ be the database over S constructed as follows

- let t be a tuple over the attributes of R_a such that $t[A_1] = 1$, for $i = 1, \dots, m$, and $t[A] = 0$, for each remaining attribute A of R_a . Let $s_a = \{t\}$ and $s_j = \emptyset$, for each remaining j .

Let B be the completion of B' with respect to Σ and V . Then, by Lemma 5 1, B satisfies $\Sigma \cup \Gamma$. Since, by assumption, $\sigma \in (\Sigma \cup \Gamma)^+$, the database B satisfies σ . That is, B satisfies the IND

$R_a[A_1 \dots A_m] \subseteq R_b[B_1 \dots B_m]$. But since r_a contains the tuple t , r_b contains a tuple t' , where $t'[B_1] = 1$, for $i = 1, \dots, m$. We can prove that claim (*) if r_j contains a tuple u such that

$u[E_p] = 1_p$, where $1_p \in \{1, m\}$, for each $p \in \{1, k\}$, then $R_a[A_{1_1} \dots A_{1_k}] \subseteq R_j[E_1 \dots E_k]$ is a theorem

of Σ in A .

From claim (*) and since $t' \in r_b$, it follows that $R_a[A_1 \dots A_m] \subseteq R_b[B_1 \dots B_m]$ is a theorem of Σ in A . Finally, since A is sound, we have that $\sigma \in \Sigma^+$, which was to be shown.

case 2 assume that σ is an FD

We show that if $\sigma \notin \Gamma^+$ then $\sigma \notin (\Sigma \cup \Gamma)^+$, which contradicts the assumption that $\sigma \in (\Sigma \cup \Gamma)^+$.

Assume that $\sigma \notin \Gamma^+$ and that σ is of the form $R_a[X \rightarrow Y]$. We may suppose without loss of generality that Y contains a single attribute A . Thus, since $\sigma \notin \Gamma^+$, we have that $A \notin \text{DEP}(X)$, where $\text{DEP}(X)$ is the dependency basis of X with respect to all FDs in Γ over R_a . Let $V = \{v_1, \dots, v_n\}$ be a set of tuples such that v_1 is over $R_1[U_1]$ and $v_1 = (0, \dots, 0)$ for $i = 1, \dots, n$. Construct a database state

$B' = \{s_1, \dots, s_n\}$ of S as follows

- construct two tuples t and u as follows
 - . $t[A] = 1$, if $A \in \text{DEP}(X)$
 - . $t[A] = 0$, if $A \in U_a - \text{DEP}(X)$
 - $u[A] = 0$, for all $A \in U_a$
- let $s_a = \{t, u\}$ and $s_1 = \emptyset$, for all other relations

Let B be the completion of B' with respect to Σ and V . Then, since B' satisfies Γ , we know by Lemma 5 1 that B satisfies $\Sigma \cup \Gamma$. But since $A \notin \text{DEP}(X)$, by construction of t and u , B does not satisfy σ . Hence, we have that $\sigma \notin (\Sigma \cup \Gamma)^+$, which was to be shown.

This concludes the proof. \square

Proof of Theorem 5 5

We first observe that R' is a restricted schema. Moreover, by the test on line 24 and the transformation on line 25, C' does not imply any vacuous EXD. Now, the dependencies in C' satisfy the conditions of Theorem 5 4. Therefore, since C' implies no vacuous EXD, we then have that

- (a) for any IND σ , $C' \models_f \sigma$ iff $\Sigma \models_f \sigma$,
- (b) for any UFD ψ of the form

$$\langle R_{1_1} X_1 \rightarrow Y_1, \dots, R_{1_m} X_m \rightarrow Y_m \rangle$$

we have that $C' \models_f \psi$ iff

$$C' \models_f R_{1_j} [X_j] \mid R_{1_k} [X_k] \text{ and } C' \models_f R_{1_j} X_j \rightarrow Y_j \text{ for any } j, k \in \{1, m\} \text{ with } j \neq k$$

Thus, using (a), Theorem 5 1 and the fact that R' is a restricted schema, we can show that C' satisfies conditions (11) and (111) (to prove (111) we have to use the transformation done on lines 10 to 12, which is not affected by the rest of the procedure). Finally, condition (1v) follows directly from (b).

This concludes the proof. \square

REFERENCES

[Ar] W W Armstrong, "Dependency Structures of Database Relationships", Proc IFIP 74 (1974) 580-583
 [BBG] C Beerl, P A Bernstein, N Goodman, "A Sophisticates' Introduction to Database

- Normalization Theory", Proc 4th Int'l Conf
on Very Large Data Bases (1978),113-124
- [Ca] M A Casanova, "A Theory of Data Dependencies
over Relational Expressions", Proc ACM
SIGMOD/SIGACT Conf on Principles of Database
Systems (1982)
- [CFP] M A Casanova, R Fagin, C H Papadimitriou,
"Inclusion Dependencies and their Inter-
action with Functional Dependencies",
Proc ACM SIGMOD/SIGACT Conf on Principles
of Database Systems (1982)
- [Ch] P Chen, "The Entity-Relationship Model -
Towards a Unified View of Data", ACM Trans
on Database Systems 1,1 (1976),9-36
- [Da1] C J Date, "Referential Integrity", Proc
7th Int'l Conf on Very Large Data Bases
(1981),2-12
- [Da2] C J Date, "An Introduction to Database
Systems", (3rd Ed), Addison-Wesley Pub Co
(1981)
- [K1] A Klug "Calculating Constraints on Relational
Expressions", ACM Trans on Database Systems
5,3 (1980)
- [Li] S H Lin, "Existential Dependencies in
Relational Databases", Ph D Thesis, UCLA
(1981)
- [NG] S B Navathe, S G Gadgil, "A Methodology for
View Integration in Logical Database Design",
Proc 8th Int'l Conf on Very Large Data Bases
(1982)
- [TF] T J Teorey, J P Fry, "Design of Database
Structures", Prentice-Hall, Inc (1982)
- [U1] J D Ullman, "Principles of Database Systems",
Computer Science Press (1979)
- [WM] G Wiederhold, R El-Masri, "A Structural Model
for Database Systems", TR STAN-CS-79-722,
Stanford University (Feb 1979)