Guaranteeing No Interaction Between Functional Dependencies and Tree-Like Inclusion Dependencies

Mark Levene
Department of Computer Science
University College London
Gower Street
London WC1E 6BT, U.K.
Email: mlevene@cs.ucl.ac.uk

George Loizou
Department of Computer Science
Birkbeck College
Malet Street
London WC1E 7HX, U.K.
Email: george@dcs.bbk.ac.uk

Abstract

Functional dependencies (FDs) and inclusion dependencies (INDs) are the most fundamental integrity constraints that arise in practice in relational databases. A given set of FDs does not interact with a given set of INDs if logical implication of any FD can be determined solely by the given set of FDs, and logical implication of any IND can be determined solely by the given set of INDs. The set of tree-like INDs constitutes a useful subclass of INDs whose implication problem is polynomial time decidable. We exhibit a necessary and sufficient condition for a set of FDs and tree-like INDs *not* to interact; this condition can be tested in polynomial time.

Keywords: Relation; Database; Logical implication; Functional dependency; Inclusion dependency

1 Introduction

The *implication problem* for FDs and INDs is the problem of deciding for a given set Σ of FDs and INDs whether Σ logically implies σ, where σ is an FD or an IND. The implication problem is central in data dependency theory and is also utilised in the process of database design, since it can be used to test whether two sets of dependencies are equivalent or to detect whether a dependency in a given set is redundant. The implication problem for FDs and INDs is known to be undecidable in the general case [Mit83, CV85] and can be decided only in exponential time when the INDs are restricted to be noncircular [CK86] or proper circular [Imi91]. On the other hand, the implication problem for FDs on their own is known to be decidable in linear time [BB79] and the implication problem for noncircular or proper circular INDs, again on their own, is known to be NP-complete [MR92] (for INDs, which may be circular, the implication problem is PSPACE-complete [CFP84]). Thus given a set Σ of FDs and INDs and an FD or IND σ, it would be desirable if the set F of FDs and the set I of INDs do *not interact*, in the sense that the implication problem of whether Σ logically implies σ can be decided by F on its own, when σ is an FD, and by I on its own, when σ is an IND. That is, if F and I do not interact then the algorithms in database design that use logical implication can be implemented more efficiently than would otherwise be the case (see [LV00]). The impact of such lack of interaction would be the greatest if the implication problem for the subclass of INDs under consideration is polynomial time decidable. It has been a long standing open problem in relational database theory to characterise useful subclasses of FDs and INDs that do not interact.

We partially solve this open problem by exhibiting a necessary and sufficient condition for no interaction between a set of FDs and a set of INDs which is *tree-like* [MR88]. Moreover, this condition can be tested in polynomial time. The implication problem for tree-like INDs, which is a proper subclass of the subclass of noncircular INDs, is polynomial time decidable. Although the subclass of tree-like INDs is somewhat restricted it covers many practical situations when inclusion dependencies are used to enforce referential integrity [Dat86].

The layout of the rest of this note is as follows. In Section 2 we briefly define the underlying concepts from relational database theory. In Section 3 we introduce the subclass of tree-like INDs and reduced set of FDs and INDs. In Section 4 we present our main result which states that a set of FDs and tree-like INDs do not interact if and only if such a set is reduced. Finally, in Section 5 we give our concluding remarks.

2 Functional and inclusion dependencies

Herein we present the preliminary concepts from relational database theory [MR92, LL99a] which are needed to obtain our results. We use the notation |S| to denote the cardinality of a set S. We often denote the singleton {A} simply by A, and the union of two sets S, T, i.e. S $\cup$ T, simply by ST.

Definition 2.1 (Database schema and database) Let $\mathcal{U}$ be a finite set of attributes. A *relation schema* R is a finite sequence of distinct attributes from $\mathcal{U}$. A *database schema* is a finite set $\mathbf{R} = \{R_1, \ldots, R_n\}$, such that each $R_i \in \mathbf{R}$ is a relation schema and $\bigcup_i R_i = \mathcal{U}$.

We assume a countably infinite domain of values $\mathcal{D}$; without loss of generality, we assume that $\mathcal{D}$ is linearly ordered. An R-tuple (or simply a tuple whenever R is understood from context) is a member of the Cartesian product $\mathcal{D} \times \ldots \times \mathcal{D}$ (|R| times).

A *relation* r over R is a finite (possibly empty) set of R-tuples. A *database* d over $\mathbf{R}$ is a family of n relations $\{r_1, \ldots, r_n\}$ such that each $r_i \in d$ is over $R_i \in \mathbf{R}$.

From now on we let $\mathbf{R}$ be a database schema and d be a database over $\mathbf{R}$. Furthermore, we let $r \in d$ be a relation over the relation schema R $\in \mathbf{R}$.

Definition 2.2 (Projection) The *projection* of an R-tuple t onto a set of attributes Y $\subseteq$ R, denoted by t[Y] (also called the Y-value of t), is the restriction of t to Y, maintaining the order of Y. The projection of a relation r onto Y, denoted as $\pi_{\mathrm{Y}}(r)$, is defined by $\pi_{\mathrm{Y}}(r) = \{t[\mathrm{Y}] \mid t \in r\}$.

Definition 2.3 (Functional Dependency) A *functional dependency* (or simply an FD) over a database schema $\mathbf{R}$ is a statement of the form R:X $\rightarrow$ Y (or simply X $\rightarrow$ Y whenever R is understood from context), where R $\in \mathbf{R}$ and X, Y $\subseteq$ R are sets of attributes. An FD of the form R:X $\rightarrow$ Y is said to be *trivial* if Y $\subseteq$ X.

An FD R:X $\rightarrow$ Y is satisfied in d, denoted by $d \models$ R:X $\rightarrow$ Y, whenever $\forall t_1, t_2 \in r$, if t_1[X] = t_2[X] then t_1[Y] = t_2[Y].

Definition 2.4 (Inclusion Dependency) An *inclusion dependency* (or simply an IND) over a database schema $\mathbf{R}$ is a statement of the form $R_i[\mathrm{X}] \subseteq R_j[\mathrm{Y}]$, where $R_i, R_j \in \mathbf{R}$ and $\mathrm{X} \subseteq R_i$, $\mathrm{Y} \subseteq R_j$ are sequences of distinct attributes such that $|\mathrm{X}| = |\mathrm{Y}|$. An IND is said to be *trivial* if it is of the form $\mathrm{R}[\mathrm{X}] \subseteq \mathrm{R}[\mathrm{X}]$.

An IND $R_i[\mathrm{X}] \subseteq R_j[\mathrm{Y}]$ over $\mathbf{R}$ is satisfied in d, denoted by $\mathrm{d} \models R_i[\mathrm{X}] \subseteq R_j[\mathrm{Y}]$, whenever $\pi_{\mathrm{X}}(r_i) \subseteq \pi_{\mathrm{Y}}(r_j)$, where $r_i, r_j \in d$ are the relations over R_i and R_j, respectively.

In the sequel we let F be a set of FDs over $\mathbf{R}$ and $F_i = \{R_i : \mathrm{X} \rightarrow \mathrm{Y} \in \mathrm{F}\}$, $i \in \{1, \ldots, n\}$, be the set of FDs in F over $R_i \in \mathbf{R}$. Furthermore, we let I be a set of INDs over $\mathbf{R}$ and let $\Sigma = \mathrm{F} \cup \mathrm{I}$.

Definition 2.5 (Logical implication) Σ is *satisfied* in d, denoted by $d \models \Sigma$, if $\forall \sigma \in \Sigma, d \models \sigma$.

Σ *logically implies* an FD or an IND σ, written $\Sigma \models \sigma$, if whenever d is a database over $\mathbf{R}$ then the following condition is true:

$$\text{if } d \models \Sigma \text{ holds then } d \models \sigma \text{ also holds.}$$

Σ *logically implies* a set Γ of FDs and INDs over $\mathbf{R}$, written $\Sigma \models \Gamma$, if $\forall \sigma \in \Gamma, \Sigma \models \sigma$. We let Σ^+, called the *closure* of Σ, denote the set of all FDs and INDs that are logically implied by Σ.

The *closure* of a set of attributes $\mathrm{X} \subseteq R_i$ with respect to F_i, denoted by $C_i(\mathrm{X})$, is the set of attributes $\{\mathrm{A} \mid \mathrm{X} \rightarrow \mathrm{A} \in {F_i}^+\}$.

The next well-known result follows from Theorem 9.2 in Chapter 9 of [MR92].

Lemma 2.1 Let F_i be a set of FDs over R_i and R_i:$\mathrm{X} \rightarrow \mathrm{Y}$ be an FD such that $\mathrm{Y} \not\subseteq C_i(\mathrm{X})$. Moreover, let $r_i \in d$ over R_i be a relation containing two tuples, t_1 and t_2 such that for all $\mathrm{A} \in R_i$, $t_1[\mathrm{A}] = 0$, for all $\mathrm{A} \in C_i(\mathrm{X})$, $t_2[\mathrm{A}] = 0$ and for all $\mathrm{A} \in R_i - C_i(\mathrm{X})$, $t_2[\mathrm{A}] = 1$. Then $d \models F_i$ but $d \not\models R_i$:$\mathrm{X} \rightarrow \mathrm{Y}$. □

The *pullback* inference rule for FDs and INDs [Mit83, CFP84], which is utilised below, is stated in the ensuing proposition.

Proposition 2.2 If $\Sigma \models \{\mathrm{R}[\mathrm{XY}] \subseteq \mathrm{S}[\mathrm{WZ}], \mathrm{S}{:}\mathrm{W} \rightarrow \mathrm{Z}\}$ and $|\mathrm{X}|{=}|\mathrm{W}|$, then $\Sigma \models \mathrm{R}{:}\mathrm{X} \rightarrow \mathrm{Y}$. □

The chase procedure provides us with a very useful algorithm which forces a database to satisfy a set of FDs and INDs.

Definition 2.6 (The chase procedure for INDs) The chase of d with respect to Σ, denoted by CHASE(d, Σ), is the result of applying the following chase rules, namely the FD and the IND rules, to the current state of d as long as possible. (The current state of d prior to the first application of either of the chase rules is its state upon input to the chase procedure.)

FD rule: If R_j:$\mathrm{X} \rightarrow \mathrm{Y} \in F_j$ and $\exists t_1, t_2 \in r_j$ such that $t_1[\mathrm{X}] = t_2[\mathrm{X}]$ but $t_1[\mathrm{Y}] \neq t_2[\mathrm{Y}]$, then $\forall \mathrm{A} \in \mathrm{Y}$, change all the occurrences in d of the larger of the values of $t_1[\mathrm{A}]$ and $t_2[\mathrm{A}]$ to the smaller of the values of $t_1[\mathrm{A}]$ and $t_2[\mathrm{A}]$.

IND rule: If R_i[X] $\subseteq$ R_j[Y] $\in$ I and $\exists t \in r_i$ such that t[X] $\notin \pi_{\mathrm{Y}}(r_j)$, then add a tuple u over R_j to r_j, where u[Y] $= t$[X] and $\forall$A $\in R_j-$Y, u[A] is assigned a new value greater than any other current value occurring in the tuples of the relations in the current state of d.

Often we refer to an application of the FD rule or IND rule during the computation of the chase as a *chase step*.

We observe that there is no loss of generality to consider an FD rule for R_j : X $\rightarrow$ Y as an FD rule for the FDs R_j : X $\rightarrow$ A, with A $\in$ Y$-$X such that t_1[A] $\neq t_2$[A]. We will utilise this observation in proofs which use the chase procedure. We also observe that, in general, the chase procedure in the presence of INDs does not always terminate [JK84]. However, for some special subclasses of INDs the chase always terminates; see Theorem 3.39 in [LL99a].

3 The Subclass of Tree-Like INDs and Reduced FDs and INDs

Herein we define the subclass of tree-like INDs and show that the implication problem for this subclass can be decided in polynomial time. We also introduce the notion of a set of FDs and INDs being reduced and show that this condition can be tested in polynomial time.

Definition 3.1 (Graph representation of INDs) The graph representation of a set of INDs I over **R** is a directed graph G_I = (N, E), which is constructed as follows. Each relation schema R in **R** has a separate node in N labelled by R; we do not distinguish between nodes and their labels. There is an arc (R, S) $\in$ E if and only if there is a nontrivial IND R[X] $\subseteq$ S[Y] $\in$ I.

Definition 3.2 (Tree-like INDs) A set I of INDs over **R** is *tree-like* if

1) for all R, S $\in$ **R**, there is at most one nontrivial IND in I of the form R[X] $\subseteq$ S[Y], and

2) G_I is a *forest*, i.e. its maximally connected subgraphs (or components) are *rooted trees* (or simply trees).

The above definition essentially excludes any subclasses of INDs inducing a cyclic graph G_I, ignoring the direction of its arcs.

The following theorem is a consequence of results in Chapter 10 of [MR92]. It shows that when I is a set of tree-like INDs, then the chase procedure terminates and satisfies Σ. It also shows that in this case the chase can be decoupled into two distinct stages. At the first stage the IND rule is applied to the current state of d exhaustively and at the second stage the FD rule is applied exhaustively to the current state of d, after the first stage has been computed, terminating with the final result.

Theorem 3.1 Let Σ = F $\cup$ I be a set of FDs and tree-like INDs over a database schema **R**. Then the following three statements are true:

(i) CHASE$(d, \Sigma) \models \Sigma$.

(ii) CHASE(d, Σ) is identical to CHASE(CHASE(d, I), F) up to renaming of new values.

(iii) CHASE(d, Σ) terminates after a finite number of applications of the IND and FD rules to the current state of d. □

The next result shows that the implication problem for tree-like INDs can be solved in polynomial time.

Proposition 3.2 Given a set I of tree-like INDs over **R** and an IND R[X] $\subseteq$ S[Y] over **R**, it can be decided in polynomial time in the size of I whether I $\models$ R[X] $\subseteq$ S[Y].

Proof. Let d be a database, where apart from r over R all the relations in d are empty, and let r over R contain a single tuple t such that for all distinct attributes A, B $\in$ R, t[A] and t[B] are pairwise distinct values. From the results in [CFP84] and Chapter 10 in [MR92] we have that I $\models$ R[X] $\subseteq$ S[Y] if and only if CHASE(d, I) $\models$ R[X] $\subseteq$ S[Y]. Thus I $\models$ R[X] $\subseteq$ S[Y] if and only if there is a tuple $u \in s$, where s is the relation in d over S, such that u[Y] $= t$[X]. It remains to show that the number of chase steps, say k, required to compute CHASE(d, I) is polynomial in the size of I. However, this easily follows from Definition 3.2, since this definition implies that $k \leq |\text{I}|$. □

Definition 3.3 (Reduced set of FDs and INDs) The projection of a set of FDs F_i over R_i onto a set of attributes Y $\subseteq R_i$, denoted by F_i[Y], is given by $F_i[\text{Y}] = \{R_i{:}\text{W} \to \text{Z} \mid R_i{:}\text{W} \to \text{Z} \in F_i^+$ and WZ $\subseteq$ Y$\}$.

A set of attributes Y $\subseteq R_i$ is said to be *reduced* with respect to R_i and a set of FDs F_i over R_i (or simply reduced with respect to F_i if R_i is understood from context) if F_i[Y] contains only trivial FDs. A set of FDs and INDs Σ = F $\cup$ I is said to be *reduced* if $\forall R_i[\text{X}] \subseteq R_j[\text{Y}] \in$ I, Y is reduced with respect to F_j.

An example of the usefulness of a reduced set I of INDs is the case when I is *key-based*, i.e. Y is a key for R_j with respect to F_j [LL99b].

The following result is also utilised in [LL99b]; for completeness we give its proof.

Proposition 3.3 It can be decided in polynomial time in the size of Σ whether Σ is reduced or not.

Proof. The condition that Y is reduced with respect to F_j is true if and only if $\forall$A $\in$ Y, (Y$-$A) $\to$ A $\notin F_j^+$. The result now follows, since (Y$-$A) $\to$ A $\notin F_j^+$ can be checked in polynomial time in the size of F_j [BB79]. □

4 Interaction between FDs and INDs

Herein we introduce the notion of no interaction between a set of FDs and a set of INDs and prove our main result.

Definition 4.1 (Interaction between FDs and INDs) A set of FDs F over **R** is said *not to interact* with of set of INDs I over **R**, if

1) for all FDs α over **R**, for all subsets G $\subseteq$ F, G $\cup$ I $\models \alpha$ if and only if G $\models \alpha$, and

2) for all INDs β over **R**, for all subsets J $\subseteq$ I, F $\cup$ J $\models \beta$ if and only if J $\models \beta$.

Theorem 4.1 Let Σ = F $\cup$ I be a set of FDs and tree-like INDs over **R**. Then F and I do not interact if and only if Σ is reduced.

Proof. If. There are two cases to consider.

Case 1. Let G $\subseteq$ F be a set of FDs over **R** and let R_i:X $\rightarrow$ Y be an FD over R_i. We need to show that if G $\cup$ I $\models R_i$:X $\rightarrow$ Y, then G $\models R_i$:X $\rightarrow$ Y. Equivalently, we need to show that if G $\not\models R_i$:X $\rightarrow$ Y, then G $\cup$ I $\not\models R_i$:X $\rightarrow$ Y. That is, we need to exhibit a database, say d, such that $d \models$ G $\cup$ I but $d \not\models R_i$:X $\rightarrow$ Y.

Let d_0 be a database, where apart from r_i over R_i all the relations in d_0 are empty, and let r_i contain two tuples, t_1 and t_2, as in Lemma 2.1. Thus $d_0 \models$ G but $d_0 \not\models R_i$:X $\rightarrow$ Y. We inductively construct a database d' by a depth-first traversal of the subtree, say T, of G_I, whose root is R_i. If T is empty then $d' = d_0$. Otherwise, consider the next arc (R, R_j) in T and its corresponding IND R[W] $\subseteq R_j$[Z] $\in$ I. Without loss of generality we let R = R_i, since our construction is identical for all arcs in T. We add two tuples u_1 and u_2 to the relation $r_j \in d$ over R_j such that for all A $\in R_j$, u_1[A] = 0, u_2[Z] = t_2[W] and $u_2[R_j-$Z] is constructed as follows. Let V be the set of all attributes A $\in$ Z such that u_2[A] = 0. Moreover, for all A $\in$ R_j-Z, if A $\in C_j$(V) then we let u_2[A] = 0, otherwise we let u_2[A] = 1. Let this intermediate database be d_i. It follows that $d_i \models R_i$[W] $\subseteq R_j$[Z] and by Lemma 2.1 $d_i \models F_j$, since due to Σ being reduced C_j(V) $\cap$ Z$-$V = $\emptyset$. The result now follows, since by the construction of d', $d' \models$ G $\cup$ I but $d' \not\models R_i$:X $\rightarrow$ Y.

Case 2. Let J $\subseteq$ I be a set of INDs over **R** and R[X] $\subseteq$ S[Y] be an IND over **R**. We need to show that if F $\cup$ J $\models$ R[X] $\subseteq$ S[Y], then J $\models$ R[X] $\subseteq$ S[Y]. Equivalently, we need to show that if J $\not\models$ R[X] $\subseteq$ S[Y], then F $\cup$ J $\not\models$ R[X] $\subseteq$ S[Y]. That is, we need to exhibit a database, say d, such that $d \models$ F $\cup$ J but $d \not\models$ R[X] $\subseteq$ S[Y].

Let d_0 be a database, where apart from r over R all the relations in d_0 are empty, and let r contain a single tuple t such that for all distinct attributes A, B $\in$ R, t[A] and t[B] are pairwise distinct values. Let d_1 = CHASE(d, J). Then by the remark in the proof of Proposition 3.2 and the assumption that J $\not\models$ R[X] $\subseteq$ S[Y] we have that $d_1 \models$ J but $d_1 \not\models$ R[X] $\subseteq$ S[Y]. Moreover, due to the fact that I is tree-like all the relations in d_1 contain at most one tuple. Therefore, by part (ii) of Theorem 3.1 d_1 = CHASE(d_1, F $\cup$ J). The result follows, since $d_1 \models$ F $\cup$ J but $d_1 \not\models$ R[X] $\subseteq$ S[Y].

Only if. Assume that Σ is not reduced and thus for some IND $R_i[Z_i] \subseteq R_j[Z_j] \in$ I, Z_j is not reduced with respect to R_j and F_j. It now follows that $F_j[Z_j]$ contains a nontrivial FD, say $R_j : X_j \rightarrow Y_j$, with $X_jY_j \subseteq R_j$. Furthermore, we have that I $\models R_i[X_iY_i] \subseteq R_j[X_jY_j]$ for some subset $X_iY_i \subseteq Z_i$, with $|X_i|$=$|X_j|$, since $X_jY_j \subseteq Z_j$. Therefore, by Proposition 2.2, $\Sigma \models R_i : X_i \rightarrow Y_i$, where $R_i : X_i \rightarrow Y_i$ is a nontrivial FD. The result follows, since $F_j\cup$ I $\models$ $R_i : X_i \rightarrow Y_i$ but $F_j \not\models R_i : X_i \rightarrow Y_i$. $\Box$

5 Concluding Remarks

We have shown that a set Σ of FDs and tree-like INDs do not to interact if and only if Σ is reduced. This partially solves an open problem in database relational theory, namely to characterise no interaction between FDs and INDs for useful subclasses of such data dependencies whose implication problem is polynomial time testable. It is still an open problem to find a simple chararcterisation of the largest subclass of FDs and INDs that do not interact. More insight into the problem can be found in [LL99b], which deals with the larger subclasses of noncircular and proper circular INDs; therein it was shown that being reduced is not a sufficient condition for a set Σ of FDs and noncircular INDs not to interact.

Acknowledgement. The authors would like to thank the referee for his/her constructive comments.

References

[BB79] C. Beeri and P.A. Bernstein. Computational problems related to the design of normal form relational schemas. *ACM Transactions on Database Systems*, 4:30–59, 1979.

[CFP84] M.A. Casanova, R. Fagin, and C.H. Papadimitriou. Inclusion dependencies and their interaction with functional dependencies. *Journal of Computer and System Sciences*, 28:29–59, 1984.

[CK86] S.S. Cosmadakis and P.C. Kanellakis. Functional and inclusion dependencies: A graph theoretic approach. In P.C. Kanellakis and F. Preparata, editors, *Advances in Computing Research*, volume 3, pages 163–184. JAI Press, Greenwich, 1986.

[CV85] A.K. Chandra and M.Y. Vardi. The implication problem for functional and inclusion dependencies is undecidable. *SIAM Journal on Computing*, 14:671–677, 1985.

[Dat86] C.J. Date. Referential integrity. In *Relational Database: Selected Writings*, pages 41–63. Addison-Wesley, Reading, Ma., 1986.

[Imi91] T. Imielinski. Abstraction in query processing. *Journal of the ACM*, 38:534–558, 1991.

[JK84] D.S. Johnson and A. Klug. Testing containment of conjunctive queries under functional and inclusion dependencies. *Journal of Computer and System Sciences*, 28:167–189, 1984.

[LL99a] M. Levene and G. Loizou. *A Guided Tour of Relational Databases and Beyond*. Springer-Verlag, London, 1999.

[LL99b] M. Levene and G. Loizou. How to prevent interaction of functional and inclusion dependencies. *Information Processing Letters*, 71:115–125, 1999.

[LV00] M. Levene and M.W. Vincent. Justification for inclusion dependency normal form. *IEEE Transactions on Knowledge and Data Engineering*, 12, 2000. In press.

[Mit83] J.C. Mitchell. The implication problem for functional and inclusion dependencies. *Information and Control*, 56:154–173, 1983.

[MR88] H. Mannila and K.-J. Räihä. Generating Armstrong databases for sets of functional and inclusion dependencies. Research Report A-1988-7, Department of Computer Science, University of Tampere, Tampere, Finland, 1988.

[MR92] H. Mannila and K.-J. Räihä. *The Design of Relational Databases*. Addison-Wesley, Reading, Ma., 1992.