2
in press

A NOTE ON LOSSLESS DATABASE DECOMPOSITIONS

Moshe Y. Vardi[†]

IBM Research Laboratory

San Jose, CA 95193

Abstract

It is known that under a wide variety of assumptions a *database decomposition* is *lossless* if and only if the database scheme has a *lossless join*. Biskup, Dayal, and Bernstein have shown that when the given dependencies are functional then the database scheme has a lossless join if and only if one of the relation scheme is a key for the universal scheme. In this note we supply an alternative proof of that characterization. The proof uses tools from the theory of *embedded join dependencies* and the theory of *tuple* and *equality generating dependencies*, but is, nevertheless, much simpler than the previously published proof.

Keywords: database, decomposition, dependency, lossless join, relational model.

[†] The research reported here was done while this author was at Stanford University and supported by a Weizmann Post-Doctoral Fellewship and AFOSR grant 80-0212.

1. Introduction

A significant portion of research on relational database theory has been concerned with the properties of *decompositions.* The generic problem can be described as follows. Given a "universal" relation scheme presented as a set of attributes and a set of dependencies, what are the conditions under which it can be decomposed into a collection of relation schemes, each with its own sets of attributes and dependencies, having some desired properties. The properties considered were, at first, various *normal forms.* (see [Ma, Ul]).

A basic assumption underlying these ideas is that when a universal scheme is decomposed into smaller schemes, each of the universal relations associated with it is decomposed into smaller relations using the *projection* operation, i.e., each such relation is projected onto each one of the smaller schemes. For a decomposition to be useful, it should be *lossless.* In other words, it should be possible to reconstruct the universal relations from their projections. The desirability of this property is called in [BBG] the *representation principle.*

The property of losslessness has been studied in numerous papers [BR,MMSU,MUV,Va]. It has been shown there that, under a wide variety of assumptions, a decomposition is lossless if and only if the database scheme has a lossless join. When the given dependencies are functional the following characterization of database schemes that have lossless join is given in [BDB]: *a database scheme has a lossless join if and only if one of the relation schemes is a key of the universal scheme.*

The proof of the above characterization in [BDB] is quite involved and consists of a detailed study of the test for losslessness in [ABU]. In this note we describe an alternative proof for that characterization. The proof uses more advanced tools from dependency theory, but is, we believe, much simpler. Specifically, we use the theory of *embedded join dependencies* [BV1] and the theory of *tuple* and *equality generating dependencies* [BV2,BV3]. Our intention is to demonstrate how "higher-level" notions from dependency theory can be used to study its most fundamental questions. We believe that techniques used here could also be applicable to other problems.

2. The Theorem

We assume familiarity with the terminology and concepts of relational database theory as presented in [Ma, Ul]. The *universe* U is a finite set of *attributes*. All attribute sets are subsets of U. An *attribute set collection* (asc) is a set $\mathbf{R}=\{R_1,\ldots,R_k\}$ of distinct attribute sets. We denote attribute sets by lightface letters and asc's by boldface letters. A *database scheme* is an asc $\mathbf{R}=\{R_1,\ldots,R_k\}$ such that $\bigcup_{i=1}^{k} R_i=U$. A relation is a relation on U unless explicitly specified otherwise. We use $I[X]$ to denote the *projection* of the relation I on the attribute set X, and $\underset{j}{*}I_j$ to denote the *join* of the set $\{I_j\}$ of relations. We assume that the relations we are dealing with and, accordingly, the dependencies that refer to them are *typed*, that is, distinct attributes have disjoint domains. We also assume that all relations are finite.

A *functional dependency* (fd) is a statement $X\rightarrow Y$, where X and Y are attribute sets. It is satisfied by a relation I if for all tuples s and t in I, if $s[X]=t[X]$ then $s[Y]=t[Y]$. A *total join dependency* (tjd) is a statement $*[\mathbf{R}]$, where $\mathbf{R}$ is a database scheme $\{R_1,\ldots,R_k\}$. It is satisfied by a relation I if $I=\overset{k}{\underset{i=1}{*}}I[R_i]$.

Let Σ be a set of dependencies and let τ be a dependency. We say that Σ implies τ, denoted $\Sigma\models\tau$ if every relation on U that satisfies all dependencies in Σ satisfies also τ. A database scheme $\mathbf{R}$ has a *lossless join* with respect to Σ if $\Sigma\models *[\mathbf{R}]$. A set Σ of fd's is *embeddable* in a database scheme $\mathbf{R}$ if for each fd $X\rightarrow Y$ in Σ there is some R in $\mathbf{R}$ such that $XY\subseteq R$.[1]

Theorem. [BDB] Let Σ be a set of fd's and let $\mathbf{R}$ be a database scheme.

(1) If $\mathbf{R}$ has a lossless join with respect to Σ, then there is some $R\in\mathbf{R}$ such that $\Sigma\models R\rightarrow U$.

[1] Following convention we denote set union by juxtaposition.

(2) If Σ is embeddable in R and there is some $R \in$ R such that $\Sigma \models R \rightarrow U$, then R has a lossless join with respect to Σ. ■

3. The Necessary Condition

3.1. Preliminary Definitions

We prove the necessary condition via an excursion through the theory of tuple and equality generating dependencies. We need first to review some definitions.

A *valuation* is a mapping on the domain of which tuples and relations are constructed. We can extend the definition of a valuation to tuples, in a component-wise manner, and to relations, in a tuple-wise manner.

Equality generating dependencies generalize fd's. An equality generating dependency (egd) says that if some tuples, fulfilling certain equalities, exist in the database, then some values in these tuples must be equal. Formally, an egd is a pair $\langle (a_1,a_2),I \rangle$, where I is a relation and a_1 and a_2 occur in I. A relation J satisfies $\langle (a_1,a_2),I \rangle$ if for any valuation h such that $h(I) \in J$ we have $h(a_1)=h(a_2)$. Note that if $a_1=a_2$ then $\langle (a_1,a_2),I \rangle$ is trivially satisfied by every relation.

Lemma 1. [BV3] Let h be a valuation and let $\langle (a_1,a_2),I \rangle$ be an egd. Then $\langle (a_1,a_2),I \rangle \models \langle (h(a_1),h(a_2)),h(I) \rangle$. ■

Total tuple generating dependencies generalize tjd's. A total tuple generating dependency (ttgd) says that if some tuples, fulfilling certain equalities, exist in the database, then another tuple, whose values are taken from these tuples, must also exist in the database. Formally, a ttgd is a pair $\langle w,I \rangle$, where I is a relation and w is a tuple whose entries occur in I. A relation J satisfies $\langle w,I \rangle$ if for any valuation h such that $h(I) \subseteq J$, we have that $h(w) \in J$. Note that if $w \in I$, then $\langle w,I \rangle$ is trivially satisfied by every relation.

Fd's and tjd's can be viewed as a special case of egd's and ttgd's, respectively. Consider the fd $X \rightarrow A$, where $A \notin X$. We define an egd $\tau_{X \rightarrow A} = \langle (a0,a1),J \rangle$ as follows: J consists of two tuples u and v such that $u[X]=v[X]$, $u[B]\neq v[B]$ for all $B\notin X$, $u[A]=a0$, and $v[A]=a1$. We leave it to the reader to verify that a relation K satisfies $X\rightarrow A$ if and only if it satisfies $\tau_{X\rightarrow A}$. Consider the tjd $*[\mathbf{R}]$, where $\mathbf{R}=\{R_1,\ldots,R_k\}$. We define a ttgd $\tau_{\mathbf{R}}=\langle w,I\rangle$, where w is an arbitrary tuple, $I=\{w_1,\ldots,w_k\}$, $w_i[R_i]=w[R_i]$, and if $A\notin R_i$ then $w_i[A]$ has a unique occurrence in I. It is shown in [ASU] that a relation K satisfies $*[\mathbf{R}]$ if and only if it satisfies $\tau_{\mathbf{R}}$.

Example 1. Let $U=ABCD$. Let I and J be the relations:

I:

A	B	C	D
a0	b0	c1	d0
a0	b1	c0	d1

J:

A	B	C	D
a0	b0	c0	d0
a1	b0	c0	d1

Let u be the tuple:

A	B	C	D
a0	b0	c0	d0

Let τ_1 be the ttgd $\langle u,I\rangle$. Then τ_1 is equivalent to the tjd $*[ABD,AC]$. Let τ_2 be the egd $\langle (a0,a1),J\rangle$. Then τ_2 is equivalent to the fd $BC\rightarrow A$. ■

An algorithm for testing implication of egd's and ttgd's, called the *chase*, was presented in [BV2], following earlier algorithms for testing implications of fd's and tjd's [ABU,MMS]. We present here the special case where the implying dependencies are functional. Without loss of generality we assume that the given fd's have a single attribute on their right-hand side. A *chase* of a relation I *by* a set Σ of fd's is a maximal sequence of distinct relations $I_0, I_1, \cdots$ such that $I=I_0$ and I_{n+1} is obtained from I_n by an application of a *chase rule*. To each fd in Σ there corresponds a chase rule;

FD-rule (for an fd $X\rightarrow Y$ in Σ): I_{n+1} is obtained from I_n by identifying all occurrences of $u[A]$ with all occurrences of $v[A]$, for some tuples u and v in I_n such that $u[X]=v[X]$.

To make the FD-rule unambiguous, we assume that the domain of values for each attribute is totally ordered and whenever two values are identified, the greater is identified with the smaller. Given $\langle w,I\rangle$, we take $w(A)$ as the smallest value in the domain for A. (We can always rename the values in w and I so that this is true). Similarly, given $\langle(a_1,a_2),I\rangle$, we take a_1 as the smallest values in the domain for A and a_2 as the next smallest. Thus, the values in w or a_1 do not change in the chase and a_2 can be identified only with a_1. It is shown in [BV2] that all chases of I by Σ are finite and they have the same final relation, denoted $chase_\Sigma(I)$.

To trace the tuples of I in the chase we adjoin to each tuple an ordinal number, that is $I=\{w_1,\ldots,w_k\}$. The ordinal numbers do not change during the computation of the chase, though the values in the tuples may change by the FD-rules. Thus, I_n consists of the tuples $w_1^n,\ldots,w_k^n$ (not necessarily distinct), which correspond to the tuples $w_1,\ldots,w_k$ of I. We denote by w'_i the tuple in $chase_\Sigma(I)$ that correspond to w_i.

Lemma 2. [BV2] Let Σ be a set of fd's and let $I=\{w_1,\ldots,w_k\}$ be a relation.

(1) $\Sigma\models\langle w,I\rangle$ if and only if $w\in chase_\Sigma(I)$.

(2) $\Sigma\models\langle(w_i[A],w_j[A]),I\rangle$ if and only if $w'_i[A]=w'_j[A]$. ■

3.2. The Proof

Suppose that $\mathbf{R}=\{R_1,\ldots,R_k\}$ has a lossless join with respect to a set Σ of fd's. That is, $\Sigma\models{*}[\mathbf{R}]$. If $U\in\mathbf{R}$, then the claim of the theorem is trivially satisfied. So assume that $U\notin\mathbf{R}$. Let $\tau_{\mathbf{R}}=\langle w,I\rangle$. That is, $I=\{w_1,\ldots,w_k\}$, $w_i[R_i]=w[R_i]$, and if $A\notin R_i$ then $w_i[A]$ has a unique occurrence in I. Since $\Sigma\models{*}[\mathbf{R}]$, also $\Sigma\models\tau_{\mathbf{R}}$. By Lemma 2, there is some i such that $w'_i=w$. That is, $w'_i[A]=w[A]$ for all $A\notin R_i$. By Lemma 2 we have that $\Sigma\models\langle(w_i[A],w[A]),I\rangle$, for all $A\notin R_i$. Let $\tau_{R_i\to A}=\langle(a0,a1),J\rangle$. That is, J consists of two tuples u and v such that $u[R_i]=v[R_i]$, $u[B]\neq v[B]$ for all $B\notin R_i$, $u[A]=a0$, and $v[B]=a1$. We define a valuation h such that $h(w_i)=u$ and $h(w_j)=v$ for $j\neq i$. It is easy to see that h is well defined, $h(I)=J$, $h(w_i[A])=a0$ and $h(w[A])=a1$. By Lemma 1, $\langle(w_i[A],w[A]),I\rangle\models\tau_{R_i\to A}$. It follows that $\Sigma\models R_i\to A$ for all $A\notin R_i$, and, consequently, $\Sigma\models R_i\to U$.

4. The Sufficient Condition

4.1. Preliminary Notions

We prove the sufficient condition via an excursion through the theory of join dependencies. We need first to review some definitions.

For an asc $\mathbf{R}=\{R_1, \ldots, R_k\}$, the attribute set of $\mathbf{R}$, dennoted $attr(\mathbf{R})$, is $\bigcup_{i=1}^{k} R_i$. A *join dependency* (jd) is a statement $*[\mathbf{R}]$, where $\mathbf{R}$ is an asc $\{R_1, \ldots, R_k\}$. (We also use the notation $*[R_1, \ldots, R_k]$.) It is satisfied by a relation I if $I[attr(\mathbf{R})]=\mathop{*}_{i=1}^{k} I[R_i]$. Note that a tjd is a sepcial case of a jd. Jd's are called in [MMS] *embedded* jd's. Let $\mathbf{R}$ and $\mathbf{S}$ be asc's such that $attr(\mathbf{R})=attr(\mathbf{S})$. We say that $\mathbf{S}$ *covers* $\mathbf{R}$, denoted $\mathbf{R}\leq\mathbf{S}$, if for all $R\in\mathbf{R}$ there is some $S\in\mathbf{S}$ such that $R\subseteq S$.

Lemma 3. [BV1] Let X, Y, and Z be attribute sets, and let $\mathbf{R}$ and $\mathbf{S}$ be asc's. Then

(1) $\emptyset\models *[X]$.

(2) If $\mathbf{R}\leq\mathbf{S}$, then $*[\mathbf{R}]\models *[\mathbf{S}]$.

(3) If $attr(\mathbf{S})\in\mathbf{R}$, then $\{*[\mathbf{R}], *[\mathbf{S}]\}\models *[\mathbf{R}-\{attr(\mathbf{S})\}\cup\mathbf{S}]$.

(4) $X\rightarrow Y\models *[XY, XZ]$. ■

4.2. The Proof

It is shown in [BB] that if $\Sigma\models R\rightarrow U$, then there there is a sequence $X_1\rightarrow Y_1, \ldots, X_n\rightarrow Y_n$ of fd's from Σ with the following property: Let $R_0=R$ and $R_i=R_{i-1}Y_i$, $1\leq i\leq n$. Then $X_i\subseteq R_{i-1}$, for $1\leq i\leq n$, and $R_n=U$. We show by induction on i that $\Sigma\models *[R, X_1Y_1, \ldots, X_iY_i]$.

Basis ($i=0$). By (1) of Lemma 3, we have $\Sigma\models *[R]$.

Induction. Suppose that $\Sigma\models *[R, X_1Y_1, \ldots, X_{i-1}Y_{i-1}]$. Since $X_i\subseteq R_{i-1}$, we have that $X_i\rightarrow Y_i\models *[X_iY_i, R_{i-1}]$ by (4) of Lemma 3. Since $R\cup(\bigcup_{j=1}^{i-1} X_jY_j)=R_{i-1}$,

$\Sigma \models {}^*[R, X_1Y_1, \ldots, X_iY_i]$ by (3) of Lemma 4.

In particular we have $\Sigma \models {}^*[R, X_1Y_1, \ldots, X_nY_n]$. But by assumption $R \in \mathbf{R}$ and Σ is embeddable in R, so $\{R, X_1Y_1, \ldots, X_nY_n\} \leq \mathbf{R}$. It follows that $\Sigma \models {}^*[\mathbf{R}]$ by (2) of Lemma 4.

References

[ABU] Aho, A.V., Beeri, C., Ullman, J.D.: The theory of joins in relational databases. ACM Trans, on Database Systems 4(1979), pp.297-314.

[BB] Beeri, C., Bernstein P.A.: Computational problems related to the design of normal form relational schemas. ACM Trans. on Database Systems 4(1979), pp. 30-59.

[BBG] Beeri, C., Bernstein, P.A., Goodman, N.: A sophisticates' introduction to database normalization theory. Proc. Int'l Conf. on Very Large Databases, Berlin, 1978, pp. 113-124.

[BDB] Biskup, J., Dayal, U., Bernstein, P.A.: Synthesizing independent database schemas. Proc. ACM Int'l Conf. on Management of Data, Boston, 1979, pp. 143-151.

[BR] Beeri, C., Rissanen, J.: Faithful representation of relational database schemes. IBM Research Report, San Jose, 1980.

[BV1] Beeri, C., Vardi, M.Y.: On the properties of join dependencies. In *Advances in Database Theory* (H. Gallaire, J. Minker and J.M. Nicolas, eds.), Plenum, 1981, pp. 25-72.

[BV2] Beeri, C., Vardi, M.Y.: A proof procedure for data dependencies. Dept. of Computer Science, The Hebrew University of Jerusalem, Dec. 1980.

[BV3] Beeri, C., Vardi, M.Y.: Formal system for tuple and equality generating dependencies. Dept. of Computer Science, The Hebrew University of Jerusalem, April 1981. To appear in SIAM J. Computing.

[Ma] Maier, D., The Theory of Relational Databases, Computer Science Press, Rockville, Maryland, 1983.