An Equivalence Between Relational Database Dependencies and a Fragment of Propositional Logic

YEHOSHUA SAGIV

University of Illinois at Urbana-Champaign, Urbana, Illinois

CLAUDE DELOBEL

University of Grenoble, Grenoble, France

D. STOTT PARKER, JR.

University of California, Los Angeles, California

AND

RONALD FAGIN

IBM Research Laboratory, San Jose, California

ABSTRACT. It is known that there is an equivalence between functional dependencies in a relational database and a certain fragment of propositional logic This equivalence is extended to include both functional and multivalued dependencies. Thus, for each dependency there is a corresponding statement in propositional logic. It is then shown that a dependency (functional or multivalued) is a consequence of a set of dependencies if and only if the corresponding propositional statement is a consequence of the corresponding set of propositional statements. Examples are given to show that these techniques are valuable in providing much shorter proofs of theorems about dependencies than have been obtained by more traditional means It is shown that this equivalence cannot be extended to include either join dependencies or embedded multivalued dependencies.

KEY WORDS AND PHRASES. relational database, functional dependency, multivalued dependency, embedded multivalued dependency, join dependency, propositional logic

CR CATEGORIES: 4.33, 5 21

1. *Introduction*

Functional dependencies, which were first defined by Codd [8], are an important and widely studied concept in the theory of relational databases. Within the last few years Fagin [11] and, independently, Zaniolo [26] defined a generalization of functional dependencies, called multivalued dependencies, which are also receiving

The work of the first author was partially supported by the National Science Foundation under Grant MCS 76-15255 and by a grant from Bell Laboratories.

Authors' addresses: Y. Sagiv, Department of Computer Science, 222 Digital Computer Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801; C Delobel, Computer Laboratory, University of Grenoble, Boite Postale 53, 38041 Grenoble Cedex, Grenoble, France; D. S. Parker, Jr, Computer Science Department, School of Engineering and Applied Science, University of California at Los Angeles, Los Angeles, CA 90024; R Fagin, IBM Research Laboratory K52/282, 5600 Cottle Road, San Jose, CA 95193

serious study (e.g., [3–7, 14, 16–18, 25]). In this paper we show that in some ways, dependencies (functional or multivalued) behave in precisely the same way as a certain fragment of propositional logic. Informally, the equivalence between dependencies and formulas of propositional logic can be described as follows. For every dependency σ (functional or multivalued) there is a corresponding formula σ in propositional logic. Let Σ be a set of dependencies, and let Σ be the set of the corresponding formulas. The Equivalence Theorem proved in this paper states that σ is implied by Σ if and only if σ is implied by Σ. This result generalizes a theorem of Fagin [12] in which the equivalence is demonstrated in the special case of functional dependencies alone (see also [9] for a closely related result).

The equivalence between dependencies and propositional formulas is helpful in the following way. Suppose we want to show that a dependency σ is a consequence of a set of dependencies Σ. So far there have been only two ways of doing this. Either we show that σ can be inferred from Σ using the inference rules of [5], or we prove (using relations) that for all relations R, σ holds in R if all the dependencies of Σ hold in R. Now there is a third way; namely, we may try to prove that for all truth assignments ψ, the propositional formula σ is true if all the propositional formulas in Σ are true. In Section 6 we demonstrate that by using this equivalence, we can obtain proofs of theorems about dependencies that are simpler and shorter than proofs obtained conventionally. Other applications of this equivalence are discussed in [10, 20, 22].

We prove the Equivalence Theorem in two different ways. One approach is a semantic proof. Suppose that Σ is a set of functional and multivalued dependencies, σ is a single dependency (functional or multivalued), and R is a relation in which all the dependencies of Σ hold and σ fails. In Section 5 we show that R has two tuples that constitute a relation in which all the dependencies of Σ hold, but σ fails. As we shall see, the Equivalence Theorem follows from this result.

Another approach is a syntactic proof. Here we simply show that the inference rules for functional and multivalued dependencies are also inference rules for a fragment of propositional logic. This approach is taken in Section 4. The syntactic proof includes a lemma that is used in Section 6 to obtain a new characterization of the dependency basis in terms of truth assignments. Most of the applications discussed in Section 6 are based on this characterization.

In Section 7 we show that our equivalence cannot be extended to include either join dependencies or embedded multivalued dependencies. In Section 8 we introduce "Boolean dependencies," of which "degenerate multivalued dependencies" are a special case. We demonstrate a puzzling analogy between multivalued dependencies and degenerate multivalued dependencies.

2. *Basic Definitions*

2.1 THE RELATIONAL MODEL. The relational model for databases assumes that the data are stored in tables, called *relations.* The columns of a table are labeled with names, called *attributes.* No two columns have the same name. Each attribute has an associated domain of values. The rows of a table, usually referred to as *tuples* or *records*, can be viewed as mappings from the attributes to their domains. Let r be a tuple of a relation R and A be an attribute of R. Then $r(A)$ is the A-component of r (i.e., the value of r for the attribute A).

In this paper we do not consider a database with several relations, but rather concentrate on a single relation R over a fixed set of attributes $\mathcal{U}$. Let X be a subset of $\mathcal{U}$, and let r be a tuple of R. The *projection* of r onto X, written $r[X]$, is a mapping

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_2	c_2	d_2
a_1	b_1	c_1	d_2
a_1	b_2	c_2	d_1
a_2	b_3	c_1	d_1
a_2	b_3	c_1	d_2

FIGURE 1

from the attributes of X to their domains such that $r[X](A) = r(A)$ for all A in X. We call $r[X]$ an *X-value* (in R). The projection of the relation R onto X is obtained by projecting each tuple in R onto X and eliminating duplicate tuples.

2.2 DEPENDENCIES. Usually the data must satisfy certain constraints. Functional [1, 8] and multivalued [5, 11, 26] dependencies are examples of such constraints. A *functional dependency* (abbreviated FD) is a statement of the form $X \rightarrow Y$, where both X and Y are sets of attributes. A relation R *satisfies* the functional dependency $X \rightarrow Y$ (or $X \rightarrow Y$ *holds* in R) if for every pair r_1, r_2 of tuples of R, if $r_1[X] = r_2[X]$, then $r_1[Y] = r_2[Y]$.

A *multivalued dependency* (abbreviated MVD) is a statement of the form $X \twoheadrightarrow Y$, where X and Y are sets of attributes. Let Z be the set of all the attributes in $\mathcal{U}$ that are neither in X nor in Y. The multivalued dependency $X \twoheadrightarrow Y$ holds in R if for all r_1 and r_2 in R, if $r_1[X] = r_2[X]$, then there are r_3 and r_4 in R such that

(i) $r_3[X] = r_1[X]$, $r_3[Y] = r_1[Y]$, and $r_3[Z] = r_2[Z]$;
(ii) $r_4[X] = r_1[X]$, $r_4[Y] = r_2[Y]$, and $r_4[Z] = r_1[Z]$.

In other words, $X \twoheadrightarrow Y$ means that the set of Y-values associated with a particular X-value must be independent of the values of the rest of the attributes.

It is easy to show that $X \twoheadrightarrow Y$ holds in R if and only if $X \twoheadrightarrow Y - X$ holds in R [11]. If the sets X, Y, and Z form a partition of $\mathcal{U}$ (i.e., every attribute of $\mathcal{U}$ is in exactly one of X, Y, and Z), then it is convenient to write a tuple r of R as (x, y, z), where x, y, and z denote the projections of r onto X, Y, and Z, respectively.

We use letters from the beginning of the alphabet (A, B, C, D, ...) to denote single attributes and letters from the end of the alphabet (..., X, Y, Z) to denote sets of attributes. XY, where both X and Y are sets of attributes, denotes the union of X and Y. Similarly, a string of attributes $A_1A_2 \cdots A_n$ denotes the set $\{A_1, A_2, \ldots, A_n\}$.

Example 1. Consider the relation of Figure 1. This relation is defined on the attributes A, B, C, and D. The functional dependency $B \rightarrow C$ and the multivalued dependency $A \twoheadrightarrow BC$ hold in this relation. □

2.3 INFERENCE RULES FOR DEPENDENCIES. A problem of practical importance is to determine when a dependency is implied by some other dependencies. Formally, we say that a dependency σ is a *consequence* of a set of dependencies Σ if for all relations R, σ holds in R if all the dependencies of Σ hold in R. Previous work in this area [1, 5] shows the existence of inference rules that are instrumental in solving this problem. These rules are sound and complete in the sense that σ is a consequence of Σ if and only if σ can be inferred from Σ by a sequence of applications of the rules. The rules are divided into three groups reflecting the fact that the set $\Sigma \cup \{\sigma\}$ may contain only functional dependencies or only multivalued dependencies, or a mixture of both. Following [5], we now give a list of these rules.

FD Rules

FD1 (Reflexivity). If $Y \subseteq X$, then $X \rightarrow Y$.

FD2 (Augmentation). If $Z \subseteq W$ and $X \rightarrow Y$, then $XW \rightarrow YZ$.

FD3 (Transitivity). If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$.

There are three additional rules that are implied by the above rules.

FD4 (Pseudotransitivity). If $X \rightarrow Y$ and $YW \rightarrow Z$, then $XW \rightarrow Z$.

FD5 (Union). If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$.

FD6 (Decomposition). If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$.

The rules of union and decomposition imply that we can always replace a set of functional dependencies with an equivalent set that has only functional dependencies of the form $X \rightarrow A$ (i.e., the right-hand side contains a single attribute).

MVD Rules

MVD0 (Complementation). Let X, Y, and Z be sets of attributes such that their union is $\mathcal{U}$ and $Y \cap Z \subseteq X$. Then $X \twoheadrightarrow Y$ if and only if $X \twoheadrightarrow Z$.

MVD1 (Reflexivity). If $Y \subseteq X$, then $X \twoheadrightarrow Y$.

MVD2 (Augmentation). If $Z \subseteq W$ and $X \twoheadrightarrow Y$, then $XW \twoheadrightarrow YZ$.

MVD3 (Transitivity). If $X \twoheadrightarrow Y$ and $Y \twoheadrightarrow Z$, then $X \twoheadrightarrow Z - Y$.

The following rules follow from the above MVD rules.

MVD4 (Pseudotransitivity). If $X \twoheadrightarrow Y$ and $YW \twoheadrightarrow Z$, then $XW \twoheadrightarrow Z - YW$.

MVD5 (Union). If $X \twoheadrightarrow Y$ and $X \twoheadrightarrow Z$, then $X \twoheadrightarrow YZ$.

MVD6 (Decomposition). If $X \twoheadrightarrow Y$ and $X \twoheadrightarrow Z$, then $X \twoheadrightarrow Y \cap Z$, $X \twoheadrightarrow Y - Z$, and $X \twoheadrightarrow Z - Y$.

Note that the rules of transitivity, pseudotransitivity, and decomposition are more restricted than the corresponding FD rules.

The FD rules are sufficient when there are only functional dependencies, and the MVD rules are sufficient when there are only multivalued dependencies. When there are both functional and multivalued dependencies, we need the FD rules, the MVD rules, and the following rules.

FD-MVD Rules

FD-MVD1. If $X \rightarrow Y$, then $X \twoheadrightarrow Y$.

FD-MVD2. IF $X \twoheadrightarrow Z$ and $Y \rightarrow Z'$, where $Z' \subseteq Z$ and Y and Z are disjoint, then $X \rightarrow Z'$.

FD-MVD3. If $X \twoheadrightarrow Y$ and $XY \rightarrow Z$, then $X \rightarrow Z - Y$.

The last two rules are equivalent in the presence of the other rules; that is, either one of them can be omitted.

Let Σ be a set of functional and multivalued dependencies, and let $\sigma_1, \sigma_2, \ldots, \sigma_n$ be dependencies (functional or multivalued). We say that $\sigma_1, \sigma_2, \ldots, \sigma_n$ is a *derivation* from Σ if the following is true. For all i ($1 \leq i \leq n$), either σ_i is in $\Sigma \cup \{\sigma_1, \ldots, \sigma_{i-1}\}$ or σ_i can be inferred from $\Sigma \cup \{\sigma_1, \ldots, \sigma_{i-1}\}$ by an application of one of the inference rules. If $\sigma_1, \sigma_2, \ldots, \sigma_n$ is a derivation from Σ, then each σ_i can be *derived* from Σ. The inference rules are *sound* if every dependency σ that can be derived from Σ is also a consequence of Σ. The inference rules are *complete* if every dependency σ that is a consequence of Σ can also be derived from Σ. In [5] it is proved that the inference rules are sound and complete. This result is known as the Completeness Theorem for functional and multivalued dependencies.

2.4. THE DEPENDENCY BASIS AND THE CLOSURE. Let Σ be a set of functional and multivalued dependencies. Given a set of attributes X, consider the following set:

$$\Omega = \{W \mid X \twoheadrightarrow W \text{ can be derived from } \Sigma\}.$$

Note that the elements of this set are sets of attributes. The rules of complementation, union, and decomposition for multivalued dependencies imply that there exists a subset of the above set, called the *dependency basis of* X [5, 11], such that

(a) the sets of the dependency basis are nonempty and their union is $\mathcal{U}$;
(b) the sets of the dependency basis are pairwise disjoint; and
(c) if $X \twoheadrightarrow Y$ can be derived from Σ, then Y is a union of some sets from the dependency basis.

The dependency basis of X contains precisely the minimal nonempty elements of Ω, that is, those nonempty elements W of Ω such that W does not contain a proper nonempty subset W' that is also a member of Ω.

The *closure* of a set of attributes X, denoted by X^*, is the set of all attributes A such that $X \rightarrow A$ can be derived from Σ. Since $X \twoheadrightarrow A$ can be derived from $X \rightarrow A$, it follows that if $A \in X^*$, then $\{A\}$ is a set of the dependency basis of X.

3. *Propositional Logic and Dependencies*

In this section we show the syntactical similarity between dependencies (functional and multivalued) and a fragment of propositional logic. We begin with a definition of this fragment of propositional logic.

3.1 FD AND MVD FORMULAS OF PROPOSITIONAL LOGIC. Let A, B, C, D, ... (possibly with subscripts) be *propositional variables.* Each propositional variable can be assigned either *true* or *false.* We also use the logical connectives + (or), · (and), and $\Rightarrow$ (imply). The formula $A_1 \cdot A_2 \cdot \cdots \cdot A_n \Rightarrow B_1 \cdot B_2 \cdot \cdots \cdot B_m$ is true if and only if all the B_i's are true or some of the A_i's are false. We may simply write this formula as $A_1A_2 \cdots A_n \Rightarrow B_1B_2 \cdots B_m$, where it is understood that a string of propositional variables (e.g., $A_1A_2 \cdots A_n$) stands for the conjunction of these variables (i.e., $A_1 \cdot A_2 \cdot \cdots \cdot A_n$). Following the notation of the previous section, the string $A_1A_2 \cdots A_n$ can be replaced with X, where X is the set $\{A_1, A_2, \ldots, A_n\}$. A set X stands for the conjunction of all the elements it contains. Consequently, a set X is true if all its elements are true, and it is false if some of them are false. The formula $X \Rightarrow Y$, where both X and Y are sets of propositional variables, is true if X is false or Y is true.

Let Σ be a set of formulas. These formulas imply other formulas, that is, formulas that are true whenever all the formulas of Σ are true. Formally, we say that a formula σ is a *logical consequence* of Σ if for all truth assignments ψ, the formula σ is true under ψ if all the formulas of Σ are true under ψ.

The syntactical correspondence between the formula $A_1A_2 \cdots A_n \Rightarrow B_1B_2 \cdots B_m$ and the functional dependency $A_1A_2 \cdots A_n \rightarrow B_1B_2 \cdots B_m$ is quite apparent. We can apply it to the inference rules for functional dependencies in order to obtain inference rules for formulas of the form $X \Rightarrow Y$. For example, the rule of transitivity for functional dependencies becomes the following rule for formulas:

$$\text{If } X \Rightarrow Y \text{ and } Y \Rightarrow Z, \text{ then } X \Rightarrow Z.$$

Syntactical similarity alone cannot guarantee that the resulting inference rules for formulas (of the form $X \Rightarrow Y$) are either sound or complete. However, it so happens that they are both sound and complete [12].

This result can be extended to multivalued dependencies as well. We assume that there is a fixed set of propositional variables $\mathcal{U}$. Let X, Y, and Z be sets of propositional variables whose union is $\mathcal{U}$ and where Z is the complement (in $\mathcal{U}$) of XY. (Recall that XY is the union of X and Y.) Consider the formula $X \Rightarrow Y + Z$. This formula is true if either

(1) X is false, or
(2) Y is true, or
(3) Z is true.

Note that some of X, Y, and Z may be empty. An empty set (i.e., conjunction) of propositional variables is always true.

We abbreviate $X \Rightarrow Y + Z$ by $X \Rightarrow\!\!\!\Rightarrow Y$. Now the correspondence to multivalued dependencies becomes obvious. This correspondence indicates how to translate the inference rules for multivalued dependencies and the mixed rules for functional and multivalued dependencies to propositional logic. We shall show that the resulting rules (including the translated FD rules) are sound and complete for the fragment of propositional logic consisting of all formulas of the form $X \Rightarrow Y$ and $X \Rightarrow\!\!\!\Rightarrow Y$.

Let σ be a dependency, and let Σ be a set of dependencies. Let σ and $\mathbf{\Sigma}$ be the corresponding formula and set of formulas, respectively, in propositional logic. A formula of the form $X \Rightarrow Y$ is called an *FD formula*, and a formula of the form $X \Rightarrow\!\!\!\Rightarrow Y$ is called an *MVD formula*. Note that $\sigma_1, \sigma_2, \ldots, \sigma_n$ is a derivation from $\mathbf{\Sigma}$ if and only if $\sigma_1, \sigma_2, \ldots, \sigma_n$ is a derivation from Σ.

We say that an MVD formula $X \Rightarrow\!\!\!\Rightarrow Y$ is *trivial* if either $Y \subseteq X$ or $XY = \mathcal{U}$. Thus, trivial MVD formulas are those that correspond to trivial multivalued dependencies [11]. As mentioned above, an empty conjunction is always true, and therefore trivial MVD formulas are tautologies (i.e., are true under every truth assignment).

The following proposition states a property of MVD formulas that corresponds to a well-known property of multivalued dependencies [11].

PROPOSITION 1. *$X \Rightarrow\!\!\!\Rightarrow Y$ is true if and only if $X \Rightarrow\!\!\!\Rightarrow Y - X$ is true.*

PROOF. Obvious. □

3.2 AN EXAMPLE. Our Equivalence Theorem states that if σ is the propositional formula corresponding to the (functional or multivalued) dependency σ, and similarly for $\mathbf{\Sigma}$ and Σ, then the following two statements are equivalent:

(1) σ is a consequence of Σ.
(2) σ is a logical consequence of $\mathbf{\Sigma}$.

As an illustration of the Equivalence Theorem, let the set of attributes be $\{A, B, C, D\}$, let σ be the dependency $A \rightarrow B$, and let Σ be $\{A \rightarrow\!\!\!\rightarrow B, C \rightarrow B\}$. Then σ is the propositional formula $A \Rightarrow B$, and $\mathbf{\Sigma}$ is $\{A \Rightarrow B + CD, C \Rightarrow B\}$. By the Equivalence Theorem, either (1) and (2) both hold or (1) and (2) both fail. We now show that in this case (1) and (2) both hold.

Let R be a relation with attributes $\{A, B, C, D\}$ in which the multivalued dependency $A \rightarrow\!\!\!\rightarrow B$ and the functional dependency $C \rightarrow B$ both hold. To show (1), we must show that the functional dependency $A \rightarrow B$ necessarily holds in R. Let (a, b, c, d) and (a, b', c', d') be two tuples of R; we must show that $b = b'$. Since $A \rightarrow\!\!\!\rightarrow B$ holds in R, the tuples (a, b', c, d) and (a, b, c', d') also appear in R (by definition). Since (a, b', c', d') and (a, b, c', d') appear in R, and since R obeys the functional dependency $C \rightarrow B$, necessarily $b = b'$, which was to be shown. So (1) holds.

We now show that (2) holds. Let ψ be a truth assignment that makes both $A \Rightarrow B + CD$ and $C \Rightarrow B$ true. We must show that ψ makes $A \Rightarrow B$ true. If ψ makes A false, we are through. So assume that ψ makes A true; we must show that ψ makes B true. Since ψ makes both A and $A \Rightarrow B + CD$ true, it makes $B + CD$ true. If it makes B true, we are through. So we can assume that it makes CD true. Thus ψ makes C true. Since it also makes $C \Rightarrow B$ true, it makes B true, which was to be shown.

It is not surprising that the proofs of (1) and (2) are quite different, since they deal with completely different universes of discourse. However, according to the Equivalence Theorem, (1) and (2) are either both true or both false (in this case, they are both true). Questions as to whether (1) holds in a particular case of interest come up quite often—for example, in database normalization, and in proofs of theorems. By the Equivalence Theorem, to determine if (1) holds, one can instead solve the perhaps easier problem as to whether (2) holds in propositional logic. One can use all the tools of propositional logic (including truth tables, theorem provers, and even intuition) to obtain the answer. In Section 6 we give several applications where proofs of known theorems using our techniques are simpler and shorter than traditional proofs.

3.3 2-Tuple Relations and Truth Assignments. Before we proceed to prove the equivalence of dependencies and the fragment of propositional logic consisting of FD and MVD formulas, we shall show that there is a correspondence between 2-tuple relations (i.e., relations with only two tuples) and truth assignments. Let R be a 2-tuple relation over the set of attributes $\mathcal{U}$. We construct a truth assignment ψ ("the special truth assignment") for the variables of $\mathcal{U}$ as follows. A variable A is assigned *true* if the two tuples of R agree in the A-column; otherwise, A is assigned *false*.

Lemma 2. *Let R be a 2-tuple relation. Then a dependency σ holds in R if and only if the formula σ is true under the special truth assignment ψ.*

Before we prove this lemma, it is convenient to define a simple new concept and to prove two simple lemmas about it.

If T is a 2-tuple relation and $U \twoheadrightarrow V$ is a multivalued dependency, then we say that $U \twoheadrightarrow V$ holds *actively* in T if it holds in T and if the two tuples in T agree in the U columns. Note that there are two ways that a multivalued dependency $U \twoheadrightarrow V$ can hold in a 2-tuple relation T; either (1) the two tuples disagree in some column of U (in which case $U \twoheadrightarrow V$ holds), or else (2) $U \twoheadrightarrow V$ holds actively in T.

Lemma 3. *Assume that U, V, and W form a partition of the attributes (i.e., each attribute is in exactly one of U, V, or W). Let T be a 2-tuple relation. Then the multivalued dependency $U \twoheadrightarrow V$ holds actively in T if and only if*

(1) *the two tuples of T agree in the U columns; and*
(2) *either the two tuples agree in the V columns, or else they agree in the W columns.*

Proof. If (1) and (2) hold, then clearly $U \twoheadrightarrow V$ holds actively in T. Conversely, assume that $U \twoheadrightarrow V$ holds actively in T. By definition, (1) holds. Write the two tuples of T as (u, v, w) and (u, v', w'). Since $U \twoheadrightarrow V$ holds in T, the tuples (u, v', w) and (u, v, w') must also be members of T. Now if $v \neq v'$ and $w \neq w'$, then it is easy to verify that the tuples (u, v, w), (u, v', w'), (u, v', w), and (u, v, w') are all distinct. But T has only two tuples. Hence $v = v'$ or $w = w'$, which was to be shown. □

The following lemma is not needed for the proof of Lemma 2, but it will be used later.

LEMMA 4. *Assume that T and T' are 2-tuple relations over the same set of attributes, and that whenever the two tuples of T agree in a column, then the two tuples of T' agree in the same column. Then each multivalued dependency that holds actively in T also holds actively in T'.*

PROOF. Assume that the multivalued dependency $U \rightarrow\!\!\!\!\rightarrow V$ holds actively in T (where U, V, and W form a partition of the set of attributes). The two tuples of T agree in the U columns, and, hence, so do the two tuples of T'. By Lemma 3, the two tuples of T either agree in the V columns or agree in the W columns. Hence, so do the two tuples of T'. Thus, $U \rightarrow\!\!\!\!\rightarrow V$ holds actively in T', as desired. □

We can now prove Lemma 2.

PROOF OF LEMMA 2. *If.* Suppose that σ is true under ψ; we must show that σ holds in R. There are two cases to be considered.

Case 1. The formula σ is an FD formula $X \Rightarrow Y$. If X is false, then the two tuples of R disagree in some X column, and hence the functional dependency $X \rightarrow Y$ holds in R. If X is true, then Y is also true, and the two tuples of R must agree in all the Y-columns. Thus the functional dependency $X \rightarrow Y$ holds in R.

Case 2. The formula σ is an MVD formula $X \Rightarrow\!\!\!\!\Rightarrow Y$. If X is false, then obviously the multivalued dependency $X \rightarrow\!\!\!\!\rightarrow Y$ holds in R. Suppose that X is true. Let Z be the complement of XY (in $\mathcal{U}$). It follows that either Y is true or Z is true. Thus, either the two columns of R agree in the Y-columns or they agree in the Z-columns. Lemma 3 implies that in either case $X \rightarrow\!\!\!\!\rightarrow Y$ holds in R.

Only if. The proof is similar to the "if" portion. □

3.4 THE EQUIVALENCE THEOREM. Let σ be a (functional or multivalued) dependency, and let Σ be a set of dependencies. We say that σ is a *consequence* of Σ *in the world of 2-tuple relations* if for all 2-tuple relations R, the dependency σ holds in R whenever Σ (i.e., every dependency of Σ) holds in R. That is, there is no "counter-example" 2-tuple relation R for which Σ holds and σ does not hold. Note that if σ is a consequence of Σ, then σ is a consequence of Σ in the world of 2-tuple relations, but the converse is not obvious (although, as we shall show, it is indeed true). We prove the following theorem.

EQUIVALENCE THEOREM. *Let σ be a dependency, and let Σ be a set of dependencies. The following are equivalent:*

(*a*) *σ is a consequence of Σ.*
(*b*) *σ is a consequence of Σ in the world of 2-tuple relations.*
(*c*) *σ is a logical consequence of Σ.*

Clearly, the major result is the equivalence of (a) and (c).

There are two ways of proving the Equivalence Theorem. The first way is to show that the inference rules are sound and complete for FD and MVD formulas. This proof is essentially a syntactical approach, and we present it in Section 4.

The second way is to show that if we are given a set of dependencies Σ, a single dependency σ, and a relation R such that all the dependencies of Σ hold in R but σ fails in R, then we can find a truth assignment ψ under which all the formulas in Σ are true and σ is false. (We also have to prove the opposite direction. However, as we shall see, it follows easily from Lemma 2.) This approach for proving the Equivalence

Theorem is followed in Section 5. Actually, we prove more than that. Assume that R and P are relations (over the same set of attributes). We say that P is a *subrelation* of R if the tuples in P are a subset of the tuples in R. We show that if R is a relation in which Σ holds and σ fails, then R has a 2-tuple subrelation P such that Σ holds in P and σ fails.

4. *The Syntactic Proof*

In this section we consider FD and MVD formulas. We show that the inference rules are complete also for this interpretation. The first step is to show that the rules are sound, that is, if a formula σ can be derived from a set of formulas Σ, then σ is a logical consequence of Σ. The soundness of the rules is later used to prove that the rules are also complete.

LEMMA 5. *The inference rules (interpreted as rules for formulas of propositional logic) are sound.*

PROOF. In [12] it is proved that the rules for FD formulas are sound. We now show that the rules for MVD formulas and the mixed rules are also sound.

(1) *Complementation.* Let X, Y, and Z be sets of variables such that their union is $\mathcal{U}$, and $Y \cap Z \subseteq X$. We have to prove that $X \Rightarrow\!\!\!\!\Rightarrow Y$ and $X \Rightarrow\!\!\!\!\Rightarrow Z$ are equivalent, that is, for all truth assignments, $X \Rightarrow\!\!\!\!\Rightarrow Y$ is true if and only if $X \Rightarrow\!\!\!\!\Rightarrow Z$ is true.

Let $Y' = Y - X$, and let $Z' = Z - X$. By Proposition 1, $X \Rightarrow\!\!\!\!\Rightarrow Y$ is equivalent to $X \Rightarrow\!\!\!\!\Rightarrow Y'$, and $X \Rightarrow\!\!\!\!\Rightarrow Z$ is equivalent to $X \Rightarrow\!\!\!\!\Rightarrow Z'$. But Z' is the complement of XY', and Y' is the complement of XZ'. Therefore, both $X \Rightarrow\!\!\!\!\Rightarrow Y'$ and $X \Rightarrow\!\!\!\!\Rightarrow Z'$ are shorthand for the same formula, namely, $X \Rightarrow Y' + Z'$, and hence $X \Rightarrow\!\!\!\!\Rightarrow Y$ and $X \Rightarrow\!\!\!\!\Rightarrow Z$ are equivalent.

(2) *Reflexivity.* If $Y \subseteq X$, then Y is true whenever X is true. Thus $X \Rightarrow\!\!\!\!\Rightarrow Y$ is always true.

(3) *Augmentation.* Suppose that $Z \subseteq W$ and $X \Rightarrow\!\!\!\!\Rightarrow Y$ is true. We wish to show that $XW \Rightarrow\!\!\!\!\Rightarrow YZ$ is true. If XW is false, we are through. So suppose XW is true, that is, both X and W are true. If Y is true, so is YZ, and hence $XW \Rightarrow\!\!\!\!\Rightarrow YZ$ is true. If Y is false, then since $X \Rightarrow\!\!\!\!\Rightarrow Y$ is true, the complement of XY is true. Hence, so is the complement of $XWYZ$ (because it is contained in the complement of XY). So once again, $XW \Rightarrow\!\!\!\!\Rightarrow YZ$ is true.

(4) *Transitivity.* Suppose that $X \Rightarrow\!\!\!\!\Rightarrow Y$ and $Y \Rightarrow\!\!\!\!\Rightarrow Z$ are true. To show that $X \Rightarrow\!\!\!\!\Rightarrow Z - Y$ is true, we assume that X is true. If Y is not true, then $Z - Y$ is true (because X and $X \Rightarrow\!\!\!\!\Rightarrow Y$ are true, and hence all the variables that are not in Y are true). So if Y is not true, then $X \Rightarrow\!\!\!\!\Rightarrow Z - Y$ is true, as desired. Therefore, let us assume that Y is true. There are two cases to be considered.

Case 1. Z is true, and hence so is $Z - Y$.

Case 2. The complement of YZ is true. (This case must occur if case 1 does not, since Y and $Y \Rightarrow\!\!\!\!\Rightarrow Z$ are true.) But Y is true, and therefore $Z - Y$ contains all the variables that are false.

Thus $X \Rightarrow\!\!\!\!\Rightarrow Y$ is true in both cases.

(5) *FD–MVD*1. If $X \Rightarrow Y$, then obviously $X \Rightarrow\!\!\!\!\Rightarrow Y$.

(6) *FD–MVD*2. Suppose that $X \Rightarrow\!\!\!\!\Rightarrow Z$ and $Y \Rightarrow Z'$ are true, where $Z' \subseteq Z$, and Y and Z are disjoint. To show that $X \Rightarrow Z'$ is true, we assume that X is true. If Z is true, then so is Z'. If Z is false, then Y is true (since X and $X \Rightarrow\!\!\!\!\Rightarrow Z$ are true, and

hence all the variables that are not in Z must be true). Therefore, $Y \Rightarrow Z'$ implies that Z' is also true. □

Now we can use the soundness of the inference rules in the proof of the following lemma. That is, we consider a set of formulas Σ and derive some additional formulas by applying the inference rules. By Lemma 5, these additional formulas are logical consequences of Σ. We also consider the dependency basis and the closure of a set of variables X. They are defined using the inference rules exactly as for dependencies (see Section 2.4). For example, X^*, the closure of X, is the set of all the variables A such that $X \Rightarrow A$ can be derived from Σ.

LEMMA 6. *Let Σ be a set of FD and MVD formulas, and let X be a set of variables. Suppose that W is a member of the dependency basis of X such that W and X^* are disjoint. If we assign* false *to every variable in W and* true *to all the other variables, then all the formulas in Σ are true.*

PROOF. First we show that all the MVD formulas in Σ are true. Let $Y \Rightarrow\!\!\!\!\Rightarrow Z$ be an MVD formula in Σ. If Y is false, then $Y \Rightarrow\!\!\!\!\Rightarrow Z$ is true. So suppose that under the above truth assignment, Y is true. Since Y is true, and since every variable in W is false, it follows that Y and W are disjoint. Let Y' be the union of all the sets from the dependency basis of X whose intersection with Y is not empty. By the definition of Y', we know that $Y \subseteq Y'$. Note that Y' is a union of members of the dependency basis, not including W. Hence Y' is disjoint from W, and so Y' is true. $Y' \Rightarrow\!\!\!\!\Rightarrow Z$ can be derived from Σ by applying augmentation to $Y \Rightarrow\!\!\!\!\Rightarrow Z$. $X \Rightarrow\!\!\!\!\Rightarrow Y'$ can be derived from Σ, since Y' is a union of sets from the dependency basis of X. Transitivity implies that $X \Rightarrow\!\!\!\!\Rightarrow Z - Y'$ can be derived from $X \Rightarrow\!\!\!\!\Rightarrow Y'$ and $Y' \Rightarrow\!\!\!\!\Rightarrow Z$. Therefore $Z - Y'$ is a union of sets from the dependency basis of X. Since W is a set in the dependency basis of X, either $W \subseteq Z - Y'$ or W and $Z - Y'$ are disjoint. But W is disjoint from Y', and, therefore, either $W \subseteq Z$ or W is disjoint from Z. Thus $Y \Rightarrow\!\!\!\!\Rightarrow Z$ must be true.

Now let $Y \Rightarrow Z$ be an FD formula in Σ. Suppose that Y is true. We must show that Z is true. Let Y' be defined as above. Since $Y \Rightarrow Z$ is in Σ, it follows that $Y \Rightarrow\!\!\!\!\Rightarrow Z$, and hence $Y' \Rightarrow\!\!\!\!\Rightarrow Z$, can be derived from Σ. The MVD formula $X \Rightarrow\!\!\!\!\Rightarrow Z - Y'$ follows from $X \Rightarrow\!\!\!\!\Rightarrow Y'$ and $Y' \Rightarrow\!\!\!\!\Rightarrow Z$ by transitivity. The FD formula $Y \Rightarrow Z - Y'$ can be derived from Σ using $Y \Rightarrow Z$ and decomposition, and hence $X \Rightarrow Z - Y'$ follows from an application of the rule FD–MVD2. Therefore $Z - Y' \subseteq X^*$, and so $Z - Y'$ is true. This implies that Z is true, because Y' is true. □

THEOREM 7. (COMPLETENESS THEOREM FOR FORMULAS). *A formula σ is a logical consequence of Σ if and only if σ can be derived from Σ.*

PROOF. The if portion follows from Lemma 5. In order to prove the other direction, we derive a contradiction by assuming that σ is a logical consequence of Σ and that σ cannot be derived from Σ.

Case 1. The formula σ is an FD formula $X \Rightarrow Y$. Since $X \Rightarrow Y$ cannot be derived from Σ, there is a variable A in Y such that $X \Rightarrow A$ cannot be derived from Σ. Therefore A is not in X^*, and there is a set W in the dependency basis of X such that $A \in W$ and W is disjoint from X^*. Consider the truth assignment under which every variable in W is false and all the other variables are true. Lemma 6 implies that all the formulas in Σ are true, but $X \Rightarrow A$ is false and hence it cannot be a logical consequence of Σ.

Case 2. The formula σ is an MVD formula $X \Rightarrow\!\!\!\!\Rightarrow Y$. Since $X \Rightarrow\!\!\!\!\Rightarrow Y$ cannot be derived from Σ, there is a set W in the dependency basis of X such that $W \cap Y \neq$

$\emptyset$ and $W \not\subseteq Y$ (otherwise Y is a union of some sets from the dependency basis of X). Note that W has more than one element, and since W is in the dependency basis, it must be disjoint from X^*. Suppose that we assign *false* to every variable in W and *true* to all the other variables. By Lemma 6, all the formulas in Σ are true, but $X \Longrightarrow Y$ is false and therefore cannot be a consequence of Σ. □

Now we can give the syntactical proof of the Equivalence Theorem.

THEOREM 8. (EQUIVALENCE THEOREM). *The following are equivalent*:

(*a*) σ *is a consequence of* Σ.
(*b*) σ *is a consequence of* Σ *in the world of 2-tuple relations.*
(*c*) σ *is a logical consequence of* Σ.

PROOF. The fact that (a) implies (b) is obvious.

Next, we show that (b) implies (c) by deriving a contradiction. Suppose that σ is a consequence of Σ in the world of 2-tuple relations, and σ is not a logical consequence of Σ. Let ψ be a truth assignment under which all the formulas of Σ are true and σ is false. We construct a 2-tuple relation R as follows. One tuple of R is defined to be 1 for all attributes A. The other tuple is 1 for an attribute A if A is true under ψ; otherwise it is 0. Lemma 2 implies that Σ holds in R and σ fails in R. This contradicts the assumption that σ is a consequence of Σ in the world of 2-tuple relations.

Finally we have to show that (c) implies (a). Suppose that σ is a logical consequence of Σ. According to Theorem 7, σ can be derived from Σ using the inference rules. But these rules are sound when we interpret them as inference rules for dependencies [5]. Therefore σ is a consequence of Σ. □

The soundness and completeness of the inference rules for dependencies have been proved in [5]. However, Theorem 7 and the portion of the proof of Theorem 8 showing that (a) implies (c) provide an alternative proof of the completeness of the inference rules for dependencies.

5. *The Semantic Proof*

In this section we prove a lemma showing that if R is a relation in which Σ holds and σ fails, then R contains two tuples that constitute a relation which is a counterexample to σ being a consequence of Σ. The semantic proof follows from this lemma, which is interesting in its own right as a model-theoretic result. Note that this lemma strengthens the result of the previous section, where we showed that if σ is not a consequence of Σ, then there is a 2-tuple relation in which Σ holds and σ fails.

LEMMA 9 (2-TUPLE SUBRELATION LEMMA). *Assume that R is a relation,* Σ *is a set of dependencies (functional or multivalued), and* σ *is a single dependency. Suppose that* Σ *holds in R but* σ *fails in R. Then R contains a 2-tuple subrelation for which* Σ *holds and* σ *fails.*

PROOF. There are two cases, depending on whether σ is a functional dependency or a multivalued dependency.

Case 1. The dependency σ is a functional dependency. We can assume without loss of generality that σ is a functional dependency $X \rightarrow A$ in which the right-hand side contains a single attribute. Since $X \rightarrow A$ fails in R, there are two tuples t_1 and t_2 of R that agree in the X columns but disagree in the A column. Consider all 2-tuple subrelations of R in which $X \rightarrow A$ fails. Of all such 2-tuple subrelations of R, let T be the one for which the maximal number of multivalued dependencies hold actively. That is, if T' is another 2-tuple subrelation of R for which σ fails, and if k is the

number of multivalued dependencies that hold actively in T', then at least k multivalued dependencies hold actively in T. We shall now show that all the dependencies of Σ hold in T (which completes the proof in case 1, where σ is a functional dependency).

All functional dependencies in Σ hold in T because they hold in R, and hence in every subrelation. Let $U \twoheadrightarrow V$ be a multivalued dependency in Σ that fails in T; we shall derive a contradiction. Assume without loss of generality that U, V, and W form a partition of the set of attributes. The two tuples in T clearly agree in the U columns (or else $U \twoheadrightarrow V$ would hold in T). Write the two tuples in T as (u, v, w) and (u, v', w'). Then $v \neq v'$ and $w \neq w'$ (or else $U \twoheadrightarrow V$ would hold in T). By assumption, $X \rightarrow A$ fails in T. Thus the two tuples agree in the X columns and disagree in the A column. Since they disagree in the A column, A is in either V or W. Assume without loss of generality that A is in V. Let T' be the 2-tuple relation containing (u, v, w) and (u, v', w). Since $U \twoheadrightarrow V$ holds in R, and since (u, v, w) and (u, v', w') are in R, (u, v', w) is necessarily in R. So T' is a 2-tuple subrelation of R. The two tuples of T' agree in the X columns (since the two tuples of T do) but disagree in the A column (because v and v' disagree in the A column). Thus $X \rightarrow A$ fails in T', and, unlike the situation in T, we see that $U \twoheadrightarrow V$ holds actively in T'. Furthermore, by Lemma 4 every multivalued dependency that holds actively in T also holds actively in T'. So more members of Σ hold actively in T' than in T. Since T' is a 2-tuple subrelation of R for which $X \rightarrow A$ fails, this is a contradiction of the "maximality" in the definition of T. This completes the proof of case 1.

Case 2. The dependency σ is a multivalued dependency $X \twoheadrightarrow Y$. Assume without loss of generality that X, Y, and Z form a partition of the set of attributes. We say that a pair of tuples (x, y, z) and (x, y', z') *witness the failure* of $X \twoheadrightarrow Y$ in a given relation if they appear in that relation and if one of (x, y', z) or (x, y, z') does not appear in that relation. Thus a multivalued dependency fails in a relation if and only if the relation has a pair of tuples that witness the failure. In particular, since the multivalued dependency $X \twoheadrightarrow Y$ fails in R, let (x, y, z) and (x, y', z') witness the failure. Thus either (x, y', z) or (x, y, z') does not appear in R. Of all 2-tuple subrelations of R that witness the failure of $X \twoheadrightarrow Y$, let T be the one for which the maximal number of multivalued dependencies in Σ hold actively. We now show that all of Σ holds in T (which completes the proof, since $X \twoheadrightarrow Y$ fails in T).

As in case 1, each functional dependency in Σ holds in T. Let $U \twoheadrightarrow V$ be a multivalued dependency in Σ that fails in T; we shall derive a contradiction. Assume that U, V, and W form a partition of the set of attributes. As in case 1, the two tuples in T agree in the U columns.

Denote by $\bar{V}$ and $\bar{W}$ those columns in V and W, respectively, for which the tuples in T disagree. Since $U \twoheadrightarrow V$ fails in T, $\bar{V}$ and $\bar{W}$ necessarily are both nonempty. We rewrite (x, y, z) and (x, y', z') as (u, v, w) and (u, v', w'), respectively. Let T_1 be the 2-tuple relation consisting of (u, v, w) and (u, v', w), and let T_2 be the 2-tuple relation consisting of (u, v, w) and (u, v, w'). Obviously T_1 and T_2 are subrelations of R, since $U \twoheadrightarrow V$ holds in R. They are 2-tuple relations since $v \neq v'$ and $w \neq w'$. By Lemma 4 every multivalued dependency of Σ that holds actively in T also holds actively in T_1 and T_2. Clearly $U \twoheadrightarrow V$ holds actively in T_1 and T_2. If $X \twoheadrightarrow Y$ fails either in T_1 or in T_2, we have derived a contradiction to the maximality of T, and hence we are done. So suppose that $X \twoheadrightarrow Y$ holds in both T_1 and T_2. Then $X \twoheadrightarrow Y$ holds actively in T_1 and T_2 since all of the tuples in T, T_1, and T_2 have the same X-value x. It follows from Lemma 3 that the two tuples in T_1 agree either in the Y columns or in the Z columns. In the former case $\bar{V} \subseteq Z$, since $\bar{V}$ contains all of the columns

in which the two tuples of T_1 disagree. In the latter case $\bar{V} \subseteq Y$. Thus we know that $\bar{V} \subseteq Y$ or $\bar{V} \subseteq Z$. Similarly, it follows from our knowledge of T_2 that $\bar{W} \subseteq Y$ or $\bar{W} \subseteq Z$. Since either $\bar{V} \subseteq Y$ or $\bar{V} \subseteq Z$, and since either $\bar{W} \subseteq Y$ or $\bar{W} \subseteq Z$, there are four possibilities:

(a) $\bar{V} \subseteq Y$ and $\bar{W} \subseteq Y$;
(b) $\bar{V} \subseteq Y$ and $\bar{W} \subseteq Z$;
(c) $\bar{V} \subseteq Z$ and $\bar{W} \subseteq Y$;
(d) $\bar{V} \subseteq Z$ and $\bar{W} \subseteq Z$.

Now $\bar{V}\bar{W}$ are all the columns in which the two tuples of T disagree. If (a) were to hold, then the two tuples in T would agree in the Z columns, and hence the multivalued dependency $X \longrightarrow\!\!\!\!\!\rightarrow Y$ would hold in T (which it does not). Similarly, (d) is impossible. So either (b) or (c) holds. We assume without loss of generality that (b) holds. Hence y and y' disagree exactly in the $\bar{V}$ columns, and z and z' disagree exactly in the $\bar{W}$ columns. Under these conditions (x, y', z) and (u, v', w) are identical, and so are (x, y, z') and (u, v, w'). But this is impossible, since (u, v', w) and (u, v, w') are in R, whereas either (x, y', z) or (x, y, z') is not in R. □

This leads us to the semantic proof of the Equivalence Theorem.

THEOREM 10 (EQUIVALENCE THEOREM). *The following are equivalent:*

(*a*) σ *is a consequence of* Σ;
(*b*) σ *is a consequence of* Σ *in the world of 2-tuple relations*;
(*c*) σ *is a logical consequence of* Σ.

PROOF. By Lemma 9, (a) and (b) are equivalent. In the proof of Theorem 8 we showed (using Lemma 2) that (b) implies (c). Since Lemma 2 is an "if and only if" result, we can use it in a similar way to show that (c) implies (b). □

6. *Applications*

One application of the Equivalence Theorem, as we saw at the end of Section 4, is a shorter proof of the Completeness Theorem for dependencies. Other applications are discussed in [10, 20, 22]. The application we describe in this section is a new technique for proving properties of dependencies. We give several examples to demonstrate that this technique provides much shorter proofs than previous methods. Most of these applications are based on a new characterization of the dependency basis in terms of truth assignments (Theorem 11).

In this section we use arguments that are based on truth assignments. For example, we may assign truth values to attributes and evaluate the resulting truth values of some dependencies. However, these arguments cannot constitute a complete formal proof. A formal proof is obtained by translating dependencies to corresponding formulas, proving the desired result for formulas, and finally applying the Equivalence Theorem to obtain a similar result for dependencies. Since the equivalence of dependencies and formulas is well established at this point, we shall forego doing so and simply view truth assignments as being applied directly to dependencies. Therefore our proofs are only an abbreviated version of the complete formal proofs.

We believe that the most important tool for applications is given by the following theorem.[1]

[1] Note that from a formal point of view this theorem should have been stated for formulas and not for dependencies

THEOREM 11. *Assume that Σ is a set of dependencies. Let W and X be disjoint sets of attributes. Consider the truth assignment ψ under which every attribute in W is false and all the other attributes are true.*

(a) If W is a set in the dependency basis of X and is disjoint from X^*, then all the dependencies in Σ are true under this truth assignment.

(b) If all the dependencies in Σ are true under this truth assignment, then W is contained in one set of the dependency basis of X. Furthermore, W is disjoint from X^*.

PROOF

(a) This part follows from Lemma 6.

(b) Suppose that W is not contained in one set of the dependency basis of X. Then there is a set V in the dependency basis of X such that $V \cap W \neq \emptyset$ and $W \not\subseteq V$. Under the truth assignment ψ all the dependencies in Σ are true, but $X \Longrightarrow V$ is false. This is a contradiction, since $X \Longrightarrow V$ can be derived from Σ.

The fact that W is disjoint from X^* is implied by the following observations. If W is not disjoint from X^*, then W (being in the dependency basis) must be a singleton set. Suppose that W is A. The attribute A is a member of X^*, and therefore Σ contains a functional dependency $Y \rightarrow Z$ where A is not a member of Y but A is a member of Z (otherwise, $X \rightarrow A$ cannot be derived from Σ). But A is the only attribute which is false, and hence $Y \rightarrow A$ is false. □

COROLLARY 12. *Let Σ be a set of multivalued dependencies, and let X and V be disjoint sets of attributes. The set V is a member of the dependency basis of X if and only if*

(1) *if the truth assignment ψ makes every attribute of V false and all the other attributes true, then all the dependencies in Σ are true under ψ;*
(2) *for every W such that W is disjoint from X and W is a proper superset of V, the truth assignment making precisely all the attributes of W false does not satisfy Σ.*

PROOF. When there are only multivalued dependencies X^* is equal to X. The rest follows from Theorem 11. □

We remark that Corollary 12 cannot be strengthened by further restricting W in (2) to contain exactly one more attribute than V. In proof, suppose that the attributes are A, B, C, D, and E. Let $\Sigma = \{ADE \twoheadrightarrow B, ACE \twoheadrightarrow B, A \twoheadrightarrow BCD\}$. The dependency basis of A is $\{A, BCD, E\}$. However, if we choose $V = B$, then condition (1) is true, and condition (2) is true if we add the restriction that W contains exactly one more attribute than B.

Theorem 11, coupled with the general concept of truth assignments, is instrumental in proving several properties of dependencies. We shall give short proofs for two theorems about properties of dependencies by applying Theorem 11. These theorems were previously proved using lengthy arguments that are based either on the method of [5] for proving the Completeness Theorem (for dependencies) or on derivations.

The following theorem provides a useful characterization of the dependency basis. It has been used to prove the correctness of some of the algorithms for constructing the dependency basis for a given set of attributes X (e.g., [14]).

THEOREM 13 (BEERI [4]). *Assume that Σ is a set of multivalued dependencies and that X, V_1, V_2, ..., V_n partition the set of attributes. Then V_1, V_2, ..., V_n is the dependency basis of X if and only if*

(1) *$X \twoheadrightarrow V_i$ is a consequence of Σ, and*

(2) *for each multivalued dependency* $Q \twoheadrightarrow R$ *in* Σ *and for each* V_i *disjoint from* Q, *if* $V_i \cap R \neq \emptyset$, *then* $V_i \subseteq R$.

PROOF. *If.* From (2), if we assign *false* to every attribute in V_i and *true* to all the other attributes, then all the dependencies in Σ are true. Thus, by Theorem 11, V_i is contained in a member of the dependency basis of X. From (1), V_i is a union of sets from the dependency basis of X. So V_i is in the dependency basis of X.

Only if. (1) is obvious. Let ψ be a truth assignment under which every attribute in V_i is false and all the other attributes are true. By Theorem 11, all the dependencies in Σ are true under ψ. Hence (2) follows. □

The next theorem is used in some of the algorithms (e.g., [4, 14]) for deciding whether a dependency σ is a consequence of a set of dependencies Σ. Let F be a set of functional dependencies, and let M be a set of multivalued dependencies. We assume that all the functional dependencies in F have only one attribute on the right-hand side. Let $\bar{F} = \{X \twoheadrightarrow A \mid X \rightarrow A \text{ is in } F\}$.

THEOREM 14 (BEERI [4]). *A multivalued dependency* $V \twoheadrightarrow W$ *can be derived from* $F \cup M$ *if and only if it can be derived from* $\bar{F} \cup M$.

PROOF. *If.* Trivial, since $X \rightarrow Y$ implies $X \twoheadrightarrow Y$.

Only if. Suppose that $V \twoheadrightarrow W$ can be derived from $F \cup M$ but not from $\bar{F} \cup M$. Let ψ be a truth assignment under which $\bar{F} \cup M$ is true and $V \twoheadrightarrow W$ is false. We claim that F is also true under ψ. Let $X \rightarrow A$ be a functional dependency in F, and assume that X is true under ψ. If A is false, then it is the only attribute which is false under ψ, because $X \twoheadrightarrow A$ is true. But if only one attribute is false, then $V \twoheadrightarrow W$ must be true. Therefore A is true, and so is $X \rightarrow A$. We may conclude that F is true under ψ, and hence $V \twoheadrightarrow W$ (which is false under ψ) cannot be derived from $F \cup M$. This contradiction completes the proof. □

7. *Nonextendibility of the Equivalence to Join Dependencies or Embedded Multivalued Dependencies*

In this section we show that the equivalence between multivalued dependencies and MVD formulas cannot be extended to cover either join dependencies [13, 15, 21] or embedded multivalued dependencies [11].

We first consider join dependencies. We show that there is no way to extend the mapping between dependencies (functional and multivalued) and propositional logic to include join dependencies and still maintain equivalence. Assume that there is such an extension of the mapping; we shall derive a contradiction. Let X, Y, and Z be a partition of the set of attributes, and let σ be the join dependency $*\{XY, XZ, YZ\}$, which holds for a relation R (over the attributes XYZ) if and only if R is the join of its projections $R[XY]$, $R[XZ]$, $R[YZ]$. (Note that this join dependency is also a mutual dependency [19].) Let $\boldsymbol{\sigma}$ be the alleged propositional formula that corresponds to σ. Now σ is a consequence of the multivalued dependency $X \twoheadrightarrow Y$. Since we assume equivalence, $\boldsymbol{\sigma}$ is a logical consequence of the MVD formula $X \Rightarrow Y + Z$. Similarly, $\boldsymbol{\sigma}$ is a logical consequence of $Y \Rightarrow X + Z$. Now if X is false, then $X \Rightarrow Y + Z$ is true; so $\boldsymbol{\sigma}$ is then true. If X is true, then $Y \Rightarrow X + Z$ is true; so $\boldsymbol{\sigma}$ is again true. Hence $\boldsymbol{\sigma}$ is true under every truth assignment; that is, $\boldsymbol{\sigma}$ is a tautology. But σ does not always hold. This is a contradiction.

We now consider embedded multivalued dependencies. An *embedded multivalued*

FIGURE 2

W	X	Y	Z
w	x	y	z
w	x'	y'	z'
w	x	y'	z
w	x'	y	z'
w	x	y	z'
w	x'	y'	z

dependency [11] $X \twoheadrightarrow Y \mid Z$ holds in a relation R over $\mathcal{U}$ (where X, Y, and Z are subsets of $\mathcal{U}$) if $X \twoheadrightarrow Y$ holds in the projection of R onto XYZ.

Let W, X, Y, and Z be a partition of the set of attributes. Probably the most natural guess as to how to extend the mapping (between dependencies and formulas of propositional logic) would be to have the propositional formula $W \Rightarrow X + Y$ correspond to the embedded multivalued dependency $W \twoheadrightarrow X \mid Y$. However, the propositional formulas $W \Rightarrow X + Y$ and $W \Rightarrow X + Z$ taken together always imply the MVD formula $W \Rightarrow X + YZ$ (in propositional logic). But the embedded multivalued dependencies $W \twoheadrightarrow X \mid Y$ and $W \twoheadrightarrow X \mid Z$ taken together do not imply the multivalued dependency $W \twoheadrightarrow X \mid YZ$. This is shown in the relation of Figure 2 in which $W \twoheadrightarrow X \mid Y$ and $W \twoheadrightarrow X \mid Z$ hold, but $W \twoheadrightarrow X \mid YZ$ fails. So this extension of the equivalence fails.

We now show that this extension (in which the propositional formula $W \Rightarrow X + Y$ corresponds to the embedded multivalued dependency $W \twoheadrightarrow X \mid Y$) provides us with a necessary (but, as we saw, not sufficient) condition for determining whether a dependency σ (functional, multivalued, or embedded multivalued) is a consequence of a set Σ of dependencies (functional, multivalued, or embedded multivalued). Let $\mathbf{\Sigma}$, $\boldsymbol{\sigma}$ be the corresponding formulas in propositional logic. We now sketch a proof that if σ is a consequence of Σ, then $\boldsymbol{\sigma}$ is a logical consequence of $\mathbf{\Sigma}$. As in Theorem 8, let (a) be the statement that σ is a consequence of Σ, let (b) be the statement that σ is a consequence of Σ in the world of 2-tuple relations, and let (c) be the statement that $\boldsymbol{\sigma}$ is a logical consequence of $\mathbf{\Sigma}$. Unlike the situation in Theorem 8, we are allowing not just functional and multivalued dependencies, but also embedded multivalued dependencies. As before, (a) implies (b) and (b) implies (c), even in this generalized context. But "(a) implies (c)" is exactly what we wanted to prove. (We note that (c) implies (b) even in this generalized context, but (c) does not imply (a), as we saw by a counterexample.)

We now show that there is no way to extend the mapping between dependencies (functional and multivalued) and propositional logic to include embedded multivalued dependencies and still maintain equivalence. Let W, X, Y, and Z be a partition of the set of attributes, where none of W, X, Y, or Z is empty. Assume that there is such an extension of the mapping; we shall derive a contradiction.

Let σ be the embedded multivalued dependency $W \twoheadrightarrow X \mid Y$, and let $\boldsymbol{\sigma}$ be the alleged propositional formula that corresponds to σ. We shall prove that $\boldsymbol{\sigma}$ must be (equivalent to) $W \Rightarrow X + Y$. A similar proof would show that the propositional formula corresponding to the embedded multivalued dependency $W \twoheadrightarrow X \mid Z$ is $W \Rightarrow X + Z$. But then we run into the same contradiction as we showed earlier. So the proof that there is no way to extend the mapping to include embedded multivalued dependencies is complete if we can show that $\boldsymbol{\sigma}$ is necessarily equivalent to $W \rightarrow X + Y$, where σ is the embedded multivalued dependency $W \twoheadrightarrow X \mid Y$.

Now $W \twoheadrightarrow X \mid Y$ is a consequence of the multivalued dependency $W \twoheadrightarrow X \mid YZ$, by projection [11]. Hence $\boldsymbol{\sigma}$ is a logical consequence of the MVD formula $W \Rightarrow X + YZ$. Thus $\boldsymbol{\sigma}$ is true if W is false or X is true. Similarly, since $W \twoheadrightarrow X \mid Y$ is a

A	*B*	*C*
a	*b*	*c*
a	*b'*	*c*
a'	*b*	*c*
a'	*b*	*c'*

FIGURE 3

consequence of $W \twoheadrightarrow XZ \mid Y$, it follows that σ is true if Y is true. So σ is true if either W is false, X is true, or Y is true. To show that σ is equivalent to $W \Rightarrow X + Y$ (and thus complete the proof), we must show that this "if" is an "if and only if," that is, that σ is false under each truth assignment in which, simultaneously, W is true, X is false, and Y is false.

Let us denote the dependencies $W \twoheadrightarrow X \mid Y$, $WY \twoheadrightarrow X \mid Z$, and $W \twoheadrightarrow X \mid YZ$ by σ, τ_1, and τ_2, respectively. It has been shown that τ_2 is a consequence of $\{\sigma, \tau_1\}$ [24, 25]. By equivalence (which we are assuming), τ_2 is a logical consequence of $\{\sigma, \tau_1\}$. Consider a truth assignment ψ under which W is true, X is false, and Y is false. Under ψ we know that τ_1 is true and τ_2 is false, since τ_1 is $WY \Rightarrow X + Z$ and τ_2 is $W \Rightarrow X + YZ$. If σ were true under ψ, then since τ_1 is true under ψ and τ_2 is a logical consequence of $\{\sigma, \tau_1\}$, it would follow that τ_2 would be true under ψ, a contradiction. So σ is false under ψ. This was to be shown.

8. *Boolean Dependencies*

It is possible to extend the equivalence between functional dependencies and FD formulas by generalizing the notion of a functional dependency. We define a *Boolean dependency* to be an arbitrary formal Boolean combination of attributes. For example, if A, B, and C are attributes, then $A + (B \cdot \sim C)$ is a Boolean dependency, which has the meaning, "For every pair of tuples, either (1) the two tuples agree in column A, or (2) the two tuples agree in column B and disagree in column C." The usual functional dependency $A_1A_2 \cdots A_n \rightarrow B_1B_2 \cdots B_m$ is a special case which has the meaning, "For every pair of tuples, if the tuples agree in columns $A_1, A_2, \ldots, A_n$, then the tuples agree in columns $B_1, B_2, \ldots, B_m$." (However, multivalued dependencies are not a special case of Boolean dependencies.) It is clear how to form the corresponding propositional formula (A is replaced by A, etc.). The semantic proof of the Equivalence Theorem for functional dependencies and FD formulas [12] goes through with minor modifications, to give the following result.

THEOREM 15. *Assume that Σ is a set of Boolean dependencies and σ is a single Boolean dependency. Let $\mathbf{\Sigma}$ and σ be, respectively, the corresponding set of propositional formulas and single propositional formula. The following are equivalent:*

(1) *σ is a consequence of Σ.*
(2) *σ is a consequence of Σ in the world of 2-tuple relations.*
(3) *σ is a logical consequence of $\mathbf{\Sigma}$.*

Let us single out those Boolean dependencies of the form $A_1A_2 \cdots A_n \Rightarrow B_1B_2 \cdots B_m + C_1C_2 \cdots C_p$, where each attribute is exactly one of $A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m, C_1, C_2, \ldots, C_p$. Let us call these *degenerate multivalued dependencies* and use $\rightarrow$ instead of $\Rightarrow$. The degenerate multivalued dependency $A_1A_2 \cdots A_n \rightarrow B_1B_2 \cdots B_m + C_1C_2 \cdots C_p$ holds for a relation R if and only if every pair of tuples of R that agree in all columns of $A_1, A_2, \ldots, A_n$ agree either in each of $B_1, B_2, \ldots, B_m$, or in each of $C_1, C_2, \ldots, C_p$. As the relation in Figure 3 shows, it is possible for the degenerate multivalued dependency $A \rightarrow B + C$ to hold without either of the functional dependencies $A \rightarrow B$ or $A \rightarrow C$ holding. We note that degenerate multivalued dependencies are also studied in [2].

Let X, Y, and Z form a disjoint partition of the set of attributes, and let R be a relation on this set of attributes. Consider the following statements.

(1) $X \rightarrow Y$ or $X \rightarrow Z$ holds in R.
(2) $X \rightarrow Y + Z$ holds in R.
(3) $X \rightarrow\!\!\!\!\rightarrow Y$ holds in R.

It is easy to show that (1) implies (2) and (2) implies (3), but (3) does not imply (2) and (2) does not imply (1). However, our theorems give a somewhat puzzling analogy between multivalued dependencies and degenerate multivalued dependencies.

THEOREM 16. *Let Σ be a set of functional dependencies and multivalued dependencies, and let σ be a single dependency (functional or multivalued). Let Σ' be the result of replacing each multivalued dependency in Σ by the corresponding degenerate multivalued dependency, and similarly for σ and σ'. The following are equivalent:*

(1) *σ is a consequence of Σ.*
(2) *σ' is a consequence of Σ'.*

PROOF. By Theorems 8 and 15, both (1) and (2) are equivalent to "σ is a logical consequence of Σ." □

9. *Historical Note*

This current paper is the end product of two earlier reports, one by Delobel and Parker [10] and the other by Sagiv and Fagin [23]. Chronologically, Sagiv was the first to conceive of extending the equivalence between functional dependencies and a fragment of propositional logic to include also multivalued dependencies, but Delobel and Parker developed the result independently and published it first. This paper is drawn mainly from the subsequent, easier-to-read report by Sagiv and Fagin.

ACKNOWLEDGMENT. The authors are grateful to J. D. Ullman for helpful discussions.

REFERENCES

1 ARMSTRONG, W W Dependency structures of database relationships Proc. IFIP 74, North-Holland, Amsterdam, 1974, pp. 580–583
2 ARMSTRONG, W W, AND DELOBEL, C. Decompositions and functional dependencies in relations. *ACM Trans. Database Syst 5*, 4 (Dec. 1980), 404–430
3 BEERI, C. On the role of data dependencies in the construction of relational database schemas. Tech Rep. No 43, Dep. of Computer Science, The Hebrew Univ of Jerusalem, Jerusalem, Israel, Jan 1979
4. BEERI, C. On the membership problem for functional and multivalued dependencies in relational databases *ACM Trans Database Syst 5*, 3 (Sept 1980), 241–259.
5 BEERI, C., FAGIN, R, AND HOWARD, J H. A complete axiomatization for functional and multivalued dependencies in database relations Proc. ACM-SIGMOD Int Conf on Management of Data, Toronto, Ontario, Canada, Aug. 1977, pp 47–61.
6. BISKUP, J On the complementation rule for multivalued dependencies in database relations *Acta Inform 10*, 3 (1978), 297–305.
7 BISKUP, J Inferences of multivalued dependencies in fixed and undetermined universes *Theoret. Comput. Sci 10*, 1 (Jan 1980), 93–105
8 CODD, E F A relational model for large shared data banks *Commun ACM 13*, 6 (June 1970), 377–387.
9. DELOBEL, C., AND CASEY, R.G Decomposition of a data base and the theory of Boolean switching functions *IBM J. Res Dev. 17*, 5 (Sept 1973), 374–386 Also "Comment," *IBM J. Res and Dev 21*, 5 (Sept. 1977), 484–485.

10. DELOBEL, C., AND PARKER, D.S. Functional and multivalued dependencies in a relational database and the theory of Boolean switching functions Tech Rep, Univ of Grenoble, Grenoble, France, Nov. 1978
11 FAGIN, R Multivalued dependencies and a new normal form for relational databases. *ACM Trans. Database Syst 2*, 3 (Sept 1977), 262–278
12 FAGIN, R. Functional dependencies in a relational database and propositional logic. *IBM J. Res. Dev 21*, 6 (Nov. 1977), 534–544.
13. FAGIN, R Normal forms and relational database operators. Proc ACM-SIGMOD Int. Conf. on Management of Data, Boston, Mass, May 1979, pp. 153–160
14 HAGIHARA, K, ITO, M, TANIGUCHI, K, AND KASAMI, T Decision problems for multivalued dependencies in relational databases. *SIAM J. Comput 8*, 2 (May 1979), 247–264.
15. MAIER, D, MENDELZON, A.O., AND SAGIV, Y. Testing implications of data dependencies. *ACM Trans Database Syst 4*, 4 (Dec. 1979), 455–469
16 MAKINOUCHI, A A consideration on normal form of not-necessarily-normalized relation in the relational database model Proc 3rd Int Conf. on Very Large Data Bases, Tokyo, Japan, Oct. 1977, pp. 447–453
17 MENDELZON, A.O On axiomatizing multivalued dependencies in relational databases. *J ACM 26*, 1 (Jan 1979), 37–44.
18 NICOLAS, J M First order logic formalization for functional, multivalued and mutual dependencies. Proc ACM-SIGMOD Int Conf on Management of Data, Austin, Texas, June 1978, pp. 40–46.
19 NICOLAS, J.M Mutual dependencies and some results on undecomposable relations. Proc. 4th Int Conf on Very Large Data Bases, West Berlin, Sept 1978, pp. 360–367.
20. PARKER, D.S., AND DELOBEL, C Algorithmic applications for a new result on multivalued dependencies. Proc 5th Int Conf on Very Large Data Bases, Rio de Janeiro, Brazil, Oct. 1979, pp. 67–74.
21. RISSANEN, J. Theory of relations for databases—a tutorial survey In *Proc. 7th Symp. on Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science 64, Springer-Verlag, Berlin and Heidelberg, 1978, pp. 536–551.
22 SAGIV, Y. An algorithm for inferring multivalued dependencies with an application to propositional logic. *J ACM 27*, 2 (April 1980), 250–262.
23 SAGIV, Y, AND FAGIN, R An equivalence between relational database dependencies and a subset of propositional logic. Res Rep RJ2500, IBM Research Lab, San Jose, Calif., March 1979.
24 SAGIV, Y, AND WALECKA, S F Subset dependencies and a completeness result for a subclass of embedded multivalued dependencies To appear *J. ACM*
25 TANAKA, K, KAMBAYASHI, Y., AND YAJIMA, S. Properties of embedded multivalued dependencies in relational databases *Trans. IECE Japan E62*, 8 (Aug. 1979), 536–543
26 ZANIOLO, C. Analysis and design of relational schemata for database systems Tech Rep UCLA-ENG-7669, Dep of Computer Science, University of California, Los Angeles, Calif, July 1976

RECEIVED JANUARY 1980; REVISED JANUARY 1980; ACCEPTED MAY 1980