R. Fagin

Functional Dependencies in a Relational Database and Propositional Logic

Abstract: An equivalence is shown between functional dependency statements of a relational database, where "→" has the meaning of "determines," and implicational statements of propositional logic, where "⇒" has the meaning of "implies." Specifically, it is shown that a dependency statement is a consequence of a set of dependency statements iff the corresponding implicational statement is a consequence of the corresponding set of implicational statements. The database designer can take advantage of this equivalence to reduce problems of interest to him to simpler problems in propositional logic. A detailed algorithm is presented for such an application. Two proofs of the equivalence are presented: a "syntactic" proof and a "semantic" proof. The syntactic proof proceeds in several steps. It is shown that 1) Armstrong's Dependency Axioms are complete for dependency statements in the usual logical sense that they are strong enough to prove every consequence, and that 2) Armstrong's Axioms are also complete for implicational statements in propositional logic. The equivalence then follows from 1) and 2). The other proof proceeds by considering appropriate semantic interpretations for the propositional variables. The Delobel-Casey Relational Database Decomposition Theorems, which heretofore have seemed somewhat fortuitous, are immediate and natural corollaries of the equivalence. Furthermore, a counterexample is demonstrated, which shows that what seems to be a mild extension of the equivalence fails.

Introduction

The concept of functional dependencies [1] is one of the few in the database area that is both intuitively simple and yet complex enough that an advanced development is possible (see, for example, [2–6]). Functional dependencies are important tools for database design: in fact, in one approach to database design [6], they are essentially the only input.

Because of their importance and intuitive simplicity, there is considerable interest in studying their properties. In this paper, it is shown that in some ways functional dependencies behave precisely the same as a certain well-studied subset of propositional logic. In particular, it is possible to take advantage of artificial intelligence research in the area of theorem-proving by directly converting results in that area into results about functional dependencies.

In this paper, we refer to functional dependencies by the name "dependency statements." This is done for several reasons. The first is to emphasize the analogy with implicational statements, defined soon. The second is that there is some confusion as to exactly what functional dependencies are. Codd [1] considers them to be statements, or sentences, that can either hold or not hold for a given database relation. However, Bernstein [6] defines a functional dependency to be a "time-varying function." Some practical distinctions resulting from the two different definitions are discussed in [7]. By using the name "dependency statements," we emphasize their role as simple sentences, which can hold for certain database relations and not hold for others.

We now give Codd's definition. Assume that $\mathscr{R}$ is a database relation, and that each column of $\mathscr{R}$ has a unique "column name." If $A_1, \cdots, A_m, B_1, \cdots, B_r$ are among the column names of $\mathscr{R}$ (they need not be distinct), then we say that $A_1, \cdots, A_m$ *determine* $B_1, \cdots, B_r$ (or $B_1, \cdots, B_r$ *depend on* $A_1, \cdots, A_m$) if whenever two tuples (that is, rows) of $\mathscr{R}$ agree in columns $A_1, \cdots, A_m$, then they also agree in columns $B_1, \cdots, B_r$. (Two tuples *agree* in a column if their entries under that column are the same.) We write $\{A_1, \cdots, A_m\} \rightarrow \{B_1, \cdots, B_r\}$, or, more simply, $A_1 \cdots A_m \rightarrow B_1 \cdots B_r$, and we call each such statement a *dependency statement*. For convenience, we assume throughout this paper that we are dealing with static (i.e., time-invariant) relations, although only trivial modifications are called for to deal with time-varying relations, such as would occur in actual relational databases. Our approach in this paper is to hold fixed a dependency statement or a set of dependency statements and then to derive properties of the collection of all relations $\mathscr{R}$ for which the given dependency statements hold.

With each column name A we associate a distinct propositional (i.e., Boolean) variable **A**. With each dependency statement $A_1 \cdots A_m \rightarrow B_1 \cdots B_r$ we associate the propositional statement $\mathbf{A}_1 \wedge \cdots \wedge \mathbf{A}_m \Rightarrow \mathbf{B}_1 \wedge \cdots \wedge \mathbf{B}_r$, or, as we shall write it, $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{B}_1 \cdots \mathbf{B}_r$. We call each such statement an *implicational statement* of propositional logic. (We write ⇒ instead of the more usual → to prevent confusion with the dependency symbol.) It is clear that the correspondence is one-one and onto; thus, for each implicational statement there is also a cor-

responding dependency statement. Let t be a *truth assignment*, that is, a mapping that assigns to each propositional variable either the value 0 (false) or 1 (true). The propositional statement $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{B}_1 \cdots \mathbf{B}_r$ has truth value 0 under truth assignment t if each of $\mathbf{A}_1, \cdots, \mathbf{A}_m$ has the truth value 1 under t and at least one of $\mathbf{B}_1, \cdots, \mathbf{B}_r$ has truth value 0; otherwise, it has truth value 1.

Before we can state the main result in this paper, we need to define some concepts. Assume that DEP is a set of dependency statements and α is a single dependency statement. When we say that "α is a consequence of DEP," we mean that α holds for every relation that obeys each dependency statement in DEP. That is, α is a consequence of DEP iff there is no "counterexample" relation $\mathscr{R}$ such that each dependency statement in DEP is true in $\mathscr{R}$ but such that α is false in $\mathscr{R}$. An example might be helpful. Let DEP be the set $\{AB \rightarrow C, AC \rightarrow D\}$ of dependency statements, and let α be the dependency statement $AB \rightarrow D$. We will show that α is a consequence of DEP. Let $\mathscr{R}$ be arbitrary. Assume that DEP holds for $\mathscr{R}$; we will show that α holds for $\mathscr{R}$. To show that α holds for $\mathscr{R}$, we assume that the two tuples $\mathbf{T}_1$ and $\mathbf{T}_2$ of $\mathscr{R}$ agree in columns A and B; we must show that $\mathbf{T}_1$ and $\mathbf{T}_2$ agree in column D. Since $\mathbf{T}_1$ and $\mathbf{T}_2$ agree in columns A and B, and since the dependency statement $AB \rightarrow C$ holds for $\mathscr{R}$, we know that $\mathbf{T}_1$ and $\mathbf{T}_2$ agree in column C. Since $\mathbf{T}_1$ and $\mathbf{T}_2$ agree in columns A and C, and since the dependency statement $AC \rightarrow D$ holds for $\mathscr{R}$, we know that $\mathbf{T}_1$ and $\mathbf{T}_2$ agree in column D, which was to be shown.

Now let **DEP** be a set of implicational statements of propositional logic, and let $\boldsymbol{\alpha}$ be a single implicational statement. (We use the name **DEP** for a set of implicational statements since, later on, we think of **DEP** as being the set of implicational statements that correspond to the set DEP of dependency statements.) When we say that "$\boldsymbol{\alpha}$ is a logical consequence of **DEP**," we mean that $\boldsymbol{\alpha}$ has truth value 1 for every truth assignment that gives truth value 1 to each implicational statement in **DEP**. That is, $\boldsymbol{\alpha}$ is a logical consequence of **DEP** iff there is no counterexample truth assignment t such that each implicational statement in **DEP** has truth value 1 under t but such that $\boldsymbol{\alpha}$ has truth value 0 under t.

As an example, let **DEP** be the set $\{\mathbf{AB} \Rightarrow \mathbf{C}, \mathbf{AC} \Rightarrow \mathbf{D}\}$ of implicational statements of propositional logic, and let $\boldsymbol{\alpha}$ be the statement $\mathbf{AB} \Rightarrow \mathbf{D}$. Then $\boldsymbol{\alpha}$ is a logical consequence of **DEP**, because if t is one of the $2^4 = 16$ possible truth assignments to (**A, B, C, D**), and if it happens that each statement in **DEP** has truth value 1 under truth assignment t, then it is easy to verify that so does $\boldsymbol{\alpha}$.

We can now state and discuss the main result in this paper, which establishes an equivalence between dependency statements and implicational statements. The proof is deferred until Section 3.

Equivalence Theorem Assume that DEP is a set of dependency statements and α is a single dependency statement. Let **DEP**, $\boldsymbol{\alpha}$ be, respectively, the corresponding set of implicational statements and single implicational statement. Then α is a consequence of DEP iff $\boldsymbol{\alpha}$ is a logical consequence of **DEP**.

We can now see the practical and theoretical utility of the Equivalence Theorem. As a first example, let DEP, as before, be the set $\{AB \rightarrow C, AC \rightarrow D\}$ of dependency statements, and let α be the dependency statement $AB \rightarrow D$; let **DEP** be the corresponding set $\{\mathbf{AB} \Rightarrow \mathbf{C}, \mathbf{AC} \Rightarrow \mathbf{D}\}$ of implicational statements and $\boldsymbol{\alpha}$ the implicational statement $\mathbf{AB} \Rightarrow \mathbf{D}$. In this case, we have shown, by two quite different proofs, that

1. α is a consequence of DEP

and that

2. $\boldsymbol{\alpha}$ is a logical consequence of **DEP**.

It is not surprising that the proofs of 1 and 2 are quite different since they deal with completely different universes of discourse. According to the Equivalence Theorem, 1 and 2 above are either both true or both false. (In this case, they are both true.) So, if a database designer were confronted with the problem (Problem 1) as to whether 1 holds in a particular case of interest (he might be normalizing relations [1] or determining keys, and many dependency statements might be involved), he could instead solve the perhaps easier problem (Problem 2) as to whether 2 holds in propositional logic. He can solve Problem 2 by whatever means he finds easiest (such as by using truth tables, by using a theorem-prover, etc.), and he is automatically guaranteed (by the Equivalence Theorem) to get the correct answer to Problem 1.

Let us look at a specific example (involving a new DEP and α) in which the database designer might solve Problem 1 by instead solving Problem 2 and using the tools of propositional logic. Assume that he is examining a relation $\mathscr{R}$ with exactly four columns, A, B, C, D, and for which the only dependency statements that hold are those in the following set DEP (and its consequences):

$AB \rightarrow D$

$BC \rightarrow A$

$BC \rightarrow D$

$CD \rightarrow A$

$CD \rightarrow B$

He is trying to decide whether AB is a key of $\mathscr{R}$. Since he already knows that $AB \rightarrow A$, $AB \rightarrow B$, and $AB \rightarrow D$, he needs to determine whether or not $AB \rightarrow C$. Let α be the dependency statement $AB \rightarrow C$. He wants to solve Problem 1, that is, to know whether α is a consequence

of DEP. By the Equivalence Theorem, he can instead solve Problem 2, that is, to determine whether α is a logical consequence of **DEP**, where, of course, α is $\mathbf{AB} \Rightarrow \mathbf{C}$, and **DEP** is

$\mathbf{AB} \Rightarrow \mathbf{D}$

$\mathbf{BC} \Rightarrow \mathbf{A}$

$\mathbf{BC} \Rightarrow \mathbf{D}$

$\mathbf{CD} \Rightarrow \mathbf{A}$

$\mathbf{CD} \Rightarrow \mathbf{B}$

Let t be the truth assignment that assigns truth value 1 to **A**, **B**, and **D**, and truth value 0 to **C**. It is very simple to check that each implicational statement in **DEP** has truth value 1 under t, whereas α has truth value 0 under t. So, since we have exhibited a counterexample to truth assignment t such that each statement in **DEP** has truth value 1 under t but such that α has truth value 0 under t, it follows that the answer to Problem 2 is "No, α is not a logical consequence of **DEP**." Hence, by the Equivalence Theorem, the answer to Problem 1 is "No, α is not a consequence of DEP," and so AB is not a key.

In addition to the truth-table method, there are fast, special-purpose theorem-provers for solving Problem 2. It follows from the Equivalence Theorem that these theorem-provers can be used directly as efficient means to solve Problem 1. In Section 2, we apply such a special-purpose theorem-prover to give an efficient solution to Problem 1. Thus, one of the practical benefits of the Equivalence Theorem is that we can take advantage of artificial intelligence research that has gone towards finding efficient solutions to Problem 2 to obtain directly efficient solutions to Problem 1.

In Sections 3 and 4, we present two proofs of the Equivalence Theorem. The first (syntactic) proof proceeds in several steps. It is shown that 1) Armstrong's Dependency Axioms (see Section 3) are complete for dependency statements. By *complete*, we mean that α is a consequence of DEP iff there is a proof (in a finite number of steps) of α from DEP by applying Armstrong's Axioms. Further, we show that 2) Armstrong's Axioms (when converted, as above, by replacing each occurrence of $\rightarrow$ by $\Rightarrow$), are complete for implicational statements of propositional logic. We then show that the Equivalence Theorem follows from 1) and 2).

Our other proof of the Equivalence Theorem is semantic in nature. It proceeds by considering appropriate interpretations for the propositional variables.

In Section 5, we show that what seems to be a mild extension of the Equivalence Theorem fails. This somewhat surprising failure shows the subtlety of the Equivalence Theorem. Thus, those who feel that the Equivalence Theorem is "obvious" might also feel that the mild extension is only slightly less obvious, although, in fact, it is false!

In Section 6, we show that the important, widely referenced Decomposition Theorems of Delobel and Casey [4] follow immediately from our Equivalence Theorem. Perhaps the main contribution of this paper is to present new proofs of these theorems. Furthermore, we feel that our Equivalence Theorem as we state it is more enlightening than the statement of the Delobel-Casey Theorems.

2. An efficient algorithm for determining consequence

Let α be a dependency statement and DEP a set of dependency statements. Let α, **DEP** be the corresponding implicational statements. By the Equivalence Theorem, the problem (Problem 1) as to whether or not α is a consequence of DEP is equivalent to the problem (Problem 2) as to whether or not α is a logical consequence of **DEP**. Now Problem 2 can be converted into the well-studied problem of satisfiability of propositional Horn clauses [8–10]. A fast algorithm for the Horn clause satisfiability problem is the "first-literal unit resolution procedure," which is due to Chang [11, p. 130]. By the Equivalence Theorem, we can exploit Chang's algorithm (which solves Problem 2) to obtain an efficient algorithm to solve Problem 1. We now explicitly describe the resulting algorithm.

For convenience, we assume that the dependency statement α and the dependency statements in DEP each have exactly one column name on the right-hand side (it is easy to see how to convert problems in which this is not the case into problems in which this is the case).

In the first step of the algorithm, we form a set $\mathscr{S}$ of strings of symbols. Each string contains three types of symbols: column names, negation signs ($\sim$), and commas (,). For each statement $\mathbf{A}_1 \cdots \mathbf{A}_m \rightarrow \mathbf{B}$ in DEP, we include in $\mathscr{S}$ the string

$$\sim\mathbf{A}_1, \cdots, \sim\mathbf{A}_m, \mathbf{B}$$

(For those who want a glimpse of what is going on behind the scenes: This string corresponds to the "Horn clause" $\sim\mathbf{A}_1 \vee \cdots \vee \sim\mathbf{A}_m \vee \mathbf{B}$, which is logically equivalent to the propositional formula $\mathbf{A}_1 \wedge \cdots \wedge \mathbf{A}_m \Rightarrow \mathbf{B}$.)

If α is the dependency statement $\mathbf{C}_1 \cdots \mathbf{C}_k \rightarrow \mathbf{D}$, then also include in $\mathscr{S}$ the $(k+1)$ strings

$$\mathbf{C}_1$$
$$\vdots$$
$$\mathbf{C}_k$$
$$\sim\mathbf{D}$$

For example, if DEP and α are as in our first example in the Introduction (that is, DEP is $\{AB \rightarrow C, AC \rightarrow D\}$, and α is $AB \rightarrow D$), then $\mathscr{S}$ contains the five strings

$\sim A, \sim B, C$
$\sim A, \sim C, D$
A
B
$\sim D$ (1)

We call each column name an "atom," and we call the concatenation of a negation sign with a column name a "negative atom." To get our terminology straight: in (1), there are two strings that are atoms (namely, A and B), one string that is a negative atom (namely, $\sim D$), and three strings that "begin with" negative atoms (the first, second, and fifth strings).

The algorithm proceeds by searching for an atom X such that

a. X is a string in $\mathscr{S}$,
and
b. There is a string in $\mathscr{S}$ that begins with $\sim X$.

In our example, there are two atoms X (namely, A and B) that satisfy a, and two atoms X (namely, A and D) that satisfy b. The only atom X that satisfies both a and b is A. (If there were several atoms X that satisfied both a and b, the algorithm would now arbitrarily select one of them.) In the next step of the algorithm, we shorten each string that begins with $\sim X$ by erasing the leading negation sign, the X, and the comma that follows X (if there is such a comma). In (1), where X is A, the first two strings are shortened, and we are left with

$\sim B, C$
$\sim C, D$
A
B
$\sim D$ (2)

We repeat the procedure by again searching for an atom X that satisfies both a and b above. In (2), the only such atom is B. After the "shortening" procedure is applied, we are left with

C
$\sim C, D$
A
B
$\sim D$ (3)

We again repeat the procedure. In (3), the only atom X that satisfies both a and b is C. After the shortening procedure, we are left with

C
D
A
B
$\sim D$ (4)

After another iteration (where X is now D), we are left with

C
D
A
B
λ (5)

where λ is the empty string. The entire algorithm halts either when 1) the empty string λ is generated (as happened in this case) or when 2) there is no atom X that satisfies both a and b. If 1) occurs first, that is, if the empty string λ is generated, then α is a consequence of DEP (as in this case). If 2) occurs first, then α is not a consequence of DEP.

It is easy to see that the algorithm must always terminate, and thereby give an answer. That this algorithm gives the *correct* answer is an immediate consequence of Chang's theorem on Horn clauses and our Equivalence Theorem.

3. Completeness of Armstrong's Axioms, and the Equivalence Theorem

Armstrong's Axioms consist of the following three schemata:

(A1) $A_1 \cdots A_m \rightarrow A_i$, for $i = 1, \cdots, m$.

(A2) $A_1 \cdots A_m \rightarrow B_1 \cdots B_r$ iff, for each i, $A_1 \cdots A_m \rightarrow B_i$.

(A3) If $A_1 \cdots A_m \rightarrow B_1 \cdots B_r$, and $B_1 \cdots B_r \rightarrow C_1 \cdots C_p$, then $A_1 \cdots A_m \rightarrow C_1 \cdots C_p$.

Here $A_1, \cdots, A_m, B_1, \cdots, B_r, C_1, \cdots, C_p$ are column names. [Actually, Armstrong's original set of axioms is slightly different from this set, but the two sets are equivalent, since it is easy to check that this set implies each axiom in the original set and that Armstrong's original axioms imply each of these. It turns out that axioms (A1) – (A3) are more convenient for our purposes than Armstrong's original set, so we use (A1) – (A3).]

If DEP is a set of dependency statements, and α is a single dependency statement, then by a "proof of α from DEP via Armstrong's Axioms," we mean a sequence of lines (the "proof"), in which every line is a dependency statement, and the last line is α. Each line of the proof is either a statement in DEP or else is obtained from earlier lines by an application of the axioms. For example, the dependency statement $A_1 \cdots A_m \rightarrow C_1 \cdots C_p$ may be

one line which is obtained from two earlier lines $A_1 \cdots A_m \rightarrow B_1 \cdots B_r$ and $B_1 \cdots B_r \rightarrow C_1 \cdots C_p$ by an application of axiom (A3). Note that axiom (A2) is really two axioms: It says first that the statement $A_1 \cdots A_m \rightarrow B_1 \cdots B_r$ may appear as a line of the proof which is obtained from earlier lines $A_1 \cdots A_m \rightarrow B_i$ $(i = 1, \cdots, r)$; second, a statement $A_1 \cdots A_m \rightarrow B_i$ may appear as a line of the proof which is obtained from an earlier line $A_1 \cdots A_m \rightarrow B_1 \cdots B_r$. Finally, an instance of axiom (A1) can appear no matter what the previous lines of the proof are, since it is "unconditional." To clarify this concept of a proof, let us give a proof via Armstrong's Axioms of $AB \rightarrow D$ from $\{AB \rightarrow C, AC \rightarrow D\}$. (This is the example we used earlier.)

1. $AB \rightarrow C$ (assumption)
2. $AC \rightarrow D$ (assumption)
3. $AB \rightarrow A$ (axiom A1)
4. $AB \rightarrow AC$ (axiom A2 applied to 3 and 1)
5. $AB \rightarrow D$ (axiom A3 applied to 4 and 2) (6)

We can now state Armstrong's Theorem (this was Theorem 5 in [2]). Let DEP be a set of dependency statements, and let DEP′ be the set of dependency statements that can be proved from DEP via Armstrong's Axioms (for convenience, we assume a fixed set of column names). We call DEP′ the *closure* of DEP.

Armstrong's Theorem Let DEP be a set of dependency statements, and DEP′ its closure under Armstrong's Axioms. Then there is a relation $\mathscr{R}$ such that DEP′ is precisely the set of dependency statements that hold for $\mathscr{R}$.

We now present the Dependency Completeness Theorem. This result was never explicitly stated by Armstrong, but, as we will see, it follows very easily from the previous theorem. The Dependency Completeness Theorem says that the following two different concepts are equivalent: 1) α is a consequence of DEP (which means that there is no counterexample relation $\mathscr{R}$ such that every dependency statement in DEP holds for $\mathscr{R}$, but such that α does not hold for $\mathscr{R}$); and 2) α can be proved from DEP via Armstrong's Axioms.

Dependency Completeness Theorem Let DEP be a set of dependency statements and α a single dependency statement. Then α is a consequence of DEP iff α can be proved from DEP via Armstrong's Axioms.

Proof: $\Leftarrow$: This is the "easy direction" of the proof, since each of Armstrong's Axioms is a valid statement about dependency statements. For example, axiom (A3) is valid, since if the first two dependency statements in (A3) hold for a relation $\mathscr{R}$, then also the third dependency statement in (A3) holds for $\mathscr{R}$.

$\Rightarrow$: Assume that α is a consequence of DEP, that is, there is no counterexample relation $\mathscr{R}$ such that every dependency statement in DEP holds for $\mathscr{R}$ but such that α does not hold for $\mathscr{R}$. We want to show that the axioms are powerful enough to prove α from DEP. Let DEP′ be the closure of DEP under Armstrong's Axioms. Clearly $\text{DEP} \subseteq \text{DEP}'$; we wish to show that $\alpha \in \text{DEP}'$. It follows from Armstrong's Theorem that there is a relation $\mathscr{R}$ such that DEP′ is precisely the set of dependency statements that hold for $\mathscr{R}$. Now DEP holds for $\mathscr{R}$, and, by assumption, whenever DEP holds, then α holds. Hence α holds for $\mathscr{R}$, and so $\alpha \in \text{DEP}'$, by construction of $\mathscr{R}$. Therefore, α can be proved from DEP via the axioms. □

We now temporarily turn our attention away from dependency statements and work completely in the realm of propositional logic, to prove the Implicational Completeness Theorem. To prevent notational confusion, we rewrite Armstrong's Axioms in propositional form.

(A1′) $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{A}_i$, for $i = 1, \cdots, m$.

(A2′) $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{B}_1 \cdots \mathbf{B}_r$ iff, for each i, $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{B}_i$.

(A3′) If $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{B}_1 \cdots \mathbf{B}_r$ and $\mathbf{B}_1 \cdots \mathbf{B}_r \Rightarrow \mathbf{C}_1 \cdots \mathbf{C}_p$, then $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{C}_1 \cdots \mathbf{C}_p$.

Here $\mathbf{A}_1, \cdots, \mathbf{A}_m, \mathbf{B}_1, \cdots, \mathbf{B}_r, \mathbf{C}_1, \cdots, \mathbf{C}_p$ are propositional variables.

Implicational Completeness Theorem Let **DEP** be a set of implicational statements of propositional logic and $\boldsymbol{\alpha}$ a single implicational statement. Then $\boldsymbol{\alpha}$ is a logical consequence of **DEP** iff $\boldsymbol{\alpha}$ can be proved from **DEP** via Armstrong's Axioms.

Proof: $\Leftarrow$: Once again, it is easy to verify that each of the axioms are valid statements about implicational statements. For example, axiom (A3′) is valid, since if the first two implicational statements in (A3′) have truth value 1 under a truth assignment t, then also the third implicational statement in (A3′) has truth value 1 under t.

$\Rightarrow$: Assume that $\boldsymbol{\alpha}$ is a logical consequence of **DEP**. We want to show that the axioms are strong enough to prove $\boldsymbol{\alpha}$ from **DEP**. Assume for definiteness that $\boldsymbol{\alpha}$ is $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{D}_1 \cdots \mathbf{D}_s$. Let PROVE be the set of all propositional variables **E** such that the propositional statement $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{E}$ can be proved from **DEP** via the axioms. By axiom (A1′), we know that $\mathbf{A}_1, \cdots, \mathbf{A}_m \in$ PROVE. Our goal is to show that $\mathbf{D}_1, \cdots, \mathbf{D}_s \in$ PROVE, since then by axiom (A2′), the implicational statement $\boldsymbol{\alpha}$ can be proved from **DEP** via the axioms—we simply put together the proofs of the statements $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{D}_i$ and then add a line $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{D}_1 \cdots \mathbf{D}_r$ by applying axiom (A2′); this is a proof of $\boldsymbol{\alpha}$ from **DEP** via the axioms.

Without loss of generality, we just show that $\mathbf{D}_1 \in$ PROVE. Assume that it is false that $\mathbf{D}_1 \in$ PROVE; we will derive a contradiction. Consider the following assignment of truth values to the propositional variables: Each propositional variable in PROVE is assigned the truth value 1, and each remaining propositional variable is assigned the truth value 0. We call this particular truth assignment the *magic* truth assignment. In particular, $\mathbf{D}_1$ is assigned the truth value 0 under the magic truth assignment. Furthermore, as we noted, $\mathbf{A}_1, \cdots, \mathbf{A}_m \in$ PROVE, and hence $\mathbf{A}_1, \cdots, \mathbf{A}_m$ are each assigned the truth value 1. Therefore, $\boldsymbol{\alpha}$ has truth value 0 under the magic truth assignment. We will show that under the magic truth assignment, each implicational statement in **DEP** has truth value 1. Then we will have shown that the magic truth assignment is a counterexample truth assignment (under which every implicational statement in **DEP** has truth value 1 but under which $\boldsymbol{\alpha}$ has truth value 0). However, $\boldsymbol{\alpha}$ is supposed to be a logical consequence of **DEP**, and so there is not supposed to be a counterexample truth assignment. This is a contradiction.

Let $\mathbf{B}_1 \cdots \mathbf{B}_r \Rightarrow \mathbf{C}_1 \cdots \mathbf{C}_p$ be an arbitrary statement in **DEP**. We are through if we can show that this implicational statement has truth value 1 under the magic truth assignment. There are two cases to consider:

Case 1 $\mathbf{B}_1, \cdots, \mathbf{B}_r \in$ PROVE. Hence the implicational statements $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{B}_i$ can be proved from **DEP** via the axioms ($i = 1, \cdots, r$). By now applying axiom (A2′), we see that the implicational statement $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{B}_1 \cdots \mathbf{B}_r$ can be proved from **DEP** via the axioms. Further, since the statement $\mathbf{B}_1 \cdots \mathbf{B}_r \Rightarrow \mathbf{C}_1 \cdots \mathbf{C}_p$ is in **DEP**, it follows from axiom (A3′) that $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{C}_1 \cdots \mathbf{C}_p$. Then by (A2′), the statements $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{C}_i$ are consequences of **DEP**. Hence, $\mathbf{C}_i \in$ PROVE for each i, so each $\mathbf{C}_i$ is assigned truth value 1 by the magic truth assignment. So, the implicational statement $\mathbf{B}_1 \cdots \mathbf{B}_r \Rightarrow \mathbf{C}_1 \cdots \mathbf{C}_p$ has truth value 1 under the magic truth assignment, as desired.

Case 2 At least one of $\mathbf{B}_1, \cdots, \mathbf{B}_r$ is not in PROVE. So at least one of $\mathbf{B}_1, \cdots, \mathbf{B}_r$ is assigned truth value 0. Hence, once again, the implicational statement $\mathbf{B}_1 \cdots \mathbf{B}_r \Rightarrow \mathbf{C}_1 \cdots \mathbf{C}_p$ has truth value 1 under the magic truth assignment, as desired. □

The Equivalence Theorem follows easily from the two Completeness Theorems, as we now see.

Equivalence Theorem Assume that DEP is a set of dependency statements and α is a single dependency statement. Let **DEP**, $\boldsymbol{\alpha}$ be, respectively, the corresponding set of implicational statements and single implicational statement. Then α is a consequence of DEP iff $\boldsymbol{\alpha}$ is a logical consequence of **DEP**.

Proof Assume that α is a consequence of DEP. By the Dependency Completeness Theorem, there is a proof of α from DEP using only Armstrong's Axioms. Hence, there is also a proof of $\boldsymbol{\alpha}$ from **DEP** by using only (the propositional form of) Armstrong's Axioms, since the proof can be obtained by a direct translation of the dependency proof, in which we replace each column name A by its corresponding propositional variable **A**, and in which we replace each occurrence of $\rightarrow$ by $\Rightarrow$. For example, the proof, in (6) where we showed, via Armstrong's axioms, that $AB \rightarrow D$ is a consequence of $\{AB \rightarrow C, AC \rightarrow D\}$, can be converted into the following proof that $\mathbf{AB} \Rightarrow \mathbf{D}$ is a logical consequence of $\{\mathbf{AB} \Rightarrow \mathbf{C}, \mathbf{AC} \Rightarrow \mathbf{D}\}$:

1. $\mathbf{AB} \Rightarrow \mathbf{C}$ (assumption)
2. $\mathbf{AC} \Rightarrow \mathbf{D}$ (assumption)
3. $\mathbf{AB} \Rightarrow \mathbf{A}$ (axiom A1′)
4. $\mathbf{AB} \Rightarrow \mathbf{AC}$ (axiom A2′ applied to 3 and 1)
5. $\mathbf{AB} \Rightarrow \mathbf{D}$ (axiom A3′ applied to 4 and 2)

So, by the "easy direction" of the Implicational Completeness Theorem, we know that $\boldsymbol{\alpha}$ is a logical consequence of **DEP**. We have shown that if α is a consequence of DEP, then $\boldsymbol{\alpha}$ is a logical consequence of **DEP**. Similarly, if $\boldsymbol{\alpha}$ is a logical consequence of **DEP**, then α is a consequence of DEP. □

4. Semantic proof of the Equivalence Theorem

It would be nice if, given a relation $\mathscr{R}$, we could find interpretations for the propositional variables such that, for example, the dependency statement $AB \rightarrow C$ would hold iff the propositional statement $\mathbf{AB} \Rightarrow \mathbf{C}$ had truth value 1. One such possible interpretation of the propositional variables might be to let **A** mean "the tuple's entry in column A has been assigned." Then, the statement $\mathbf{AB} \Rightarrow \mathbf{C}$ would say "If the tuple's entry in column A has been assigned, and if the tuple's entry in column B has been assigned, then the tuple's entry in column C has been assigned." However, this seems difficult to formalize (for example, who assigns the value of an entry of the tuple? Where are the quantifiers?). Another possible approach is to let **A** mean "Tuples 1 and 2 agree in column A." Again, there are difficulties – Are tuples 1 and 2 special tuples? Are they somehow "representative" tuples? We now present a semantic proof of the Equivalence Theorem in which we use this "two-tuple" interpretation of the propositional variables in a precise way, by showing that, roughly speaking, we can restrict our attention to two-tuple relations.

Semantic proof of the Equivalence Theorem Let DEP, α, **DEP**, $\boldsymbol{\alpha}$ be as before. We must show that the following are equivalent:

1. α is a consequence of DEP.
2. $\boldsymbol{\alpha}$ is a logical consequence of **DEP**.

Define a two-tuple relation to be a relation with exactly two tuples (that is, rows). Define "α is a consequence of DEP in the world of two-tuple relations" to mean that α holds in every two-tuple relation that obeys each dependency statement in DEP. That is, α is a consequence of DEP in the world of two-tuple relations iff there is no counterexample two-tuple relation $\mathscr{R}$ such that each dependency statement in DEP holds for $\mathscr{R}$ but such that α does not hold for $\mathscr{R}$. To prove that 1 and 2 above are equivalent, we show first that 1 is equivalent to

3. α is a consequence of DEP in the world of two-tuple relations.

Then we show that 2 is equivalent to 3. It follows that 1 and 2 are equivalent, as desired.

We now show that 1 and 3 are equivalent. It is clear that 1 implies 3. So, we need only show that 3 implies 1. Assume not. Let DEP, α be dependency statements such that 3 holds but not 1. Let $\mathscr{R}$ be a relation (which may contain many tuples) such that each statement in DEP holds for $\mathscr{R}$ but such that α does not hold for $\mathscr{R}$. There is such an $\mathscr{R}$ since 1 fails. Assume for definiteness that α is $A_1 \cdots A_m \rightarrow B_1 \cdots B_r$. Then there are two tuples of $\mathscr{R}$ such that the two tuples agree in columns $A_1 \cdots A_m$, but disagree in one of columns $B_1, \cdots, B_r$. Let $\mathscr{P}$ be a two-tuple relation which contains only these two tuples. It is easy to verify that each statement in DEP holds for $\mathscr{P}$ but that α does not hold for $\mathscr{P}$. This contradicts 3.

We have shown that 1 and 3 are equivalent. We now show that 2 and 3 are equivalent. We need the following lemma.

Semantic Lemma Let t be a truth assignment and $\mathscr{P}$ a two-tuple relation, where t and $\mathscr{P}$ interrelate in the following special way: For each column name A, the two tuples in $\mathscr{P}$ agree in column A iff the corresponding propositional variable **A** is assigned truth value 1 by t. Then the (arbitrary) dependency statement $A_1 \cdots A_m \rightarrow B_1 \cdots B_r$ holds for $\mathscr{P}$ iff the corresponding implicational statement $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{B}_1 \cdots \mathbf{B}_r$ has truth value 1 under truth assignment t.

Proof of lemma Assume first that the dependency statement $A_1 \cdots A_m \rightarrow B_1 \cdots B_r$ holds for $\mathscr{P}$. We will show that the implicational statement $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{B}_1 \cdots \mathbf{B}_r$ has truth value 1 under the truth assignment t. There are two cases, depending on whether or not the two tuples of $\mathscr{P}$ agree in all of the columns $A_1, \cdots, A_m$.

Case 1 The two tuples of $\mathscr{P}$ agree in all of the columns $A_1, \cdots, A_m$. Since the dependency statement $A_1 \cdots A_m \rightarrow B_1 \cdots B_r$ holds for $\mathscr{P}$, it follows that the two tuples agree in columns $B_1, \cdots, B_r$. So, by assumption, each of $\mathbf{B}_1, \cdots, \mathbf{B}_r$ has truth value 1 under truth assignment t. Hence, the implicational statement $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{B}_1 \cdots \mathbf{B}_r$ has truth value 1, as desired.

Case 2 The two tuples of $\mathscr{P}$ disagree in at least one of the columns $A_1, \cdots, A_m$, say in A_1. Then $\mathbf{A}_1$ has truth value 0 under truth assignment t. Hence, once again, the implicational statement $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{B}_1 \cdots \mathbf{B}_r$ has truth value 1.

We have shown that if the dependency statement $A_1 \cdots A_m \rightarrow B_1 \cdots B_r$ holds for $\mathscr{P}$, then the implicational statement $\mathbf{A}_1 \cdots \mathbf{A}_m \Rightarrow \mathbf{B}_1 \cdots \mathbf{B}_r$ has truth value 1 under the truth assignment t. The converse can be proved by a very similar argument.

This concludes the proof of the lemma. We now continue with our proof of the theorem.

We are trying to show that 2 and 3 are equivalent. We show first that 3 implies 2. Assume not. Then α is a consequence of DEP in the world of two-tuple relations, but $\boldsymbol{\alpha}$ is not a logical consequence of **DEP**. Since $\boldsymbol{\alpha}$ is not a logical consequence of **DEP**, there is a truth assignment t to the propositional variables such that every statement in **DEP** has truth value 1 under t, but such that $\boldsymbol{\alpha}$ has truth value 0 under t. Define a two-tuple relation $\mathscr{P}$ for which the column names will be those appearing in DEP and/or α. The first tuple of $\mathscr{P}$ has 1 as every entry. The second tuple of $\mathscr{P}$ has 0 as entry in column A if t assigns truth value 0 to **A**; otherwise, this entry is 1. This procedure is followed for each column name A. It is easy to see that t and $\mathscr{P}$ interrelate as in the hypotheses of the Semantic Lemma. Therefore, since every implicational statement in **DEP** has truth value 1 under t, it follows from the Semantic Lemma that every dependency statement in DEP holds for $\mathscr{P}$. Since α is a consequence of DEP in the world of two-tuple relations, it follows that also α holds for $\mathscr{P}$. So, by the Semantic Lemma again, we know that $\boldsymbol{\alpha}$ has truth value 1 under t. This is a contradiction.

We conclude our proof by showing that 2 implies 3. Assume not. Then $\boldsymbol{\alpha}$ is a logical consequence of **DEP**, but α is not a consequence of DEP in the world of two-tuple relations. Since α is not a consequence of DEP in the world of two-tuple relations, there is a two-tuple relation $\mathscr{P}$ for which each statement in DEP holds but for which α does not hold. Define a new truth assignment t as follows. If the two tuples of $\mathscr{P}$ agree in column A, then t assigns to propositional variable **A** the truth value 1, and otherwise 0. This procedure is followed for each column name A. Once again, t and $\mathscr{P}$ interrelate as in the hypotheses of the Semantic Lemma. So, since each statement in DEP holds for $\mathscr{P}$, it follows from the Semantic Lemma that each statement in **DEP** has truth value 1 under t. Since $\boldsymbol{\alpha}$ is a logical consequence of **DEP**, also $\boldsymbol{\alpha}$ has truth value 1 under t. So by the Seman-

tic Lemma again, α holds for $\mathscr{P}$. But this is a contradiction. □

5. Counterexample to an extension of the Equivalence Theorem

In this section we show that a natural extension of the Equivalence Theorem is false.

Let α and β be dependency statements, and let DEP be a set of dependency statements. Recall that when we say that "α is a consequence of DEP," we mean that α holds for every relation that obeys each dependency statement in DEP. Thus, α is a consequence of DEP iff there is no counterexample relation $\mathscr{R}$ such that each dependency statement in DEP holds for $\mathscr{R}$ but such that α does not hold for $\mathscr{R}$. We similarly define the meaning of "$\alpha \vee \beta$ is a consequence of DEP" to mean that either α or β holds for every relation that obeys each dependency statement in DEP. Thus, $\alpha \vee \beta$ is a consequence of DEP iff there is no counterexample relation $\mathscr{R}$ such that each dependency statement in DEP holds for $\mathscr{R}$ but such that neither α nor β holds for $\mathscr{R}$.

Now let $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ be implicational statements, and **DEP** a set of implicational statements. Recall that when we say that "$\boldsymbol{\alpha}$ is a logical consequence of **DEP**," we mean that $\boldsymbol{\alpha}$ has truth value 1 for every truth assignment that assigns truth value 1 to each implicational statement in **DEP**. Thus, $\boldsymbol{\alpha}$ is a logical consequence of **DEP** iff there is no counterexample truth assignment t such that each implicational statement in **DEP** has truth value 1 under t but such that $\boldsymbol{\alpha}$ has truth value 0 under t. We similarly define the meaning of "$\boldsymbol{\alpha} \vee \boldsymbol{\beta}$ is a logical consequence of **DEP**" to mean that either $\boldsymbol{\alpha}$ or $\boldsymbol{\beta}$ has truth value 1 for every truth assignment that assigns truth value 1 to each implicational statement in **DEP**. Thus, $\boldsymbol{\alpha} \vee \boldsymbol{\beta}$ is a logical consequence of **DEP** iff there is no counterexample truth assignment t such that each implicational statement in **DEP** has truth value 1 under t but such that neither α nor β has truth value 1 under t.

Recall that the Equivalence Theorem states that if DEP, α, **DEP**, $\boldsymbol{\alpha}$ are as before, then α is a consequence of DEP iff $\boldsymbol{\alpha}$ is a consequence of **DEP**. Consider the following fairly natural generalization.

Alleged extension of Equivalence Theorem Assume that DEP is a set of dependency statements and α and β are a pair of dependency statements. Let **DEP**, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ be, respectively, the corresponding set of implicational statements and pair of implicational statements. Then $\alpha \vee \beta$ is a consequence of DEP iff $\boldsymbol{\alpha} \vee \boldsymbol{\beta}$ is a logical consequence of **DEP**.

We now show by example that the alleged extension of the Equivalence Theorem is false. Let DEP contain only the single dependency statement $A \rightarrow A$ (we could just as well have taken DEP to be the empty set in this example, but we choose not to in order to prevent possible confusion). Let α be the dependency statement $A \rightarrow B$, and let β be the dependency statement $B \rightarrow A$. It is *false* that $\alpha \vee \beta$ is a consequence of DEP. That is, there *is* a counterexample relation $\mathscr{R}$ such that each dependency statement in DEP holds for $\mathscr{R}$ but such that neither α nor β holds for $\mathscr{R}$. One such counterexample relation $\mathscr{R}$ is exhibited in Table 1 (as the reader can easily verify).

Table 1 Counterexample.

A	B
0	0
0	1
1	0

The corresponding set **DEP** of implicational statements contains only the single implicational statement $\mathbf{A} \Rightarrow \mathbf{A}$. Further, the corresponding $\boldsymbol{\alpha}$ is the implicational statement $\mathbf{A} \Rightarrow \mathbf{B}$, and $\boldsymbol{\beta}$ is $\mathbf{B} \Rightarrow \mathbf{A}$. We now show that it is *true* that $\boldsymbol{\alpha} \vee \boldsymbol{\beta}$ is a logical consequence of **DEP**. Assume not. Then there is a counterexample truth assignment t such that each (in this case, the only) implicational statement in **DEP** has truth value 1 under t but such that neither $\boldsymbol{\alpha}$ nor $\boldsymbol{\beta}$ has truth value 1 under t. Since $\boldsymbol{\alpha}$ (that is, $\mathbf{A} \Rightarrow \mathbf{B}$) has truth value 0 under t, this means that t assigns truth value 1 to **A** and truth value 0 to **B**. But then β (that is, $\mathbf{B} \Rightarrow \mathbf{A}$) has truth value 1 under t. This is a contradiction.

Thus, we have exhibited DEP, α, β such that 1) $\alpha \vee \beta$ is *not* a consequence of DEP, although 2) $\boldsymbol{\alpha} \vee \boldsymbol{\beta}$ is a logical consequence of **DEP**. Hence, our seemingly mild extension of the Equivalence Theorem fails.

As we now show, our example can also be used to prove the following theorem.

Theorem A There is a set **DEP** of implicational statements and a pair $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ of implicational statements such that simultaneously

a. $\boldsymbol{\alpha}$ is not a logical consequence of **DEP**.
b. $\boldsymbol{\beta}$ is not a logical consequence of **DEP**.
c. $\boldsymbol{\alpha} \vee \boldsymbol{\beta}$ is a logical consequence of **DEP**.

Proof As before, let **DEP** contain only the implicational statement $\mathbf{A} \Rightarrow \mathbf{A}$ (or, even simpler, let **DEP** be the empty set), let $\boldsymbol{\alpha}$ be $\mathbf{A} \Rightarrow \mathbf{B}$, and let $\boldsymbol{\beta}$ be $\mathbf{B} \Rightarrow \mathbf{A}$. We have already shown that c holds. To show that a holds, let t be the truth assignment that assigns truth value 1 to **A** and truth value 0 to **B**. Then t is a counterexample truth assignment that assigns truth value 1 to each implicational statement in **DEP** but that assigns truth value 0 to $\boldsymbol{\alpha}$. Therefore, a holds. We can similarly show that b holds. □

By contrast, we have the following theorem about dependency statements.

Theorem B It is impossible that there is a set DEP of dependency statements and a pair α, β of dependency statements such that simultaneously

a. α is not a consequence of DEP.
b. β is not a consequence of DEP.
c. $\alpha \vee \beta$ is a consequence of DEP.

Proof Assume that there exist DEP, α, β such that a, b, and c all hold simultaneously. Let DEP′ be the closure of DEP under Armstrong's Axioms. By Armstrong's Theorem, there is a relation $\mathscr{R}$ such that DEP′ is precisely the set of dependency statements that hold for $\mathscr{R}$. Since DEP holds for $\mathscr{R}$, it follows from c that either α or β holds for $\mathscr{R}$. Assume that α holds for $\mathscr{R}$; we will derive a contradiction. (Similarly, the assumption that β holds for $\mathscr{R}$ leads to a contradiction.) Since α holds for $\mathscr{R}$, it follows by definition of $\mathscr{R}$ that α is in DEP′. By the Dependency Completeness Theorem, DEP′ is the set of dependency statements that are consequences of DEP. Therefore, since α is in DEP′, it follows that α is a consequence of DEP. This contradicts a. □

Under the terminology of Beeri, Fagin, and Howard [3], Theorem A shows that Armstrong's Axioms are not "strongly complete" for implicational statements, although, by the Dependency Completeness Theorem, they are complete for implicational statements. By contrast, Armstrong's Axioms *are* strongly complete for dependency statements.

6. The Delobel-Casey Theorems

In this section, we show that the Delobel-Casey Relational Database Decomposition Theorems, which heretofore have seemed somewhat unexpected and surprising, are natural consequences of the Equivalence Theorem.

Let $A_1 \cdots A_m \rightarrow B_1 \cdots B_r$ be a typical dependency statement. The *first Delobel-Casey transform* of this dependency statement is the propositional (or Boolean) statement $\mathbf{A_1 \cdots A_m B_1' + \cdots + A_1 \cdots A_m B_r'}$. Here $\mathbf{B_i'}$ is the negation of $\mathbf{B_i}$, and "+" is the "logical or" (or Boolean sum). Thus, this propositional statement has truth value 1 iff first, $\mathbf{A_1, \cdots, A_m}$ each have truth value 1, and second, for some i it happens that $\mathbf{B_i}$ has truth value 0. If DEP is a *set* of dependency statements, then the first Delobel-Casey transform of the set DEP is the propositional statement which is the Boolean sum of the first Delobel-Casey transforms of each of its members. For example, if DEP is {AB → CD, C → A}, then the first Delobel-Casey transform of DEP is **ABC′ + ABD′ + CA′**.

The first Delobel-Casey Theorem relates the equivalence of two sets of dependency statements to the equivalence of the corresponding first Delobel-Casey transforms. We will now look at an example, which is taken from Delobel and Casey's paper [4]. Consider the following set DEP_1 of dependency statements:

P → T

PH → Y

PH → N

HN → P

HN → Y

HY → P

HY → N

The first Delobel-Casey transform of this set is the Boolean expression $BOOL_1$ given by

PT′ + PHY′ + PHN′ + HNP′ + HNY′ + HYP′ + HYN′.

By using Karnaugh maps, Delobel and Casey show that this Boolean expression $BOOL_1$ is equivalent to the Boolean expression $BOOL_2$ given by

PT′ + HYT′ + HYTN′ + PTHN′ + NHTY′ + NYTHP′ + NHT′.

This expression is the first Delobel-Casey transform of the following set DEP_2 of dependency statements:

P → T

HY → T

HYT → N

PTH → N

NHT → Y

NYTH → P

NH → T

The First Delobel-Casey Theorem tells us that because $BOOL_1$ and $BOOL_2$ are equivalent Boolean expressions, it follows that the sets DEP_1 and DEP_2 are equivalent sets of dependency statements (DEP_1 and DEP_2 are said to be equivalent if each statement in DEP_1 is a consequence of the set DEP_2 and each statement in DEP_2 is a consequence of the set DEP_1).

First Delobel-Casey Theorem Let DEP_1 and DEP_2 be sets of dependency statements and let $BOOL_1$ and $BOOL_2$ be the first Delobel-Casey transforms. Then DEP_1 is equivalent to DEP_2 iff $BOOL_1$ is equivalent to $BOOL_2$.

Proof Let $\mathbf{DEP_1}$ be the set of implicational statements which correspond to DEP_1 as before, in which we replace each column name A by its corresponding propositional

variable **A**, and in which we replace each occurrence of $\rightarrow$ by $\Rightarrow$. Similarly, define $\mathbf{DEP}_2$. It is straightforward to check that the conjunction of the implicational statements in $\mathbf{DEP}_1$ is equivalent to the negation of BOOL_1; similarly for $\mathbf{DEP}_2$ and BOOL_2. It follows easily that BOOL_1 is equivalent to BOOL_2 iff $\mathbf{DEP}_1$ is equivalent to $\mathbf{DEP}_2$. So, to prove the theorem, we need only prove that DEP_1 is equivalent to DEP_2 iff $\mathbf{DEP}_1$ is equivalent to $\mathbf{DEP}_2$. But this follows from the Equivalence Theorem, as we will show. Actually, we will only show that if DEP_1 is equivalent to DEP_2, then $\mathbf{DEP}_1$ is equivalent to $\mathbf{DEP}_2$; the proof of the converse is very similar. Assume that DEP_1 is equivalent to DEP_2. To show that $\mathbf{DEP}_1$ is equivalent to $\mathbf{DEP}_2$, we must show that each implicational statement in $\mathbf{DEP}_1$ is a logical consequence of $\mathbf{DEP}_2$ and that each implicational statement in $\mathbf{DEP}_2$ is a logical consequence of $\mathbf{DEP}_1$. Without loss of generality, we will only show that each implicational statement in $\mathbf{DEP}_2$ is a logical consequence of $\mathbf{DEP}_1$. Let α be an arbitrary implicational statement in $\mathbf{DEP}_2$; we must show that α is a logical consequence of $\mathbf{DEP}_1$. Let α be the dependency statement in DEP_2 which corresponds to the implicational statement α in $\mathbf{DEP}_2$. Since DEP_2 is equivalent to DEP_1, it follows that α (like every other dependency statement in DEP_2) is a consequence of DEP_1. So, by the Equivalence Theorem, α is a logical consequence of $\mathbf{DEP}_1$, which was to be shown. □

We now discuss the second Delobel-Casey Theorem. Assume that $\mathrm{K}_1, \cdots, \mathrm{K}_m$ are some (or all) of the column names of relation $\mathscr{R}$. We say that $\{\mathrm{K}_1, \cdots, \mathrm{K}_m\}$ (or, more simply, $\mathrm{K}_1 \cdots \mathrm{K}_m$) is a *key* of $\mathscr{R}$ if no two distinct tuples of $\mathscr{R}$ agree in all of the columns $\mathrm{K}_1, \cdots, \mathrm{K}_m$. Thus, $\mathrm{K}_1 \cdots \mathrm{K}_m$ is a key iff $\mathrm{K}_1 \cdots \mathrm{K}_m \rightarrow \mathrm{A}$ for each column name A. (We are tacitly assuming that the same tuple does not appear twice in relation $\mathscr{R}$.) For convenience, we are allowing the possibility that a proper subset of a key be a key (our definition of key corresponds to Bernstein's [6] definition of "superkey.")

If $\mathrm{D}_1, \cdots, \mathrm{D}_n$ are *all* of the column names of $\mathscr{R}$, and if DEP is a set of dependency statements (involving only column names $\mathrm{D}_1, \cdots, \mathrm{D}_n$), then by the *second Delobel-Casey transform* of DEP, we mean the propositional statement which is the Boolean sum of $\mathbf{D}_1 \cdots \mathbf{D}_n$ and the first Delobel-Casey transform of DEP. For example, if DEP is $\{\mathrm{AB} \rightarrow \mathrm{CD}, \mathrm{C} \rightarrow \mathrm{A}\}$, and A, B, C, D are all of the column names of $\mathscr{R}$, then the second Delobel-Casey transform of DEP is $\mathbf{ABCD} + \mathbf{ABC}' + \mathbf{ABD}' + \mathbf{CA}'$.

Second Delobel-Casey Theorem The following are equivalent:

1. It is a consequence of DEP that $\mathrm{K}_1 \cdots \mathrm{K}_m$ is a key.
2. The propositional statement $\mathbf{K}_1 \cdots \mathbf{K}_m$ logically implies the second Delobel-Casey transform of DEP.

Note In 2 above, when we say that one propositional statement logically implies a second, we mean that the second statement has truth value 1 for every truth assignment that assigns truth value 1 to the first propositional statement.

Proof Statement 1 is equivalent to the assertion that the dependency statement $\mathrm{K}_1 \cdots \mathrm{K}_m \rightarrow \mathrm{D}_1 \cdots \mathrm{D}_n$ is a consequence of DEP, where $\mathrm{D}_1, \cdots, \mathrm{D}_n$ are all of the column names. Hence, by the Equivalence Theorem, 1 holds iff the implicational statement $\mathbf{K}_1 \cdots \mathbf{K}_m \rightarrow \mathbf{D}_1 \cdots \mathbf{D}_n$ is a logical consequence of the set **DEP** (of propositional statements) which corresponds to DEP. What about 2? It is straightforward to verify that the second Delobel-Casey transform of DEP is logically equivalent to the propositional statement $\beta \Rightarrow \mathbf{D}_1 \cdots \mathbf{D}_n$, where β is the conjunction of the propositional statements in **DEP**. So, 2 says that γ logically implies the statement $\beta \Rightarrow \delta$, where γ is $\mathbf{K}_1 \cdots \mathbf{K}_m$, where β is the conjunction of the statements in **DEP**, and where δ is $\mathbf{D}_1 \cdots \mathbf{D}_n$. In general, "$\gamma$ logically implies the statement $\beta \Rightarrow \delta$" holds iff "β logically implies the statement $\gamma \Rightarrow \delta$"; this can easily be verified by considering each of the $2^3 = 8$ possible truth assignments to (β, γ, δ). But in this case, as we showed, the sentence "β logically implies the statement $\gamma \Rightarrow \delta$," i.e., "**DEP** logically implies the statement $\mathbf{K}_1 \cdots \mathbf{K}_m \Rightarrow \mathbf{D}_1 \cdots \mathbf{D}_n$," is equivalent to 1. So 1 and 2 are equivalent. □

We close this section with remarks on earlier proofs of the Delobel-Casey Theorems. Delobel and Casey's original proofs are somewhat involved and contain case-by-case examination of the effect of the "star algorithm" for generating prime implicants of disjunctive Boolean formulas. Armstrong [2] gave another proof in which he interprets the propositional variable **A** corresponding to column name A as a certain Boolean function of Boolean functions. Hopefully, our proof eliminates some of the mystery.

7. Multivalued dependency statements

The main result of this paper is that the relational database concept of "determines" (where the dependency statement $\mathrm{A} \rightarrow \mathrm{B}$ is read "A determines B") has some of the same formal properties as the propositional concept of "implies." We remark that the author has defined another natural kind of relational database dependency, called "multivalued dependency" [12], which has quite different formal properties (although the dependency statements dealt with in the present paper turn out to be a special case). A complete axiomatization for multivalued dependency statements is given in Beeri, Fagin, and Howard [3]. Of course, this axiomatization is different from that given by Armstrong's Axioms.

8. Summary

We have demonstrated an equivalence between dependency statements (or functional dependencies) of a relational database on the one hand and of implicational statements of propositional logic on the other hand. We have exploited this equivalence to prove the Delobel-Casey Relational Database Decomposition Theorems. This equivalence may also be of use to a database designer, who can use the tools of propositional logic to answer questions about dependency statements. We have presented a detailed algorithm for such an application. Furthermore, we have demonstrated an example that shows that an apparently mild extension of the equivalence fails.

Acknowledgments

The author is grateful to R. G. Casey for reading the paper and providing useful comments, and to J.-M. Cadiou and J. H. Howard for helpful discussions.

References

1. E. F. Codd, "Further Normalization of the Data Base Relational Model," *Courant Computer Science Symposia 6, Data Base Systems*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1971, p. 33.
2. W. W. Armstrong, "Dependency Structures of Data Base References," *IFIP Conf. Proc. 1974*, North Holland Publishing Co., Amsterdam, 1974, p. 580.
3. C. Beeri, R. Fagin, and J. H. Howard, "A Complete Axiomatization for Functional and Multivalued Dependencies in Database Relations," *Proc. ACM SIGMOD*, D. C. P. Smith, ed., New York, NY, 1977, p. 47.
4. C. Delobel and R. G. Casey, "Decomposition of a Data Base and the Theory of Boolean Switching Functions," *IBM J. Res. Develop.* **17**, 374 (1973).
5. J. Rissanen, "Independent Components of Relations," *ACM Trans. Database Syst.*, to be published.
6. P. A. Bernstein, "Synthesizing Third Normal Form Relations from Functional Dependencies," *ACM Trans. Database Syst.* **1**, 277 (1976).
7. R. Fagin, "The Decomposition Versus the Synthetic Approach to Relational Database Design," *Proceedings of the 1977 Very Large Data Bases Conference*, to be published.
8. A. Horn, "On Sentences Which are True of Direct Unions of Algebras," *J. Symbol. Logic* **16**, 14 (1951).
9. L. Henschen and L. Wos, "Unit Refutations and Horn Sets," *J. ACM* **21**, 590 (1974).
10. D. Kuehner, "Some Special Purpose Resolution Systems," *Machine Intelligence*, Vol. 7, B. Meltzer and D. Michie, eds., American Elsevier, New York, 1972, p. 117.
11. C. L. Chang, "DEDUCE—A Deductive Query Language for Relational Data Bases," *Pattern Recognition and Artificial Intelligence*, C. H. Chen, ed., Academic Press, Inc., New York, 1976, p. 108.
12. R. Fagin, "Multivalued Dependencies and A New Normal Form for Relational Databases," *ACM Trans. Database Syst.* **2**, 262 (1977).

Received May 6, 1977; revised July 8, 1977

The author is located at the IBM Research Division laboratory, 5600 Cottle Road, San Jose, California 95193.