The Relational Model of Data and Cylindric Algebras

TOMASZ IMIELiNSKI AND WITOLD LIPSKI, JR.

Institute of Computer Science, Polish Academy of Sciences
P.O. Box 22, 00-901 Warsaw PKiN, Poland

Received August 24, 1982; revised April 25, 1983

It is shown how the theory of cylindric algebras (a notion introduced by Tarski and others as a tool in the algebraization of the first order predicate calculus) can give a new insight into Codd's relational model of data. The relational algebra of Codd can be embedded in a natural way into a cylindric algebra where the join operation becomes the usual set-theoretical intersection. It is shown, by using known facts from the theory of cylindric algebras, that a version of the relational algebra is not finitely axiomatizable and that the equivalence problem for certain relational expressions is undecidable. A duality between the project-join and select-union operator pairs is also briefly discussed.

1. INTRODUCTION

A general classification of query languages which turned out to be convenient in database theory and practice is that into nonprocedural and procedural languages. A query in a nonprocedural language expresses what we want without necessarily saying how to obtain it. A query in a procedural language explicitly specifies the actions that should be taken, or procedures to be invoked, to obtain the response. While no practical query language is either purely procedural or purely nonprocedural, it is clear that nonprocedural languages are more convenient for the user, whereas procedural languages are easier to implement.

In this paper we shall be concerned with the relational model of data [Cod 1] (see also [Ull]). In this model data are presented in the form of a finite collection of relations, or tables, with columns corresponding to attributes. In the context of the relational model, a typical nonprocedural query language is the relational calculus, which is based on the predicate calculus (especially when formulated as a "domain calculus," with variables ranging over attribute domains, see [Ull]), and a typical procedural language is the relational algebra [Cod 2] (see also [Ull]). (Note: All predicate calculi considered in this paper are assumed not to contain any function symbols.)

Example. Let a database consist of the following two tables:

PART, with columns (attribute names), P (part name) and N (supplier name); and SUPPLIER, with columns, N and L (location). Below is an instance of our database:
Suppose we want to ask the query "Give the names of all parts available in London." This query is expressed in the relational calculus as

\[ \{x: \exists y (\text{PART}(x, y) \land \text{SUPPLIER}(y, \text{London})) \} \]

and in the relational algebra as

\[ \pi_p (\text{PART} \bowtie_{\text{LONDON}} (\text{SUPPLIER})) \]

where \( \pi_p \), \( \sigma_{\text{LONDON}} \), and \( \bowtie \) are the relational operators of projection, selection, and join, respectively (see Section 3). Notice that the second query, unlike the first one, explicitly specifies a sequence of operations to be performed over the tables PART and SUPPLIER in order to obtain the response.

One of the early results in relational database theory is Codd’s completeness theorem [Cod 2] (see also [Ull]), which asserts that the relational calculus and relational algebra are equivalent in expressive power, i.e., for any query expressed in the relational calculus there is a semantically equivalent query formulated in the relational algebra, and vice versa (actually, Codd was concerned only with the first part of this theorem).

By this equivalence, the relational algebra can be treated as an “algebraic version” of the predicate calculus.

A fact that seems to have been overlooked by researchers in the relational database theory is that there has been extensive research in mathematical logic concerning the algebraization of the (first-order) predicate calculus. This research led Alfred Tarski to define, about 1952, the notion of a cylindric algebra. Cylindric algebras bear the same relation to the predicate calculus as Boolean algebras bear to the propositional...
calculus. In this way, the inquiries into the algebraization of the predicate calculus can be treated as a direct extension of the work of George Boole who algebraized the propositional calculus in the middle of the Nineteenth Century. Roughly speaking, a cylindric algebra is a Boolean algebra with some additional unary operations called *cylindrifications*, corresponding to existential quantification. The theory of cylindric algebras is now a well-developed branch of mathematical logic and algebra (see, e.g., [HMT, HMTAN, HT]).

We show in this paper that the relational algebra can be treated as a disguised version of a so-called cylindric set algebra. This makes the relational algebra an especially simple algebraic structure: disregarding the projection operation, it may be treated as an ordinary Boolean set algebra (with the join operation becoming the usual set-theoretical intersection).

The paper is organized as follows: Section 2 contains basic definitions and notation concerning cylindric algebras. The relation between cylindric algebras and various predicate calculi is also explained. In Section 3, the relational algebra is precisely defined, and it is shown how it can be embedded into a cylindric set algebra. Section 4 is devoted to axiomatizability and decidability problems. By using results known in the theory of cylindric algebras we show that a certain version of the relational algebra (with the difference operation) is not finitely axiomatizable, and that the equivalence problem for certain relational expressions is undecidable. The paper is concluded by some remarks on the duality between the project-join and select-union operator pairs.

2. CYLINDRIC ALGEBRAS

Informally, a cylindric algebra is a Boolean algebra with additional unary operations $c_i, i \in I$, called *cylindrifications*, and constants (i.e., distinguished elements of the algebra) $d_{ij}, i,j \in I$, called *diagonals*, such that certain natural conditions (axioms (C1)-(C7)) are satisfied. We shall be mainly concerned with the case where $I$ is finite, and we shall call $n = |I|$ the *dimension* of the cylindric algebra (in the general case, $I$ may be any ordinal number). The axioms of a cylindric algebra include familiar properties of existential quantification (with each cylindrification corresponding to quantification with respect to a different variable), and of equalities between variables (these equalities are represented by the diagonals). For example, axiom (C4) corresponds to the fact that $\exists v_1 \exists v_j \phi$ is equivalent to $\exists v_1 \exists v \phi$, for any formula $\phi$, etc.

Formally, by a *cylindric algebra of dimension* $n$ $(CA_n)$ we mean any algebraic structure

$$\mathbb{A} = \langle A, +, \cdot, -, 0, 1, c_i, d_{ij} \rangle, \quad 1 \leq i, j \leq n,$$

such that $0, 1,$ and $d_{ij}, 1 \leq i, j \leq n$ are distinguished elements of $A$; $-$ and $c_i$,
1 \leq i \leq n$, are unary operations; $+$ and $\cdot$ are binary operations; and the following conditions are satisfied for any $x, y \in A$ and any $i, j, k$ ($1 \leq i, j, k \leq n$):

(C0) \quad \langle A, +, \cdot, -, 0, 1 \rangle is a Boolean algebra,
(C1) \quad c_i 0 = 0,
(C2) \quad x \leq c_i x \text{ (i.e., } x \cdot c_i x = x),
(C3) \quad c_i (x \cdot c_i y) = c_i x \cdot c_i y,
(C4) \quad c_i c_j x = c_j c_i x,
(C5) \quad d_{ii} = 1,
(C6) \quad \text{if } i \neq j \text{ and } i \neq k \text{ then } d_{jk} = c_i (d_{ij} \cdot d_{ik}),
(C7) \quad \text{if } i \neq j \text{ then } c_i (d_{ij} \cdot x) \cdot c_i (d_{ij} \cdot -x) = 0.

Notice that for any $n \geq 0$ the class of all $CA_n$'s is \textit{equationally definable}, i.e., it consists of exactly those structures (2.1) in which a fixed set of equalities is identically satisfied. In fact, in our case this set of equalities can be chosen finite (if $n$ is finite), since Boolean algebras can be characterized by a finite set of equalities.

It can easily be shown [HMT] that the following are logical consequences of (CO)-(C7):

\begin{align*}
c_i x = 0 \text{ iff } x = 0, & \quad (2.2) \\
c_i 1 = 1, & \quad (2.3) \\
c_i c_i x = c_i x, & \quad (2.4) \\
x \cdot c_i y = 0 \text{ iff } y \cdot c_i x = 0, & \quad (2.5) \\
c_i (x + y) = c_i x + c_i y, & \quad (2.6) \\
x \leq c_i y \text{ iff } c_i x \leq c_i y, & \quad (2.7) \\
x = c_i x \text{ iff } -x = c_i - x, & \quad (2.8) \\
c_i x \cdot c_j y = 0 \text{ iff } c_j x \cdot c_i y = 0, & \quad (2.9) \\
d_{ij} = d_{ji}. & \quad (2.10)
\end{align*}

An example of a $CA_n$ which is most important for our purposes is a \textit{cylindric set algebra of dimension $n$} ($Cs_n$). By a $Cs_n$ with \textit{base} $X$ ($X \neq \emptyset$) we mean any structure

\[ A = \langle A, \cup, \cap, -, \emptyset, X^n, C_i, D_{ij} \rangle, \quad 1 \leq i, j \leq n, \]

where $\langle A, \cup, \cap, -, \emptyset, X^n \rangle$ is a Boolean algebra of (not necessarily all) subsets of $X^n$ containing all the diagonals $D_{ij}$ defined by

\[ D_{ij} = \{ x \subset X^n : x_i = x_j \}, \]

and closed under all the operations $C_i$ defined by

\[ C_i R = \{ y \in X^n : y(i/a) \in R \text{ for some } a \in X \}, \]
where \( y(i/a) \) denotes the tuple in \( X^n \) defined by \( y(i/a)_i = a \) and \( y(i/a)_j = y_j \) for all \( j \neq i \); i.e., \( y(i/a) = (y_1, \ldots, y_{i-1}, a, y_{i+1}, \ldots, y_n) \).

The easy verification of the fact that axioms (C0)–(C7) hold true in every \( Cs_n \) is left to the reader. Notice that elements of \( Cs_n \) are subsets of \( X^n \), i.e., \( n \)-ary relations, and that \( C_iR \) can be treated as a “cylinder” obtained by a translation of the “base” \( R \) parallel to the \( i \)-th axis of the Cartesian space \( X^n \) (hence the name of the algebra).

By a diagonal-free cylindric algebra of dimension \( n \) (\( Df_n \)), we mean any algebraic structure

\[
\mathbb{A} = (A, +, \cdot, -, 0, 1, c_i), \quad 1 \leq i \leq n
\]

satisfying conditions (C0)–(C4). The notion of a diagonal-free cylindric set algebra of dimension \( n \) (\( Dfs_n \)) is defined in a similar way as that of a \( Cs_n \): Let \( X_1, \ldots, X_n \) be nonempty sets, and let \( 1 = X_1 \times \cdots \times X_n \). By a \( Dfs_n \) (of subsets of \( 1 \)) we mean any Boolean algebra of subsets of \( 1 \) closed under operations \( C_i \) defined by

\[
C_iR = \{ y \in 1 : y(i/a) \in R \text{ for some } a \in X_i \}. \quad (2.11)
\]

Clearly, any \( Dfs_n \) is a \( Df_n \). Also every \( CA_n \) defines a \( Df_n \) obtained simply by “forgetting the \( d_{ij} \).” Notice, however, that a given \( Dfs_n \), \( \mathbb{A} \) of subsets of \( X^n \) is not necessarily extendable to a \( Cs_n \) by adding the \( D_{ij}, 1 \leq i, j \leq n \), since the diagonals need not be in \( A \).

We shall now show the connection between cylindric set algebras and first-order predicate calculus. In fact, we consider only a very simple form of this connection, one which is the most suitable in the context of the relational model of data (see [HMT] for more detail). Consider a version of an “\( n \)-sorted” predicate calculus where we have only \( n \) variables \( v_1, \ldots, v_n \), one for each sort, and a certain number of predicate symbols \( P_1, P_2, \ldots \). We assume that for any \( i, P_i \) is of type \( a_i = (a_{i1}, \ldots, a_{in_i}), 1 \leq a_{i1} < \cdots < a_{in_i} \leq n \), which means that \( P_i \) is an \( n_i \)-ary predicate symbol with the \( j \)-th argument of sort \( a_{ij}, j = 1, \ldots, n_i \). Let \( M \) be a fixed relational structure for our predicate calculus; i.e., \( M \) consists of nonempty sets \( X_1, \ldots, X_n \), one for each sort, and a relation \( R_i \subseteq X_{a_{i1}} \times \cdots \times X_{a_{in_i}} \) for each predicate symbol \( P_i \). By a valuation (of variables) we shall mean any function \( f : \{v_1, \ldots, v_n\} \to \bigcup_{i=1}^n X_i \) such that \( f(v_i) \in X_i \), \( 1 \leq i \leq n \). Let us denote \( 1 = X_1 \times \cdots \times X_n \). Clearly, any valuation \( f \) can be identified with a tuple \( \langle x_1, \ldots, x_n \rangle \in 1 \) (\( x_i = f(v_i) \)). Let \( \phi \) be any well-formed formula built up from atomic formulas of the form \( P_i(v_{a_{i1}}, \ldots, v_{a_{in_i}}) \) by means of the logical connectives \( \lor, \land, \neg \), and quantifiers \( \exists v_i \) (\( 1 \leq i \leq n \)). The classical Tarski’s definition [Tar 1] of what it means that “formula \( \phi \) is true in \( M \) under valuation \( \langle x_1, \ldots, x_n \rangle \)” (in symbols, \( M \models \phi[x_1, \ldots, x_n] \)) can be found in any textbook on mathematical logic (see, e.g., [Sho]). Denoting by \( \| \phi \|_M \) (or just \( \| \phi \| \)) the set of all valuations which make \( \phi \) true in \( M \), i.e.,

\[
\| \phi \| = \{ \langle x_1, \ldots, x_n \rangle \in 1 : M \models \phi[x_1, \ldots, x_n] \},
\]
this definition can be expressed in the following inductive form:

\[ P_{i}(v_{a_{i}},...,v_{a_{m}}) = \{ (x_{1},...,x_{n}) \in \mathcal{I} : (x_{a_{i}},...,x_{a_{m}}) \in R_{i} \} \]  
\[ \phi \lor \psi = \| \phi \| \cup \| \psi \| , \]  
\[ \phi \land \psi = \| \phi \| \cap \| \psi \| , \]  
\[ \neg \phi = -\| \phi \| \]  

(“¬” is the complement with respect to \( \mathcal{I} \)),

\[ \| \exists v_{i} \phi \| = \{ (x_{1},...,x_{n}) \in \mathcal{I} : \text{there exists a } z \in X_{i} \text{ such that } M \models \phi[x_{1},...,x_{i-1},z,x_{i+1},...,x_{n}] \} \]

= \{ x \in \mathcal{I} : x(i/z) \in \| \phi \| \text{ for some } z \in X_{i} \}. \]

By (2.11), the last set is exactly \( C_{i} \| \phi \| \), so that

\[ \| \exists v_{i} \phi \| = C_{i} \| \phi \| . \]  

This is shown in Fig. 1 for the case \( n = 2 \). Notice that although \( v_{2} \) is bound in \( \exists v_{2} \phi \), and the truth value of \( \exists v_{2} \phi \) does not depend on the value assigned to \( v_{2} \), nevertheless the members of \( \| \exists v_{2} \phi \| \) are “total” valuations, defined on \( \{ v_{1}, v_{2} \} \). The fact that the truth value of \( \exists v_{2} \phi \) is independent of the value assigned to \( v_{2} \) is reflected by the “cylindric shape” of \( \| \exists v_{2} \phi \| \).

In this way the logical connectives and quantifiers \( \exists v_{i} \) can be interpreted as the corresponding operations in the Dfs, of subsets of \( \mathcal{I} \) generated by all subsets of the form (2.12). In a similar way, the usual (one-sorted) predicate calculus with equality corresponds to cylindric set algebras (with diagonals). Let \( V \) be a set of variables, and let \( M \) be a fixed relational structure for our predicate calculus. \( M \) consists of a

\[ \text{Fig. 1. The operation of cylindrification as corresponding to existential quantification.} \]
nonempty universe $X$ and of a relation $R \subseteq X^{a(P)}$ for every predicate symbol $P$ of our predicate calculus ($a(P)$ denotes the arity of $P$). Similarly as before, we define

$$\|P(v_1, \ldots, v_k)\| = \{f \in X^k : \langle f(v_1), \ldots, f(v_k) \rangle \in R\},$$  \hspace{1cm} (2.17)

where $k = a(P)$, $R$ is the relation corresponding to $P$, and $X^k$ denotes the set of all functions $f : V \rightarrow X$, i.e., of all valuations,

$$\| \phi \lor \psi \| = \| \phi \| \cup \| \psi \|,$$

$$\| \phi \land \psi \| = \| \phi \| \cap \| \psi \|,$$

$$\| \neg \phi \| = -\| \phi \| \quad \text{(complement with respect to } X^v),$$

$$\| \exists \psi \| = C_v \| \phi \|,$$

where for any $A \subseteq X^v$

$$C_v A = \{f \in X^v : f(v/a) \in A \text{ for some } a \in X\}$$

and

$$f(v/a)(u) = f(u) \quad \text{if } u \neq v,$$

$$= a \quad \text{if } u = v.$$  

Finally,

$$\| v = w \| = \{f \in X^v : f(v) = f(u)\} = D_{v,v}.$$  \hspace{1cm} (2.18)

 Typically, the set of variables $V$ is infinite and is given as a sequence (of type $\omega$) $v_0, v_1, v_2, \ldots$. This situation leads to the notion of an \textit{infinite dimensional} (more exactly, $\omega$-\textit{dimensional}) cylindric algebra $\text{CA}_\omega$. The definition of a $\text{CA}_\omega$ ($\text{CS}_\omega$, etc.) is exactly the same as that of a $\text{CA}_n$ (resp. $\text{CS}_n$, etc.), except for the fact that there are infinitely many cylindrifications and diagonals corresponding to all variables and all pairs of variables, respectively. For any $A \subseteq X^v$, the \textit{dimension set} of $A$ is defined by

$$\Delta A = \{v \in V : C_v A \neq A\}.$$  \hspace{1cm} (2.19)

It is easily seen that for every element $A$ of our cylindric set algebra corresponding to a predicate calculus, $\Delta A$ is finite (this reflects the fact that any formula depends only on a finite number of variables). Cylindric algebras such that the dimension set of every element is finite are called \textit{locally finite dimensional} (or just \textit{locally finite}).

3. EMBEDDING THE RELATIONAL ALGEBRA INTO A CYLINDRIC SET ALGEBRA

We begin by recalling some basic definitions concerning the relational model of data [Cod 1, Ull].
Throughout the paper $\mathcal{A}$ will be a fixed set of attributes. Attributes will usually be denoted by $A, B, C,$ and sets of attributes by $X, Y, Z.$ We shall usually write $\{A, B\}$ as $AB,$ etc. Associated with every $A \in \mathcal{A}$ is a nonempty attribute domain $D(A).$ By a relation of type $X$ we mean any subset $R \subseteq \mathcal{A}_{A \in X} D(A);$ any element $t$ of $R$ is called a tuple (of type $X$). For such a relation $R$ and tuple $t$ we write $a(R) = a(t) = X$ (formally, we consider a different empty relation of type $X$ for every $X$). In other words, a tuple of type $X$ is a mapping which associates a value $t(A) \in D(A)$ with every $A \in X.$ For any $Y \subseteq X,$ a restriction of this mapping to $Y$ is denoted by $t[Y].$

In database theory, relations are usually assumed to be finite, and any finite relation of type $X$ is represented in the natural way by a table with columns corresponding to attributes in $X,$ and with rows corresponding to tuples. We shall consider the following basic relational operations.

**Projection.** $\pi_Y(R) = \{t[Y] : t \in R\} (Y \subseteq a(R)).$

**Selection.** $\sigma_E(R) = \{t \in R : E(t) = \text{true}\}.$

Here $E$ is a selection condition, usually an expression built up from atomic conditions of the form $(A = a), (A = B), A, B \in \mathcal{A}, a \in D(A)$ by means of the logical connectives $\lor$ (or), $\land$ (and), $\neg$ (not). The value $E(t)$ is defined in the natural way: we replace all occurrences of every $A \in \mathcal{A}$ in $E$ by $t(A),$ which reduces every atomic condition to true or false, and then we apply the logical connectives.

It is assumed that all attributes occurring in $E$ are in $a(R)$.

**Union.** $R \cup Q,$ i.e., the usual set-theoretical union.

**Difference.** $R - Q = R \setminus Q,$ i.e., the usual set-theoretical difference. In the case of union and difference we always assume that $a(R) = a(Q)$.

**Join** (natural join). $R \bowtie Q = \{t : a(t) = X \cup Y \land t[X] \in R \land t[Y] \in Q\},$ where $X = a(R), Y = a(Q).$

It should be remarked here that the term "relational algebra is used in this paper in a rather general and informal way, and some of the versions of relational algebra considered are not "relationally complete" in the sense of Codd [Cod 2], i.e., they are strictly weaker in expressive power than the usual predicate calculus. An example of such a "relationally incomplete" version is the relational algebra based on a fixed finite set of attributes and the relational operations described above. We shall return to this problem later on.

Let $R$ be any relation, let $X = a(R),$ and let us denote $1 = X, E \in P$ $D(A).$ We define

$$h(R) = \{t \in 1 : t[X] \in R\}. \quad (3.1)$$

In other words, $h(R)$ is obtained by extending in all possible ways each tuple in $R$ to the whole set of attributes. We shall treat $h(R)$ as an element of a $\mathfrak{Dfs}_n$ of subsets of $1,$ with cylindrifications corresponding to attributes in $\mathcal{A}$ ($n = |\mathcal{A}|$).

It turns out that the mapping $h$ defines a natural embedding of the relational algebra into the $\mathfrak{Dfs}_n$ of subsets of $1.$ (Here, as in the case of any embedding of one algebraic structure into another, an embedding should preserve in some way the operations; this is usually formalized by requiring that the embedding be a
homomorphism). More exactly, we have the following theorem, where for any \( X = \{A_1, \ldots, A_k\} \subseteq \mathcal{U} \) and any \( R \subseteq \mathcal{U} \), we denote

\[ C_X R = C_{A_1} \cdots C_{A_k} R. \]

**Theorem 1.**

(a) \( h(\pi_Y(R)) = C_{\mathcal{U} \setminus Y} h(R) \).

(b) \( h(\sigma_F(R)) = \sigma_F(h(R)) = h(R) \cap \sigma_F(1) \).

(c) \( h(R \cup Q) = h(R) \cup h(Q) \) \( (a(R) = a(Q)) \).

(d) \( h(R \setminus Q) = h(R) \setminus h(Q) = h(R) \cap h(Q) \) \( (a(R) = a(Q)) \).

(e) \( h(R \bowtie Q) = h(R) \cap h(Q) \).

**Proof.**

(a) \( t \in h(\pi_Y(R)) \iff t[Y] \in \pi_Y(R) \)

\[ \iff (\exists s \in R) s[Y] = t[Y] \]

\[ \iff (\exists \bar{s} \in h(R)) \bar{s}[Y] = t[Y] \]

\[ \iff t \in C_{\mathcal{U} \setminus Y} h(R). \]

Conditions (b), (c), (d) are obvious.

(e) Let \( a(R) = X \), \( a(Q) = Y \).

\[ t \in h(R \bowtie Q) \iff t[X \cup Y] \in R \bowtie Q \]

\[ \iff t[X] \in R \land t[Y] \in Q \]

\[ \iff t \in h(R) \land t \in h(Q) \]

\[ \iff t \in h(R) \cap h(Q). \]

Another important operation on relations is that of “renaming an attribute.” Suppose that \( A \in a(R), B \notin a(R), \) and \( D(A) = D(B) \). Let \( s^A_B(R) \) be the result of renaming attribute \( A \) of \( R \) with attribute \( B \), i.e.,

\[ s^A_B(R) = \{ s^A_B(t): t \in R \}, \]

where \( s^A_B(t) \) is the tuple of type \( (a(R) \setminus \{A\}) \cup \{B\} \) with

\[ (s^A_B(t))(C) = t(C) \quad \text{if} \quad C \in a(R) \setminus \{A\}, \]

\[ = t(A) \quad \text{if} \quad C = B. \]

Only if our set \( \mathcal{U} \) of attributes is infinite and if we add the operation of renaming to the repertoire of the relational operations described above, our relational algebra becomes relationally complete in the sense of Codd. (More exactly, we may assume that there exists a finite set of “sorts,” corresponding to the attributes in the database schema, and that for each sort there is an infinite sequence \( A_1, A_2, \ldots \), of attributes with \( D(A_i) = D(A_j) \) for all \( i, j \)). Indeed, only under such an extension, it is possible to construct a relational expression corresponding to a formula of the predicate calculus involving an arbitrary number of variables. Notice that by using the operation of
renaming, it is possible to express the Cartesian product of arbitrary relations \( R, S \), by first renaming the columns of \( S \) so that, for the resulting relation \( S' \), we have \( \alpha(R) \cap \alpha(S') = \emptyset \), and then by performing the natural join \( R \bowtie S' \).

Theorem 1 remains true for an infinite set \( \mathcal{X} \). Assume, for simplicity, that all attribute domains are equal, and denote the common attribute domain by \( D \). We consider a locally finite cylindric set algebra of subsets of \( \mathcal{X} = D^\mathcal{X} \), and we define \( h(R) \) as before, by (3.1). We can restrict ourselves to a locally finite cylindric set algebra, since for every relation \( R \), \( \alpha(R) \) is finite, and the dimension set \( \Delta h(R) \) is contained in \( \alpha(R) \). The expression \( C_{\mathcal{X} \setminus Y} h(R) \) in Theorem 1(a) should now be understood as \( C_{\Delta h(R) \setminus Y} h(R) \).

It is easily seen that for the operation \( s^A_R \) we have
\[
h(s^A_R) = C_A(h(R) \cap D_{A,B}). \tag{3.2}
\]

Also, for the equi-join operation [Cod 2], defined as
\[
R[A = B]S = \{ t : t[\mathcal{X}] \in R \land t[Y] \in S \land t(A) = t(B) \},
\]
where \( \alpha(R) = X, \alpha(S) = Y, X \cap Y = \emptyset, A \in X, B \in Y \), we have
\[
h(R[A = B]S) = h(R) \cap h(S) \cap D_{A,B}.
\]

Another property of our mapping, which partly justifies the name "embedding," is given in

**Theorem 2.** Let \( R, S \) be two finite relations, and let either \( \alpha(R) = \alpha(S) \), or there is at least one attribute in the symmetric difference

\[
\alpha(R) \oplus \alpha(S) = (\alpha(R) \setminus \alpha(S)) \cup (\alpha(S) \setminus \alpha(R))
\]

with an infinite attribute domain. Then
\[
R \not\equiv S \Rightarrow h(R) \not\equiv h(S).
\]

**Proof.** Let us first notice that for any \( Y \subseteq \alpha(R), \pi_Y(h(R)) = \pi_Y(R) \). In particular, if \( \alpha(R) = \alpha(S) \) and \( h(R) = h(S) \) then
\[
R = \pi_{\alpha(R)}(R) = \pi_{\alpha(S)}(h(R)) = \pi_{\alpha(S)}(h(S)) = \pi_{\alpha(S)}(S) = S.
\]

Suppose now that \( \alpha(R) \not\equiv \alpha(S) \), say \( A \in \alpha(R) \setminus \alpha(S) \), with \( D(A) \) infinite. Then
\[
\pi_A(h(R)) = \pi_A(R)
\]
is finite, while
\[
\pi_A(h(S)) = D(A)
\]
is infinite. Consequently, \( h(R) \not\equiv h(S) \). \( \blacksquare \)
Several comments concerning Theorem 1 should be made. First notice that if we informally identify $R$ with $h(R)$—which is a common practice with embeddings in mathematics—then the classical relational algebra becomes a subset of a larger algebra where all operations are total, i.e., there are no restrictions on types of argument relations since all relations are of the same type, $\mathcal{H}$. In this larger algebra "the most complicated" of the relational operators, the join, becomes the ordinary set-theoretical intersection.\(^1\) On the other hand, projection, "the simplest" of the relational operators, becomes the main source of troubles—without this operation (and without the operation of renaming the columns) our algebra is simply a Boolean algebra of subsets of $\mathbf{1}$. Notice the change of intuition connected with projection: In the classical relational algebra it produces a relation which is "smaller" than the argument (some columns are deleted), while in our algebra it produces a "bigger" relation (some rows are added).

The operation of selection may be based on an arbitrary function $E: \mathbf{1} \rightarrow \{\text{true, false}\}$, not necessarily generated by atomic conditions of the form $A = a$ and $A = B$. Similarly as before, we define

$$\sigma_E(R) = \{t \in R: E(t) = \text{true}\}$$

for any $R \subseteq \mathbf{1}$.

Notice that $h(R)$ is in general infinite (since the attribute domains may be infinite), even if $R$ is finite. In many cases we can, however finitely, represent infinite relations in the following way: Let $t$ be a tuple of type $X$, and let $\bar{t}$ be the tuple of type $\mathcal{H}$ such that $\bar{t}[X] = t[X]$ and $\bar{t}(A) = *$ for every $A \in \mathcal{H} \setminus X$; here $*$ is a special symbol not in the domain of any attribute. Tuple $\bar{t}$ will represent the set of all tuples $q \in \mathbf{1}$ such that $q[X] = t$. A set $Q$ of tuples of type $\mathcal{H}$, with $*$ allowed to occur in these tuples, will represent the union $\bigcup_{q \in Q} \rho(q)$, where $\rho(q)$ is the set of tuples represented by $q$ (this union need not be disjoint). We write $Q$ as a table (called a $*$-table), in the usual way, and we omit a column in this table if all tuples $q \in Q$ contain $*$ in this column. Notice that under these rules $R$ (treated as a $*$-table) represents $h(R)$.

A relational expression will be called restricted if all relations occurring in it are assumed to be of specified types and all restrictions concerning types of arguments of the relational operations occurring in the expression are satisfied (i.e., the target attribute set of every projection is contained in the type of the argument, the selection condition does not depend on the attributes outside the type of the argument, unions and differences are applied to relations of equal types only). If these conditions are not assumed, then we speak about unrestricted relational expressions.

Let $f(R_1, \ldots, R_k)$ be an unrestricted relational expression which does not contain operations other than projection, union, renaming, join, and selection with the selection condition of the form

$$(A_1 = a_1) \land \cdots \land (A_p = a_p), \quad p \geq 1, A_i \neq A_j \text{ for } i \neq j$$

\(^1\)It has recently been pointed out to the second author by L. Zadeh that a similar interpretation of the join operation was mentioned in [Zad].
(a_i \in D(A_i), 1 \leq i \leq \rho). It can be shown that if \( R_1, \ldots, R_k \) are representable by finite \(*\)-
tables, then \( f(R_1, \ldots, R_k) \) is also representable by a finite \(*\)-table. The simple proof of
this fact is left to the reader, we only give an example.

**Example.** Let \( \mathcal{H} = \{A, B, C, D, E, F\} \), let

\[
\begin{array}{ccc}
A & B \\
R = & a & b \\
a_1 & b_1 \\
\end{array}
\quad \begin{array}{ccc}
B & C & E \\
Q = & b & c & e \\
b_1 & c_1 & e_1 \\
\end{array}
\quad \begin{array}{ccc}
A & D \\
T = & a & d \\
& * & d_1 \\
\end{array}
\]

\((a \neq a_1, b \neq b_1, c \neq c_1, d \neq d_1)\), and consider the relational expression

\[
f(R, Q, T) = \sigma_{F = f_1(\sigma_{(C = c) \land (B = b)}(R \cup Q) \Join T)}.
\]

The process of evaluating this expression is

\[
\begin{array}{ccc}
A & B & C & E \\
R_1 = R \cup Q = & a & b & * & * \\
a_1 & b_1 & * & * \\
* & b & c & e \\
* & b_1 & c_1 & e_1 \\
\end{array}
\quad \begin{array}{ccc}
A & B & C & E \\
R_2 = \sigma_{(C = c) \land (B = b)}(R_1) = & a & b & c & * \\
& b & c & e \\
\end{array}
\quad \begin{array}{ccc}
A & B & C & D & E \\
R_3 = R_2 \Join T = & a & b & c & d & * \\
a & b & c & d_1 & * \\
* & b & c & d_1 & e \\
\end{array}
\]

In evaluating the join of two \(*\)-tables we proceed as in the case of the usual join of
two relations, using the informal rule that * matches any other symbol s, giving s in
the resulting tuple.
Finally,

\[
\begin{array}{cccccc}
A & B & C & D & E & F \\
\hline
a & b & c & d & \ast & f_1 \\
\ast & b & c & d_1 & e & f_1 \\
\end{array}
\]

\[R_4 = \sigma_{f \ne f_1}(R_3) = \]

It may be noted that our operations over \(*\)-tables can be extended to the case where any positive (i.e., not containing \("\)\) Boolean combination of conditions of the form \((A = a)\) is allowed as a selection condition. Indeed, any such selection condition can be transformed into an equivalent disjunction of conditions of the form (3.3), and we may then use the rule

\[
\sigma_{E_1 \lor \ldots \lor E_k}(T) = \bigcup_{i=1}^{k} \sigma_{E_i}(T).
\]

It may be noted that the symbols \(*\) play a role similar to that of the null values \(\checkmark\) of Biskup [Bis]. Biskup interprets null values of this kind as \"attribute applicable but its value irrelevant for the intended processing,\" as opposed to his null values \(\exists\), meaning \"attribute applicable but its value at present unknown.\" He gives a definition of a union of two incompatible relations which is similar to ours (in fact, his definition is more general, since the symbol \(\exists\) is allowed), and he correctly notices that the outer union of Codd [Cod 3] (where the tuples are extended with existential null values \(\exists\) instead of \(\checkmark\) is less natural.

Of course, practical queries usually correspond to restricted relational expressions. However, considering unrestricted expressions may be convenient in equivalent transformation and optimization of relational queries, since we may then use identities of the type

\[
\sigma_E(R) \bowtie S = R \bowtie \sigma_E(S), \quad \pi_Y(R) \cup \pi_Y(S) = \pi_Y(R \cup S),
\]

etc. Our embedding of the relational algebra into a cylindric set algebra gives a precise and natural meaning to unrestricted relational expressions.

Obviously, the fact that in our approach we identify any relation \(R\) of type \(X\) with a relation \(R'\) of type \(\forall\) such that \(R = \pi_X(R') (R' = h(R))\) has nothing to do with the universal instance assumption (see, e.g., [Ull]). The universal instance assumption simply corresponds to the situation where all relational expressions considered contain only one relation symbol \(I\) of type \(\forall\) (\(\pi_Y(I)\) is then usually denoted by \(Y\)).

We conclude this section by noting that, in view of Theorem 1, the relation between the predicate calculus and cylindric set algebras discussed in the preceding section provides a simple \"explanation\" of Codd's Completeness theorem. More exactly, it is clear that an \(n\)-sorted predicate calculus (without equality) with only one
variable for each sort (which can be considered a version of the relational calculus) is equal in expressive power to the \( \text{Dfs}_n \) generated by the relations corresponding to the predicate symbols. Similarly, the usual relational calculus based on the predicate calculus (with equality) with an unlimited number of variables available is equal in expressive power to a locally finite \( \text{Cs}_\omega \), which is in fact closely related to the version of the relational calculus considered by Codd (this becomes clear if we try to consistently reformulate Codd's relational algebra into a form where every relation has columns corresponding to attributes (= variables), and no ordering of columns is ever considered in defining relational operations).

4. Axiomatizability and Decidability Problems

An important problem in query processing is that of query optimization (see, e.g., [ASU1, ASU2, CM, Got, GACLP, Hal, Min, Pal, Pec, SY, SC, Ull, WY, Yao]). This problem consists of finding, for a given relational expression \( f(R_1, \ldots, R_k) \), an equivalent relational expression \( g(R_1, \ldots, R_k) \) which is "better" according to specified criteria. Two (unrestricted) relational expressions \( f(R_1, \ldots, R_k) \) and \( g(R_1, \ldots, R_k) \) are called equivalent if the value of \( f(R_1, \ldots, R_k) \) is equal to the value of \( g(R_1, \ldots, R_k) \), for all relations \( R_1, \ldots, R_k \). Sometimes a slightly different notion of a typed equivalence is also considered, where \( R_1, \ldots, R_k \) are restricted to range over relations of specified types, say \( X_1, \ldots, X_k \), respectively. Obviously, the typed equivalence between \( f(R_1, \ldots, R_k) \) and \( g(R_1, \ldots, R_k) \) holds iff for all relations \( R_1, \ldots, R_k \) (with no restrictions on types), the equalities \( R_i = \pi_{X_i}(R) \), \( 1 \leq i \leq k \), logically imply the equality \( f(R_1, \ldots, R_k) = g(R_1, \ldots, R_k) \).

A natural technique of equivalent transformation of relational expressions is that based on using a set of axioms. Any axiom should be universally valid, i.e., it should hold true in any \( \text{Cs}_n \) for a fixed specified \( n \) (or for any locally finite \( \text{Cs}_\omega \), depending on the version of the relational algebra we consider; in what follows we always deal with a relational algebra based on unrestricted expressions and we identify it with a suitable cylindric set algebra via Theorem 1).

An example of such an axiom set is that consisting of \( \text{(CO)-(C7)} \) (see Section 2). A natural question which arises is whether these axioms fully characterize cylindric set algebras of dimension \( n \). Denoting by \( \text{ICs}_n \) the class of all those \( \text{CA}_n \)'s which are isomorphic to \( \text{Cs}_n \)'s, the problem is whether every \( \text{CA}_n \) which satisfies \( \text{(CO)-(C7)} \) is in \( \text{ICs}_n \). A rather trivial reason why the answer to this problem is negative is that \( \text{ICs}_n \) cannot be characterized by any set of equalities. Indeed, any equationally definable class is closed under products (in fact, by the celebrated Birkhoff Theorem, see, e.g. [Coh], a class of algebras is equationally definable iff it is closed under arbitrary products, subalgebras and homomorphic images), and \( \text{ICs}_n \) is not closed under products, as we show below.

Notice that in any \( \text{Cs}_n \), and hence in any algebra in \( \text{ICs}_n \),

\[
x \neq 0 \Rightarrow c_1 \cdots c_n x = 1.
\] (4.1)
On the other hand, this formula is not satisfied in a product of two $C_{s_n}$'s, e.g., in $A \times A$, where $A$ is a $C_{s_n}$. Indeed, consider the element

$$x = \langle 0, 1 \rangle \in A \times A.$$ 

Clearly, $x \neq 0 = \langle 0, 0 \rangle$, but

$$c_1 \cdots c_n x = \langle 0, 1 \rangle \neq 1 = \langle 1, 1 \rangle.$$

Hence $A \times A \notin IC_{s_n}$.

By the way, notice that (4.1), translated into the usual relational algebra, expresses a rather esoteric fact that

$$\pi_\phi(R) = \{ \varepsilon \} \quad \text{if} \quad R \neq \phi,$$

$$= \emptyset \quad \text{if} \quad R = \phi,$$

where $\{ \varepsilon \}$ is a nonempty relation of type $\phi$, consisting of a single tuple $\varepsilon$ of type $\phi$.

A $CA_n$ is called representable if it is isomorphic to the product of a family of $C_{s_n}$'s. The class of all representable $CA_n$'s is denoted by $RCA_n$. Although $IC_{s_n}$ is not equationally definable, Tarski [Tar 2] has shown that the class $RCA_n$ is equationally definable for every $n$ (and also for every infinite dimension).

Representable cylindric algebras are related to so-called generalized cylindric set algebras [IIIMTAN]. Let $X_i$, $i \in I$ be nonempty sets, let $X_i \cap X_j = \emptyset$ for any two distinct $i, j \in I$, and denote $V = \bigcup_{i \in I} X_i$. By an $n$-dimensional generalized cylindric set algebra, or $G_{s_n}$, we mean any Boolean algebra of subsets of $V$ closed under the operations $C_i$ defined by

$$C_i R = \left\{ y \in V : y(i/\alpha) \in R \text{ for some } \alpha \in \bigcup_{i \in I} X_i \right\},$$

and containing all diagonals

$$D_{ij} = \{ y \in V : y_i = y_j \}.$$

Clearly, any $G_{s_n}$ is isomorphic to a product of $C_{s_n}$'s, and conversely, any product of $C_{s_n}$'s is isomorphic to a $G_{s_n}$. It is also easy to see that a $G_{s_n}$ is a $C_{s_n}$ iff it satisfies (4.1) for every $x$.

Another important result of Tarski related to representable cylindric algebras is that any locally finite cylindric algebra of infinite dimension is representable. On the other hand, the class of all locally finite cylindric algebras of any fixed infinite dimension is not equationally definable (since it is not closed under arbitrary products).

Assume that a set $\mathcal{E}$ of equalities characterizes $RCA_n$. It is then easy to see that an algebraic identity (in the form of an equality of two expressions in the language of $CA_n$) holds true in every $C_{s_n}$ iff it holds true in every representable $CA_n$, i.e., iff our identity is a logical consequence of $\mathcal{E}$. In other words, $\mathcal{E}$ is a complete axiom set for
equivalent transformations of expressions in $C_{s_n}$'s. By the usual completeness theorem, an identity holds true in every $C_{s_n}$ iff it can be proved by using axioms in $\mathscr{A}$.

If $n = 1$ then it may be shown that every $CA_n$ is representable, and consequently $RCA_n$ is exactly characterized by (CO)–(C7). In the case of $n = 2$, a complete axiom set can be obtained by adjoining the following two simple equations to (CO)–(C7):

$$c_i(x \cdot y \cdot c_j(x \cdot -y)) \cdot -c_j(c_i x \cdot -d_i) = 0,$$

1 $\leq i, j \leq 2$, $i \neq j$ (Henkin [Hen 1], see also [HT]). Monk [Mon 1] proved that for $n \geq 3$ no finite set of equations (more generally, no finite set of first-order axioms) characterizes $RCA_n$. He also proved, for every infinite dimension $\alpha$, that no finite schema (of the type of that consisting of (CO)–(C7) characterizes $RCA_\alpha$. An explicit (but rather complicated) infinite set of equations characterizing $RCA_n$ is also given in [Mon 1], for every $\alpha \geq 3$.

Similar results have been obtained for diagonal-free cylindric algebras. Let the definition of the class $RDF_n$ of representable $Df_n$'s be analogous to the definition of $RCA_n$. It can be shown that the class $RDF_n$ is equationally definable for every $n$. It turns out that not only every $Df_1$ but also every $Df_2$ is representable. Johnson [Joh] proved that for every (finite) $n \geq 3$ the class $RDF_n$ cannot be characterized by any finite set of first-order axioms. By the results of Section 3, we obtain the important

**Corollary 3.** If $\vert \mathscr{A} \vert \geq 3$ then there is no finite complete set of identities for equivalent transformations of unrestricted relational expressions built up from relations of types contained in $\mathscr{A}$ by using projection, join, union, and complementation.

Corollary 3 is closely related to the fact that certain proofs in the usual predicate calculus inherently require many “auxiliary variables,” not occurring in the formula to be proved. For related results see [Hen 2, Mon 3].

Many other interesting properties of cylindric set algebras can be found in [HMTAN]. For example, there exists an equation which holds identically in every $C_{s_n}$ ($n \geq 3$) with a finite base, but fails in some finite $C_{s_n}$ with an infinite base.

Another important problem is the following. Given an equation, determine whether or not it is satisfied identically in every $CA_n$ ($C_{s_n}$). From results of Tarski and others (see [Mad, Pie, Mon 2]) it follows that this problem (both for $CA_n$'s and for $C_{s_n}$'s) is decidable if $n \leq 2$ and undecidable if $n \geq 3$. Hence we obtain

**Corollary 4.** If $\vert \mathscr{A} \vert \geq 3$ then there is no algorithm for determining whether or not two unrestricted relational expressions built up from relation symbols (ranging over relations of types contained in $\mathscr{A}$) by means of projection, join, union, complementation, and restriction ($\sigma_{A=B}$) are equivalent.

By more closely analyzing the proof given in [Mad], one can easily show that the equivalence problem for restricted expressions with just three attributes, involving projection, join, union, and difference is undecidable.
Corollaries 3 and 4 concern the case where the full strength of the relational algebra, including the difference operation, is involved. It would be interesting to investigate weaker versions of the relational algebra. A simple case is that involving only projection and join (relational expressions involving only projection and join will be called PJ-expressions). An interesting case is also that involving only one relation symbol (this is connected with the universal instance assumption).

If $|\mathcal{U}| = 2$, say $\mathcal{U} = AB$, then it is easy to see that the only non-equivalent PJ-expressions involving one relation symbol, $R$, are

$$R, \pi_a(R), \pi_b(R), \pi_a(R) \bowtie \pi_b(R).$$

It is interesting to note that if the difference operation is allowed then the number of non-equivalent expressions becomes infinite. Indeed, letting

$$f_0(R) = R, \quad f_{n+1}(R) = \pi_a(f_n(R) - R) \bowtie R,$$

we obtain an infinite sequence $f_0(R), f_2(R), ...$, of non-equivalent expressions (see [HMT, Theorem 2.1.1]). It is also easy to see that the number of non-equivalent PJ-expressions involving two relation symbols is infinite.

We now consider the case $|\mathcal{U}| \geq 3$.

**Theorem 5.** If $|\mathcal{U}| \geq 3$ then there exist infinitely many nonequivalent PJ-expressions containing only one relation symbol.

**Proof.** We may restrict ourselves to the case $\mathcal{U} = ABC$. For every expression $e$, let

$$g_b(e) = \pi_{ab}(e) \bowtie \pi_{bc}(e),$$

$$g_c(e) = \pi_{bc}(e) \bowtie \pi_{ac}(e),$$

$$f(e) = g_b(g_c(e)).$$

Let $f^0(e) = e, f^{n+1}(e) = f(f^n(e)), n \geq 0$. We shall prove that the expressions

$$R, f(R), f^2(R), ..., \quad (4.2)$$

are pairwise non-equivalent. Let $\mathcal{Z}$ be the set of integers, and let for every $i \geq 0$

$$R_i = \{ (a, b, c) \in \mathcal{Z}^3: |a - c| \leq 2i \land |b - c| \leq 1 \}.$$

Clearly $R_i \nsubseteq R_j$ for $i < j$. Let $R = R_0$. We claim that for every $i \geq 0, f^i(R) = R_i$. We use induction on $i$. Our claim is obviously true for $i = 0$. Suppose now that $i \geq 0$ and that $f^i(R) = R_i$. We have $f^{i+1}(R) = f(f^i(R)) = f(R_i)$.
\[ \langle a, b, c \rangle \in f(R_i) \Leftrightarrow \langle a, b, c \rangle \in g_C(g_C(R_i)) \]
\[ \Leftrightarrow (\exists a_1, c_1) \langle a_1, b, c \rangle, \quad \langle a, b, c \rangle \in g_C(R_i) \]
\[ \Leftrightarrow (\exists a_1, c_1, a_2, b_2, a_3, b_3) \quad \langle a_2, b, c \rangle, \]
\[ \langle a_1, b_2, c \rangle, \]
\[ \langle a_3, b, c \rangle, \]
\[ \langle a, b_3, c_1 \rangle \in R_i. \]

It follows that \( \langle a, b, c \rangle \in f(R_i) \) implies
\[ |b - c| \leq 1, \]
\[ |a - c| \leq |a - c_1| + |b - c_1| + |b - c| \leq 2i + 2 = 2(i + 1), \]
so that \( f(R_i) \subseteq R_{i+1} \). On the other hand, any \( \langle a, b, c \rangle \in R_{i+1} \) can easily be shown to be in \( f(R_i) \), by a suitable choice of \( a_1, c_1, a_2, b_2, a_3, b_3 \). Hence \( f(R_i) = R_{i+1} \). \( \blacksquare \)

It is clear that by replacing \( Z \) by an interval \( \{1, \ldots, m\} \), we can construct a finite relation \( R \) which makes arbitrarily many initial terms in (4.2) pairwise different, letting \( m \) be sufficiently large.

An interesting open problem is whether the equivalence of PJ-expressions is finitely axiomatizable (for related results see [YP]).

5. Duality

We conclude this paper by brief remarks which shed some light on the nature of the duality between the project-join and select-union operator pairs (this question is asked, e.g., by Sciore [Sci]).

It can easily be shown [HMT, Sect. 1.4] that the mapping \( x \mapsto -x \) defines an isomorphism from any CA,
\[ A = \langle A, +, \cdot, -, 0, 1, c_i, d_{ij} \rangle, \quad 1 \leq i, j \leq n \]
into a CA \( A^\theta \) (called the dual of \( A \)):
\[ A^\theta = \langle A, \cdot, +, -, 1, 0, c_i^\theta, -d_{ij} \rangle, \quad 1 \leq i, j \leq n \]
where
\[ c_i^\theta x = -c_i - x. \]

The operations \( c_i \) and \( c_i^\theta \) are sometimes called outer and inner cylindrifications, respectively, since in the case of cylindric set algebras \( c_i R \) and \( c_i^\theta R \) (usually written as \( C_i R, C_i^\theta R \)) are the smallest cylinder (with axis parallel to the \( i \) axis) containing \( R \),
and the largest cylinder contained in $R$. This is illustrated, in the case of $n = 2$, in Fig. 2.

Clearly, the relation between inner cylindrification and universal quantification is the same as that between outer cylindrification and existential quantification.

Let $X = \{A_1, \ldots, A_k\}$ be a set of attributes. We define, for any relation $R$ of type $\mathcal{H}$,

$$C_X^{\delta} R = C_{A_1}^{\delta} \cdots C_{A_k}^{\delta} R$$

and

$$\pi_X^{\delta} R = C_{\mathcal{H} \setminus X}^{\delta} R$$

(cf. Theorem 1). The “dual projection” operation $\pi_X^{\delta}$ can be treated as a special case of Codd’s division operation (see, e.g., [Cod 3]). On the other hand, if $R$ is represented by a finite table, and if all attribute domains corresponding to the attributes occurring in the table are infinite, then $\pi_X^{\delta}$ can be treated as a kind of “selection” which selects those rows of the table which have *’s in all columns corresponding to attributes in $\mathcal{H} \setminus X$ (recall that if an attribute $A$ does not appear in a table, then we assume $t(A) = *$ for any tuple $t$ in this table). For example, if

\[
\begin{array}{cccc}
A & B & C & D \\
\hline
\ast & \ast & \ast & \ast \\
\ast & \ast & c & \ast \\
\ast & b_1 & c_1 & \ast \\
a_1 & b_2 & c_2 & \ast \\
\end{array}
\]

\[
T = 
\begin{array}{cccc}
A & B & C & D \\
\hline
\ast & \ast & \ast & \ast \\
\ast & \ast & c & \ast \\
\ast & b_1 & c_1 & \ast \\
a_1 & b_2 & c_2 & \ast \\
\end{array}
\]
Let $f$ be the PJ-expression

$$f(R) = \bigwedge_{i=1}^{k} \pi_{X_i}(R).$$

Clearly $R \subseteq f(R) = f(f(R))$. If $R = f(R)$, i.e., if $R$ is the join of its projections on $X_1, ..., X_k$, then we say that $R$ satisfies the join dependency [Ris]

$$\mathcal{H} = \bigwedge_{i=1}^{k} X_i$$

(5.1)

(the case of $k = 2$ corresponds to multivalued dependencies, see [Zan, Fag1]). It is easily seen that a relation satisfies the join dependency (5.1) if it can be represented as a join of any $k$ relations of types $X_1, ..., X_k$, respectively. Consider now a dual expression

$$f^\delta(R) = \bigvee_{i=1}^{k} \pi_{X_i}^\delta(R).$$

Clearly $R \supseteq f^\delta(R) = f^\delta(f^\delta(R))$. If $R = f^\delta(R)$, then we say that $R$ satisfies the dual join dependency

$$\mathcal{H} = \bigvee_{i=1}^{k} X_i.$$ 

Roughly speaking, $R$ satisfies the above dual join dependency if $R$ can be decomposed into the union of $k$ (in general, mutually incompatible) relations of types $X_1, ..., X_k$, respectively.

Let $X, Y, Z$ be nonempty mutually disjoint sets such that $XYZ = \mathcal{H}$ (as usual in database theory, concatenation of attribute sets stands for their union). In Fig. 3 we show a simple "geometric" interpretation of the join dependency $\mathcal{H} = XY \ast XZ$ (or, in the usual notation, the multivalued dependency $X \rightarrow Y$) and of the dual join dependency $\mathcal{H} = XY + XZ$. In the former case, for any combination $x_0$ of the values of attributes in $X$, the section $X = x_0$ of our relation represented in a three-dimensional $X, Y, Z$-space is rectangular (i.e., it has the form of a Cartesian product of a set of $Y$-values and a set of $Z$-values). In the latter case, the section $X = x_0$ is co-rectangular, i.e., it has the form of a complement (with respect to the $Y, Z$ plane) of a rectangle.
FIG. 3. A geometric interpretation of (a) the join dependency $\mathcal{H} = XY \ast XZ$, and (b) of the dual join dependency $\mathcal{H} = XY + XZ$.

Notice that such a co-rectangle is the union of a relation of type $Y$ and a relation of type $Z$.

It is interesting to note that the case of nonapplicable attributes can be modeled by a relation satisfying a dual join dependency. Let us consider the example from [Fag2], the vehicle schema with attributes VEHICLE-NAME, WINGSPAN and SAIL-AREA. Some of these attributes are not applicable to some vehicles, e.g., WINGSPAN applies to air vehicles but does not apply to water vehicles. If we use the symbol $\ast$ to denote the fact of nonapplicability of an attribute to a tuple, then the resulting $\ast$-table satisfies the dual join dependency $\mathcal{H} = X + Y$, where

$$\mathcal{H} = \{\text{VEHICLE-NAME, WINGSPAN, SAIL-AREA}\},$$
$$X = \{\text{VEHICLE-NAME, WINGSPAN}\},$$
$$Y = \{\text{VEHICLE-NAME, SAIL-AREA}\}.$$

In a similar way, with any multi-relational database scheme $X_1, \ldots, X_k$, where $X_i \not\subseteq X_j$ for $i \neq j$, we can associate a one-relation scheme over $X = X_1 \cup \cdots \cup X_k$ endowed with the dual join dependency $X = \bigoplus_{i=1}^k X_i$. It should be emphasized that such a construction has nothing to do with the universal instance assumption—we do not assume the instances of $X_1, \ldots, X_k$ to satisfy any conditions. If all attribute domains are infinite then it is easy to see that the mapping

$$f(R_1, \ldots, R_k) = \bigcup_{i=1}^k R_i,$$

where $R_i$ is a finite relation instance corresponding to relation scheme $X_i$, $i = 1, \ldots, k$, is always lossless in the sense that by applying $\pi_i^{2_k}$ we can get any of the relations $R_i$ back from $\bigcup_{i=1}^k R_i$ (unlike in the case of the dual mapping, $\pi^k \bigcup_{i=1}^k R_i$, which is

2 A database scheme is a collection of relation schemes, and each relation scheme is a subset of $\mathcal{H}$. An instance of relation scheme $X$ is any relation of type $X$ (see [UL1]).
lossless only if $R_1, \ldots, R_k$ satisfy the universal instance assumption). In this way we can talk about a multi-relational database scheme in terms of one relational scheme without preassuming any conditions to be satisfied by the instances of our multi-relational scheme.

REFERENCES


A. Pieczkowski, Undecidability of the homogeneous formulas of degree 3 of the predicate calculus, *Studia Logica* 22 (1968), 7–16.


