LITERATURE CITED


DATABASE MODELS AND CLOSURE OPERATORS*

J. Demetrovic, L. O. Libkin,
and I. B. Muchnik

UDC 519.6

We consider functional dependency structures in relational databases. New results are obtained for closure operators of changing databases.

1. INTRODUCTION

The relational database model was introduced by Codd [1]. Given is a set of attributes \( \Omega = \{a_1, \ldots, a_n\} \) (e.g., name, age, etc.) and for each attribute \( a_i \in \Omega \) is defined the set of its values \( D_i \). A relation on \( \Omega \) is a subset \( R \) of the Cartesian product \( D_1 \times \ldots \times D_n \).

Let a relation \( R \) be given on \( \Omega \). It can be represented as a matrix (a table) whose rows correspond to the elements of \( R \) and columns to the attributes. In other words, a row in the table is a record in the database characterized by the attribute values.

The table representation led Armstrong [2] to introduce the concept of functional dependency. We say that \( B \subseteq \Omega \) is functionally dependent (or, in short, dependent) on \( A \subseteq \Omega \), which is denoted by \( A \rightarrow B \), if the values of all the attributes in \( B \) can be determined given the values of the attributes in \( A \) for each element of \( R \). More rigorously, \( A \rightarrow B \) if and only if the matrix representing \( R \) does not contain two rows in which the values of the attributes from \( A \) are equal while the values of the attributes from \( B \) are different.

Armstrong [2] also introduced the following axioms for functional dependencies:

\[
A \rightarrow A; \quad (F1)
\]

*This study was supported by the Hungarian Foundation for Scientific Research (Grant 1066).

A family of functional dependencies that satisfies (F1)-(F4) is called complete [3]. The family of all dependencies that can be constructed from the relation R on Ω is complete and, conversely, for each complete family of functional dependencies there exists a representing relation on Ω.

Assume that the relation R on Ω and its representing complete family are given. For each A ⊆ Ω define \( L_R(A) = \{ a ∈ Ω | A → a \} \). Armstrong showed that \( L_R \) has the following properties:

\[
\begin{align*}
A ⊆ L_R (A); \\
A ⊆ B ⇒ L_R (A) ⊆ L_R (B); \\
L_R (L_R (A)) = L_R (A).
\end{align*}
\]

In other words, \( L_R \) is a closure operator (or simply closure) on Ω [4]. Moreover, for any closure \( L \) on Ω there exists a relation \( R \) on Ω such that \( L = L_R \) [2]. Thus, closure operators on Ω may be viewed as a model of a database (more precisely, of its functional dependencies).

The construction of a semilattice database model [3, 5] involves the following equivalent description of closures on Ω. Let \( L \) be a closure on Ω and

\[
Z (L) = \{ L (A) | A ⊆ Ω \},
\]

Then \( Z(L) \) satisfies the conditions

\[
Ω ∈ Z (L); \\
A, B ∈ Z (L) ⇒ A ∩ B ∈ Z (L),
\]

i.e., \( Z(L) \) is a (lower) semilattice of sets. If a semilattice \( Z \) satisfying (S1) is given and

\[
∀ A ⊆ Ω : L (A) = ∩ (B | A ⊆ B, B ∈ Z),
\]

then \( Z = Z(L) \) (see [3]).

Thus, semilattices also may be viewed as a database model.

Remark 1. In database literature [3, 5, 6] it is often assumed that

\[
L_R (Ø) = Ø.
\]  

This imposes a constraint on \( R \): the matrix does not contain constant columns. We assume that \( R \) is arbitrary and that in general (C4) does not hold (although this restriction is sometimes imposed in what follows).

Remark 2. Strictly speaking, closures and semilattices are models of complete families of functional dependencies, and not of databases as such. In other words, there are many different relations generating the same closure. In particular, a necessary and sufficient condition for representability of closure by a relation was derived in [7] and minimal matrix representations were studied in [8].