Available online at www.sciencedirect.com
ScienceDirect
Discrete Applied Mathematics 156 (2008) 911 – 923
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

Non-deterministic ideal operators: An adequate tool for formalization in Data Bases☆

P. Cordero, A. Mora, I.P. de Guzmán, M. Enciso

Depto. Matemática Aplicada, E.T.S.I. Informática, Universidad de Málaga, 29071 Málaga, Spain

Received 25 July 2003; received in revised form 27 May 2005; accepted 28 February 2007
Available online 16 June 2007

Abstract

In this paper, we propose the application of formal methods to *Software Engineering*. The most used data model is the relational model and we present, within the general framework of lattice theory, this analysis of functional dependencies. For this reason, we characterize the concept of f-family by means of a new concept which we call non-deterministic ideal operator (*nd.ideal-o*). The study of nd.ideal-o.s allows us to obtain results about functional dependencies as trivial particularizations, to clarify the semantics of the functional dependencies and to progress in their efficient use, and to extend the concept of *schema*. Moreover, the algebraic characterization of the concept of *Key of a schema* allows us to propose new formal definitions in the lattice framework for classical normal forms in relation schemata. We give a formal definition of the *normal forms* for functional dependencies more frequently used in the bibliography: the second normal form (2FN), the third normal form(3FN) and Boyce–Codd's normal form (FNBC).

Keywords: Non-deterministic operator; Ideal; Lattice; Functional dependency; Database; f-Family; Normal form

1. Introduction

In the last few years, works on *Software Engineering* are being aimed at obtaining design and system development tools with greater reasoning and deduction power. Such an aim is covered, generally, with mathematical tools which allow the development of automatic techniques. In the field of storage devices, the most used data model is the relational model. This model can be seen as a set of elements (attributes) which maintain some relations called functional dependencies (FDs). The design process consists of the creation of a set of tables which group the attributes, keeping a series of good properties which ensure their correct functioning and a limited redundancy. For this task, the theory of normalization has a special prominence as it proposes some properties that databases must meet in order to avoid redundancies and inconsistencies in the stored data.

Recently, researchers have related FDs theory with emerging technology. Thus, the extension of FDs to XML has been treated in [6,7,18,25,31] and the extraction of FDs from large databases has been performed using data mining techniques in [4,19,22,26,27,35]. Among the applications of FDs we can cite [24] where dependencies theory is used

☆ Partially supported by Spanish DGI project TIN2007-65819 and Junta de Andalucia project TIC115.
E-mail addresses: pcordero@uma.es (P. Cordero), amora@ctima.uma.es (A. Mora), guzman@ctima.uma.es (I.P. de Guzmán), enciso@lcc.uma.es (M. Enciso).

0166-218X/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2007.02.014

to control a distributed system and [34] where a study is being carried out on how information dependencies can be applied to the design of OLAP cubes in datawarehousing environments.

Nevertheless, the wide bibliography on FDs in Data Bases appeared during the last two decades reveals that there is much to do yet on the plan of formalization. In [36] and in [5][1] a series of directions on where to continue work in the area of databases are established. Both proposals favor a coordinated development of practical results with those with a strong formal component. In [23] authors introduce a formalization of relational databases and FDs using category theory and remark that "*Database theory has had to be formalized at a single level in the past because appropriate mathematical formulations were not available as the technology was being developed*".

We believe, as Bernard Thalheim in [36] that *there is still a great deal of work to be done in this area*. Our paper focuses on these points. Concretely, we present, within the general framework of the lattice theory, the analysis of FDs. To that end, we characterize the concept of f-family (widely known in the bibliography on Data Bases) by means of a new concept which we call non-deterministic ideal operator.

The study of nd.ideal-o.s in the context of the lattice theory, as we will see, allows us:

- To obtain, as trivial particularizations, results about FDs which appear scattered over the relevant authors' papers in the area of Data Bases.
- To clarify the semantics of the FDs and to progress in their efficient use as for the management of Data Bases.
- To extend the concept of *scheme* and, consequently, to widen its scope of application.
- As final aim of this paper, the algebraic characterization of the concept of *Key of a scheme* allows us to propose new definitions for normal forms in relation schemata.

2. Closure operators and non-deterministic operators

The importance of the concept of closure operator in a certain amount of domains [8] is widely accepted: algebra, topology, geometry, logic [28], computer science [13], relational databases [14], etc. This concept that we deal with now will be a basic tool in the development of this paper.

Definition 2.1. Let $(A, \leqslant)$ be a poset (partially ordered set). We say that an application $c : A \to A$ is a *closure operator* if c is extensive, idempotent and monotone. That is, if c satisfies the following conditions:

- for all $a \in A$, we have that $a \leqslant c(a)$ and $c(c(a)) \leqslant c(a)$;
- for all $a, b \in A$, if $a \leqslant b$ then $c(a) \leqslant c(b)$.[2]

If a is a fixed point of c (i.e. $c(a) = a$), then we say that a is *c-closed* and we denote by $\mathscr{S}_c(A)$ the set of c-closed elements in A. That is,

$$\mathscr{S}_c(A) = \{a \in A | c(a) = a\}.$$

As examples of closure operators we have the lower and upper closure operators: if $(U, \leqslant)$ is a poset, $\uparrow, \downarrow: 2^U \to 2^U$ are given by

$$X\uparrow = \bigcup_{x \in X} [x) = \bigcup_{x \in X} \{y \in U | x \leqslant y\}, \quad X\downarrow = \bigcup_{x \in X} (x] = \bigcup_{x \in X} \{y \in U | y \leqslant x\}.$$

Hereinafter, we will say *lower closed* instead of $\downarrow$-closed and *upper closed* instead of $\uparrow$-closed.

In the rest of the paper we will use the well-known concepts of $\vee$-semilattice and lattice [21]. Likewise, we will use the concept of ideal in a poset as a subset that is directed[3] and lower closed. Particularly, an ideal in an $\vee$-semilattice is a lower closed sub-$\vee$-semilattice.

Now, we introduce the notion of non-deterministic operator.

[1] This group and its 56 open problems have been proposed by the most famed researchers in the area J. Biskup, J. van der Bussche, P. De Bra, J. Demetrovics, G. Gottlob, S. Hegner, A. Heuer, G. Katona, H. Nam Son, J. Paredaens, L. Tenenbaum, B. Thalheim.

[2] Therefore, $c(c(a)) = c(a)$.

[3] $B \subseteq A$ is directed in $(A, \leqslant)$ if any non-empty finite subset of B has an upper bound in B.