Available online at www.sciencedirect.com
SCIENCE DIRECT®
Electronic Notes in Theoretical Computer Science
ELSEVIER Electronic Notes in Theoretical Computer Science 123 (2005) 125–137
www.elsevier.com/locate/entcs

The Implication Problem of Functional Dependencies in Complex-value Databases

Sven Hartmann[1] and Sebastian Link[2]

Information Science Research Centre, Massey University, Palmerston North, New Zealand

Abstract

Modern applications increasingly require the storage of data beyond relational structure. The challenge of providing well-founded data models that can handle complex objects such as lists, sets, multisets, unions and references has not been met yet. The success of such data models will greatly depend on the existence of automated database design techniques that generalise achievements from relational database design. In this paper, a provably-correct and polynomial-time algorithm for deciding implication of functional dependencies in the presence of all combinations of records, lists, sets, and multisets is proposed. The notion of a functional dependency is based on a Brouwerian algebra of subattributes, yielding a complementary expressiveness.

Keywords: Logic in Databases, Implication Problem, Functional Dependency, Brouwerian Algebra, Data Type

1 Introduction

Functional dependencies (FDs) were introduced in the context of the relational data model (RDM) by Codd in 1972 (see [18]). FDs are expressions of the form $X \to Y$ on a relation schema R with $X, Y \subseteq R$. A relation r over R is said to satisfy the FD $X \to Y$ if and only if any two tuples of r that coincide on X also coincide on Y. FDs are not independent from one another. That is, an FD $X \to Y$ is *implied* by a set Σ of FDs, if $X \to Y$ is satisfied by every relation which already satisfies all dependencies in Σ. A sound and

[1] Email:s.hartmann@massey.ac.nz
[2] Email:s.link@massey.ac.nz

doi:10.1016/j.entcs.2004.04.051

complete set of inference rules for the implication of FDs in the RDM has been discovered by Armstrong in [3]. In the context of the RDM such inference rules are easily available. The set of all attribute sets for some relation schema forms a Boolean algebra with respect to set inclusion, union, intersection and complement. On the basis of Armstrong's axiomatisation, polynomial time algorithms for deciding the implication problem [7], deciding the equivalence of two given sets of FDs [10] and deriving minimal covers for FDs [37] have been developed. A solution to these problems was a big step towards automated database schema design [10] which some researchers see as the ultimate goal in dependency theory [8]. Moreover, normal form proposals such as Boyce-Codd Normal Form and Third Normal Form [8,7,11] have been semantically justified [20,50] by formally proving the equivalence to the absence of redundancies and abnormal update behavior using again Armstrong's axiomatisation.

During the last couple of decades, many new and different data models have been introduced. First, so called semantic data models have been developed [17,33,47], which were originally just meant to be used as design aids, as application semantics was assumed to be easier captured by these models. Later on some of these models, especially the nested relational model [35], object-oriented and object-relational models [6,24,43,44] have become interesting as data models in their own right and some dependency and normalisation theory has been carried over to these advanced data models [25,26,27,39,41,46,52]. Most recently, the major research interest is on the model of semi-structured data and XML [48]. Work on integrity constraints in the context of XML can be found in [2,15,23,22,51]. One key problem is to develop dependency theories (or preferably a unified theory) for these advanced data models. Biskup [13] lists in particular two challenges for database design theory: finding a unifying framework and extending achievements to deal with advanced database features such as complex object types. We propose to classify data models according to the type constructors which are supported by the model. The RDM, for instance, is completely captured by the record type, the nested relational data model by the record and set type. This view allows to study problems in dependency theory for various classes of dependencies in the presence of various combinations of types, as illustrated in Figure 1.

In the present paper we consider all combinations of record, set, multiset and list type that include at least the record type. The need for these various types arises from applications that store ordered relations, time-series data, meteorological and astronomical data streams, runs of experimental data, multidimensional arrays, textual information, voices, sound, images, video, etc. They have been subject to studies in the deductive and temporal database community for some time [42,40], and occur also naturally in object-oriented