Abstract. Traditionally, dependency theory has been developed for uninterpreted data. Specifically, the only assumption that is made about the data domains is that data values can be compared for equality. However, data is often interpreted and there can be advantages in considering it as such, for instance obtaining more compact representations as done in constraint databases. This paper considers dependency theory in the context of interpreted data. Specifically, it studies constraint-generating dependencies. These are a generalization of equality-generating dependencies where equality requirements are replaced by constraints on an interpreted domain. The main technical results in the paper are a general decision procedure for the implication and consistency problems for constraint-generating dependencies, and complexity results for specific classes of such dependencies over given domains. The decision procedure proceeds by reducing the dependency problem to a decision problem for the constraint theory of interest, and is applicable as soon as the underlying constraint theory is decidable. The complexity results are, in some cases, directly lifted from the constraint theory; in other cases, optimal complexity bounds are obtained by taking into account the specific form of the constraint decision problem obtained by reducing the dependency implication problem.

1 Introduction

Relational database theory is largely built upon the assumption of uninterpreted data. While this has advantages, mostly generality, it foregoes the possibility of exploiting the structure of specific data domains. The introduction of constraint databases [21] was a break with this uninterpreted-data trend. Rather than defining the extension of relations by an explicit enumeration of tuples, a
constraint database uses constraint expressions to implicitly specify sets of tuples. Of course, for this to be possible in a meaningful way, one needs to consider interpreted data, that is, data from a specific domain on which a basic set of predicates and functions is defined. A typical example of constraint expressions and domain are linear inequalities interpreted on the reals. The potential gains from this approach are in the compactness of the representation (a single constraint expression can represent many, even an infinite number of, explicit tuples) and in the efficiency of query evaluation (computing with constraint expressions amounts to manipulating many tuples simultaneously).

Related developments have concurrently been taking place in temporal databases. Indeed, time values are intrinsically interpreted and this can be exploited for finitely representing potentially infinite temporal extensions. For instance, in [19] infinite temporal extensions are represented with the help of periodicity and inequality constraints, whereas in [10, 11] and [3] deductive rules over the integers are used for the same purpose. Constraints have also been used recently for representing incomplete temporal information [31, 23].

If one surveys the existing work on databases with interpreted data and implicit representations, one finds contributions on the expressiveness of the various representation formalisms [2, 5, 4], on the complexity of query evaluation [9, 12, 25, 31], and on data structures and algorithms to be used in the representation of constraint expressions and in query evaluation [28, 7, 8, 22]. However, much less has been done on extending other parts of traditional database theory, for instance schema design and dependency theory. It should be clear that dependency theory is of interest in this context. For instance, in [18], one finds a taxonomy of dependencies that are useful for temporal databases. Moreover, many integrity constraints over interpreted data can be represented as generalized dependencies. For instance, the integrity constraints over databases with ordered domains studied in [17, 33] can be represented as generalized dependencies. Also, some versions of the constraint checking problem studied in [16] can be viewed as generalized dependency implication problems.

One might think that the study of dependency theory has been close to exhaustive. While this is largely so for dependencies over uninterpreted data (that is, the context in which data values can only be compared for equality) [29], the situation is quite different for dependencies over data domains with a richer structure. The subject of this paper is the theory of these interpreted dependencies.

Specifically, we study the class of constraint-generating dependencies. These are the generalization of equality-generating dependencies [6], allowing arbitrary constraints on the data domain to appear wherever the latter only allow equalities. For instance, a constraint-generating dependency over an ordered domain can specify that if the value of an attribute $A$ in a tuple $t_1$ is less than the value of the same attribute in a tuple $t_2$, then an identical relation holds for the values of an attribute $B$. This type of dependency can express a wide variety of constraints on the data. For instance, most of the temporal dependencies appearing in the taxonomy of [18] are constraint-generating dependencies.