Chasing Relational Database Constraints Backwards

Stéphane Coulondre

LIRMM (CNRS/Université Montpellier II)
161 rue Ada, Montpellier Cedex 5, France
coulondre@lirmm.fr

Abstract. Data dependencies are well known in the context of relational database. They aim to specify constraints that the data must satisfy to model correctly the part of the world under consideration. The implication problem for dependencies is to decide whether a given dependency is logically implied by a given set of dependencies. A proof procedure for the implication problem, called “chase”, has already been studied in the generalized case of tuple-generating and equality-generating dependencies. The chase is a bottom-up procedure: from hypotheses to conclusion, and thus is not goal-directed. It also requires the dynamic creation of new symbols. This paper introduces a new proof procedure which is top-down: from conclusion to hypothesis, that is goal-directed. The originality of this procedure is that it does not act as classical theorem proving procedures, by requiring a special form of expressions, such as clausal form, obtained after skolemisation. We show, with our procedure, that this step is useless, and that the notion of piece allows inferring directly on dependencies, without dynamically creating new symbols. With the recent introduction of constrained dependencies, some interesting perspectives also arise.

1 Introduction

Dependency theory allows the expression and modelling of constraints that the data must satisfy in order to reflect correctly the world that a relational database intends to describe. Since the introduction of functional dependencies by Codd ([Cod72]), many kinds of dependencies have been studied in the litterature, and a lot of work has been carried out in the late 70’s and early 80’s. Database dependencies theory is still an active area of research [SF00] [Her95] [LL97a] [LL97b] [LL98]. Functional and multivalued dependencies are the most known classes of data dependencies. In practice, these two kinds are sufficient in general to express constraints ([Ul88]). Nevertheless, more general classes have been introduced, with the purpose of finding a uniform way to express constraints ([BV81],[SU82]). This paper deals with the class commonly known for generalizing most of the dependencies: that of tuple-generating and equality-generating dependencies (TGDs and EGDs) ([BV84b]). For a survey on this general class of dependencies, we refer the reader to [FV86].

© Springer-Verlag Berlin Heidelberg 2000
The central problem is the implication problem, which is to decide whether a dependency is logically implied by a given set of dependencies. A process that could solve this problem would provide a solution to find the minimal cover of a set of dependencies, to decide whether a dependency is redundant within a set, useful during the constraint acquisition stage, etc. A procedure has already been designed in [BV84b] for that purpose: the well-known *chase*. Unfortunately, as the implication problem for TGDs and EGDs is semi-decidable, the *chase* is only a proof procedure, and therefore the process may run forever. As we argue in this paper, the *chase* is clearly a bottom-up procedure: from hypotheses to conclusion. Also, the *chase* requires the dynamic creation of new symbols.

We introduce a new proof procedure which is top-down: from conclusion to hypothesis that is goal-directed. The originality of this procedure is that it does not act as classical theorem proving procedures, by requiring a special form of expressions, such as clausal form, obtained after skolemisation. We show, with our procedure, that this step is useless, and that the notion of *piece* allows inferring directly on dependencies without dynamically creating new symbols.

Therefore, our top-down chase is not simply the usual chase reversed, but a new way of solving the implication problem. The fact it can be performed top-down is the first contribution of this paper. The second contribution is to avoid dynamic creation of symbols, as well as skolemization usually applied on the original formulae prior to top-down proofs. This is realized by taking advantage of the form of the dependencies. To our knowledge, this has not been realized before. Indeed, dynamic creation of symbols can be a costly operation in general proof procedures, such as the *chase*, in order to take into account the effect of existential quantifiers.

While it is true that top-down approaches can take exponential time longer w.r.t. bottom-up approaches, many reasons allow us to think that the proof procedure presented in this paper can be efficient. These arguments will be detailed hereinafter. The improvement is currently being formally assessed.

Recently, constrained dependencies have been introduced ([Mah94], [BCW95], [Mah97]). They originate in the constraint programming field and permit expression of semantic relations on variables, thus giving them an interpretation. The chase procedure has been redesigned in [MS96], still in a bottom-up way, in order to deal with constrained tuple-generating dependencies. This work in the dependency theory gives new perspectives for the top-down chase procedure we present.

The top-down chase originates in the conceptual graphs model, which is a knowledge representation model introduced by Sowa ([Sow84]). The base model has been extended with graph rules and an inference method, called *piece resolution* ([SM96]). The logical roots of this process have been studied in [CS98], and constitute the basis of the top-down chase. Proofs of the lemmas and theorems of this paper are therefore derived from these last-mentioned.

Section 2 describes the framework and the implication problem for data dependencies. We sketch the existing (bottom-up) chase. In section 3 the top-down chase is explained. Section 4 closes the paper with some concluding remarks. For