Preferred Repairs for Inconsistent Databases

Sergio Greco  
DEIS Università della Calabria, Italy

Cristina Sirangelo  
DEIS Università della Calabria, Italy

Irina Trubitsyna  
DEIS Università della Calabria, Italy

Ester Zumpano  
DEIS Università della Calabria, Italy

INTRODUCTION

The objective of this article is to investigate the problems related to the extensional integration of information sources. In particular, we propose an approach for managing inconsistent databases, that is, databases violating integrity constraints. The problem of dealing with inconsistent information has recently assumed additional relevance as it plays a key role in all the areas in which duplicate information or conflicting information is likely to occur (Agarwal et al., 1995; Arenas, Bertossi & Chomicki, 1999; Bry, 1997; Dung, 1996; Lin & Mendelzon, 1999; Subrahmanian, 1994).

As known, the presence of inconsistent data can be resolved by repairing the database, that is, by providing a computational mechanism that ensures obtaining consistent “scenarios” of the information or by consistently answering queries posed on an inconsistent set of data.

Example 1

Consider the following database schema consisting of the single relation Teaches (Name, Faculty, Course) with the functional dependency Name → Faculty. Assume to have three different sources for the relation Teaches containing, respectively, the tuples Teaches(john, science, databases), Teaches(john, engineering, algorithms), and Teaches(john, science, operating_systems). The three different source relations satisfy the functional dependencies, but from their integration, we derive the inconsistent relation D= \{(john, engineering, algorithms), (john, science, operating_systems), (john, science,databases)\}.

The integrated relation D can be repaired by applying a minimal set of update operations. In particular, it admits two repairs: R1 obtained by deleting the tuple (john, engineering, algorithms) and R2 obtained by deleting the two tuples (john, science, databases) and (john, science, operating_systems).

In the presence of an alternative set of repairs, it is natural to allow user expressing preferences. In particular, methods for ranking and returning the preferred information is an increasingly key goal in AI applications ranging from information filtering and extraction to user profiling. The importance of this issue is reflected by an extensive number of proposals allowing the specification of preferences (Brewka & Eiter, 1999; Delgrande & Schaub, 2000; Gelfond & Son, 1997; Sakama & Inoue, 2000). This article is a contribution in this direction as it aims at filtering out the preferred repairs in the presence of preferences. For instance, if in the above example, preferred repairs are those minimizing the number of deletion and insertion of tuples, then the repair R1 is preferred to the repair R2.

In this article, we consider preferences among repairs and possible answers by introducing a partial order among them on the base of some preference criteria. More specifically, preferences are expressed by considering polynomial functions applied to repairs and returning real numbers. The goodness of a repair is measured by estimating how much it violates the desiderata conditions, and a repair is preferred if it minimizes the value of the polynomial function used to express the preference criteria.

Note that, while integrity constraints can be considered as a query which must always be true after a modification of the database, the conditions expressed by the evaluation function can be considered as a set of desiderata which are satisfied if possible by a generic repair. The goodness of a repair is measured by estimating how much the updates to be performed on the inconsistent database respect the preference criteria or, in other words, how much the repaired database violates them.

The main contribution of this work consists in the proposal of a logic approach for querying and repairing inconsistent databases that extend previous works (Greco,
Preferred Repairs for Inconsistent Databases

Greco & Zumpano, 2001; Greco & Zumpano, 2000) by allowing to express and manage preference criteria. The approach here proposed allows to express reliability on the information sources and is also suitable for expressing decision and optimization problems. The introduction of preference criteria strongly reduces the number of feasible repairs and answers; for special classes of constraints and functions, it gives a unique repair and answer.

BACKGROUND

A database D has associated schema DS = (Rs,IC) which defines the intentional properties of D: Rs denotes the set of relation schemas whereas IC contains the set of integrity constraints. Integrity constraints express semantic information over data, that is, relationships that must hold among data in the theory. Generally, integrity constraints represent the interaction among data and define properties which are supposed to be explicitly satisfied by all instances over a given database schema. Therefore, they are mainly used to validate database transactions.

Definition 1

A full (or universal) integrity constraint is a formula of the first order predicate calculus of the form:

\[(\forall X)[B_1 \land \ldots \land B_n \land \varphi \supset A_1 \lor \ldots \lor A_m \lor \psi_1 \lor \ldots \lor \psi_k]\]

where A1,...,Am, B1,...,Bn are base positive literals, \(\varphi, \psi_1, \ldots, \psi_k\) are built-in literals, X denotes the list of all variables appearing in B1,...,Bn and it is supposed that variables appearing in A1,...,Am, \(\varphi, \psi_1, \ldots, \psi_k\) also appear in B1,...,Bn.

In the definition above, the conjunction \(B_1 \land \ldots \land B_n\) \(\varphi\) is called the body and the disjunction \(A_1 \lor \ldots \lor A_m \lor \psi_1 \lor \ldots \lor \psi_k\) the head of the integrity constraint. Moreover, an integrity constraint is said to be positive if no negated literals occur in it (classical definitions of integrity constraints only consider positive nondisjunctive constraints, called embedded dependencies (Kanellakis, 1991).

Often, we shall write our constraints in a different format by moving literals from the head to the body and vice versa.

PREFERRED REPAIRS FOR INCONSISTENT DATABASES

In this section, we introduce a polynomial function, through which, expressing preferences criteria. The function introduces a partial order among repairs to allow the evaluation of the goodness of a repair for an inconsistent database. Moreover, we define preferred repairs as feasible repairs that are minimal with respect to the partial order.

Let us first recall the formal definition of consistent database and repair.

Definition 2

Given a database schema DS = (Rs,IC) we say that IC is consistent if there exists a database instance D over DS such that D \(\models IC\). Moreover, we say that a database instance D over DS is consistent if D \(\models IC\), that is, if all integrity constraints in IC are satisfied by D, otherwise it is inconsistent.

Intuitively, a repair for a (possibly inconsistent) database D is a minimal consistent set of insert and delete operations which makes D consistent, whereas a consistent answer for a query consists of two sets containing, respectively, the maximal set of true and undefined atoms which match the query goal; atoms which are neither true nor undefined can be assumed to be false.

More formally: see definition 3.

Definition 3

Given a database schema DS = (Rs,IC) and a database D over DS a repair for D is a pair of sets of atoms (R+,R-) such that 1) \(R+ \cap R = \emptyset\), 2) \(D \cup R+ - R \models IC\) and 3) there is no pair \((S+, S-)\) such that \(R+ \subset S+\), \(R \subset S-\) and \(D \cup S+ - S- \models IC\). The database \(D \cup R+ - R\) will be called the repaired database.

Thus, repaired databases are consistent databases which are derived from the source database by means of a minimal (under total semantics) set of insert and deletion of tuples. Given a repair R for D, \(R^g\) denotes the set of tuples which will be added to the database, whereas \(R^d\) denotes the set of tuples of D which will be canceled. In the following, for a given repair R and a database D, R(D) = \(D \cup R^g - R^d\) denotes the application of R to D.

Moreover, given a database schema DS, we denote with D the set of all possible database instances over DS.

Definition 4

Given a (possibly inconsistent) database D over a fixed schema DS and a polynomial function \(f: (D, D) \times D \rightarrow \mathbb{R}\). A repair R1 is preferable to a repair R2, w.r.t. the function f, written \(R_1 < R_2\) if \(f(R_1, D) \leq f(R_2, D)\).

A repair R for D is said to be preferred w.r.t. the function f if there is no repair \(R'\) for D such that \(R' < R\). A repaired
Related Content

**INDUSTRY AND PRACTICE: INDUSTRY NEWSBRIEFS**

*Database Pointers in Navigational and Object–Oriented Database Management Systems: A Comparison*
[www.igi-global.com/article/database-pointers-navigational-object-oriented/51155?camid=4v1a](www.igi-global.com/article/database-pointers-navigational-object-oriented/51155?camid=4v1a)

*The Evolution of the Meta-Data Concept: Dictionaries, Catalogs, and Repositories*
[www.igi-global.com/article/evolution-meta-data-concept/51122?camid=4v1a](www.igi-global.com/article/evolution-meta-data-concept/51122?camid=4v1a)

[www.igi-global.com/article/rich-base-schema/3247?camid=4v1a](www.igi-global.com/article/rich-base-schema/3247?camid=4v1a)