Controlled Generation of Intensional Answers *

Alain PIROTTE† Dominique ROELANTS‡ Esteban ZIMANYI§

Abstract

Intensional answers are conditions that tuples of values must satisfy to belong to the usual extensional answer of a query addressed to a deductive database. This paper motivates the concept of intensional answers and introduces a general method for generating them as logical consequences of the query and of deduction rules. It then shows how integrity constraints can filter out inadequate answers and produce simpler and more informative answers. An efficient organization for the combination of answers and constraints is described. Beyond the mechanics of answer generation, the interest of the approach also depends on a strategy for selecting answers to a user submitting a query. This requires techniques for user modeling and dialogue management similar to those required for expert systems.

1 Introduction

A deductive database comprises an extensional part and an intensional part. The extensional database (or EDB) contains explicit tuples of base relations while the intensional database (or IDB) comprises deduction rules and integrity constraints. Deduction rules specify sets of tuples for virtual relations in terms of base relations and of other virtual relations. Integrity constraints state conditions that the extension of virtual and base relations must satisfy.

Rules, constraints, and queries are written in a database language, typically based on a first-order language with a predicate associated with each relation (base or virtual) of the database. A query $Q(\bar{z})$ to the database is a formula of the database language with free variables $\bar{z}$ called query variables. Its extensional answer is the set of tuples $\bar{a}$ such that $Q(\bar{a})$ can be shown to be true on the extensional database when the deduction rules defining the virtual predicates in $Q$ are taken into account.

Informally, an intensional answer to a query $Q(\bar{z})$ is a formula $A(\bar{z})$, obtained from the intensional database and from the query, that states a condition to be satisfied by tuples of values for $\bar{z}$ in order to be part of the extensional answer of that query.

A great variety of intensional answers can be constructed for a given query. A first idea is to construct a single intensional answer equivalent to the initial query by replacing a virtual predicate by the disjunction of all its partial definitions in deduction rules.

*A complete version of this paper will appear in IEEE Transactions on Knowledge and Data Engineering. We are grateful for having received the permission to republish it.

†Philips Research Laboratory Belgium, 4, Avenue Albert Einstein, 1348 Louvain-la-Neuve, Belgium, e-mail: pirotte@prlb.philips.be
‡Philips Research Laboratory Belgium, 4, Avenue Albert Einstein, 1348 Louvain-la-Neuve, Belgium, e-mail: dominique@prlb.philips.be
§Université Libre de Bruxelles, C.P. 125, 50 avenue Roosevelt, 1050 Brussels, Belgium, e-mail: ez@prlb.philips.be
Then another intensional answer would be constructed by making explicit the definition of another virtual predicate, and so on. However, after a few replacements, the formula obtained would become complex and intensional answers would lose their intuitive appeal. In fact, a major interest of intensional answers is that they provide compact and intuitive characterizations of sets of facts, making explicit why a specific set of facts answers the query instead of just which facts belong to the answer. Therefore, at least for human consumption, a complex intensional answer would have to be simplified and decoded to be of any use.

Thus, a better approach is to construct a collection of reasonably simple intensional answers, each of which states a sufficient condition for a set of facts to belong to the extensional answer of the query. The question will then be to decide when enough intensional answers have been generated to provide a sufficient answer for the query.

An important aspect of intensional answers is that they are constructed without accessing the extensional database. Since the latter is typically much larger than the intensional database, they can be constructed more efficiently than the extensional answer. Also, it is usually the case that the extensional database changes more often through updates of individual facts than the intensional database. Therefore, intensional answers provide a more stable answer than the extensional answer.

An intensional answer is similar to a query and, if desired, it can be evaluated against the extensional database like a query and return a partial extensional answer. Intensional answers then appear as a kind of explanation or justification of the adequacy of partial extensional answers.

An important contribution of this paper is the design of mechanisms for controlling the generation of intensional answers through integrity constraints. The assertions specified by constraints bear on the database but also on queries and intensional answers, since the latter specify sets of values from the database. An intensional answer interacts with a constraint when both have predicates in common. Constraints thus relevant to answers can be used to transform some answers, most importantly to simplify them, as shown in the following example.

**Example 1.** In the presence of the constraints "all salaries of employees are lower than 20" and "all employees have a salary", the query "list the employees with a salary lower than 20" has the simple intensional answer "all the employees". As a special case of transformation, constraints may show that some answers will always be empty when evaluated against the extensional database.

If a constraint contains a virtual predicate, then the deduction rules defining the virtual predicate somehow supplement the constraint. Another constraint may be deduced by replacing the virtual predicate in the first constraint by its (partial) definition according to a deduction rule. Both constraints are candidates for interacting with an intensional answer depending on which predicates are present in the answer.

**Example 2.** If we add to Example 1 the rule "all technicians are employees", then the constraint "all salaries of technicians are lower than 20" may be derived. Thus, controlling the generation of intensional answers with constraints inherently leads to handling large sets of constraints and, therefore, to the necessity of managing them tightly.

The paper is structured as follows. Section 2 recalls basic definitions and assumptions for deductive databases. Section 3 gives a precise definition for the concept of intensional answer. Section 4 presents the principle of generating candidate intensional answers from