DL Techniques for Intensional Query Answering in OODBs

Sonia Bergamaschi* , Claudio Sartori**, Maurizio Vincini*

Int. Workshop KI-95. editor Dreschler-Fischer and S. Pribbenow, Bielefeld, Germany 1995

1 Introduction

It is well known that in general, a query issued on a database can be rewritten in many ways maintaining, as a result, the same set of items (say, records or objects, depending on the data model). Such rewriting has been devised with the main purpose of query optimization, i.e. to minimize the execution costs. Traditionally, database theory focused on algebraic rewriting, which depends only on formal properties of the data model and manipulation language. Some works introduced also the idea of semantic query optimization[10, 3, 4, 2], which rewrites queries also on the basis of semantic problem-specific knowledge, such as integrity constraints.

In this paper we exploit the idea of rewriting a query not only for the semantic optimization task, as proposed by the authors in [3, 4], but also for another query-related task: intensional query answering. In particular, we focus on Object Oriented Databases and give a general definition of *semantic transformation* and of *semantic expansion* of a query. Then we will show how this concepts can be exploited in intensional query answering.

2 Semantic transformation and expansion of a query

In database environment, semantic knowledge is usually expressed in terms of *Integrity Constraints* (IC) rules, that is *if then* rules on the attributes of a *database schema* (i.e., roughly a Tbox of a Terminological Knowledge Representation System) to guarantee data consistency. In this context, we can say that a query Q' is a *semantic transformation* of the query Q if it gives the same result of Q for any database instance which satisfy the given IC rules.

In [3, 4] the authors proposed a method for semantic query optimization, applicable to the class of conjunctive queries, based on two fundamental ingredients. The first one is the *ODL* description logics proposed as a common formalism to express: class descriptions, a relevant set of IC rules and queries as ODL types. The second one are DL (Description Logics) inference techniques exploited to evaluate the logical implications expressed by IC rules and thus to produce the *semantic expansion* of a given query. The semantic expansion of a query is a semantic transformation of a query which incorporates any possible restriction which is not present in the original query but is *logically implied* by the query and by the overall schema (classes + IC rules).

ODL (Object Description Logics) was proposed in [5] and extends the expressiveness of implemented description logics languages in order to represent the semantics of complex object data models (*CODMs*), recently proposed in the areas of deductive databases [1] and object oriented databases [8]. The present version of ODL allows the declarative formulation of a relevant set of database integrity constraints. Actual database schemata are, in fact, given in terms of base classes (i.e. primitive concepts) while further knowledge is expressed with IC rules. In general, integrity constraints go beyond data model expressiveness and are expressed in various fashions, depending on the database data model: e.g. subsets of first order logic, inclusion dependencies and predicates on row values, procedural methods in OO environments. In particular, ODL includes *quantified path types* and IC rules. The former extension has been introduced to deal easily and powerfully with nested structures. Paths, which are essentially sequences of attributes, represent the central ingredient of OODB query languages to navigate through the aggregation hierarchies of classes and types of a schema. In particular, we provide *quantified* paths to navigate through multi-valued attributes. The allowed quantifications are existential and universal and they can appear more than once in the same path.

*Dip. di Scienze dell'ingegneria, Università di Modena, Via G. Campi 213/B, I-41100 Modena, Italy.
**Dip. di Elettronica, Informatica e Sistemistica, Università di Bologna, Viale Risorgimento 2, I-40136 Bologna, Italy.
e_mail: { sbergamaschi, csartori}@deis.unibo.it

Viewing a database schema as a set of ODL *inclusion statements* allows the declarative formulation of another relevant set of integrity constraints, expressing *if then rules* whose antecedent and consequent are ODL *virtual* types (i.e. defined concepts). For example, it is possible to express correlations between structural properties of the same class or sufficient conditions for populating subclasses of a given class. A *generalized database schema* can be thus defined as a set of inclusion statements between general ODL types.

A relevant set of queries, corresponding to the so called single-operand queries [7], can be expressed as *virtual* ODL types. Subsumption computation, incoherence detection and canonical form generation proposed in [5] can be used to produce the *semantic expansion* $EXP(Q)$ of a query Q. Following the approach of [10], we perform the semantic expansion of the types included at each nesting level in the query description. Type expansion is based on the iteration of this simple transformation: if a type implies the antecedent of an IC rule then the consequent of that rule can be added. Logical implications between these types (the type to be expanded and the antecedent of a rule) are evaluated by means of *subsumption computation* [5].

Semantic expansion is an iterative process which produces, at any step, a query which is semantically equivalent to the original one. Moreover, during the transformation, we compute and substitute in the query, at each step, the maximal subsumed classes satisfying the query. Therefore, each of the intermediate results of semantic expansion is a valid semantic transformation of the query and is a candidate for the intensional answer. The result of semantic expansion of a query coincides with the *lowest query* in the taxonomy among all the semantically equivalent ones [3].

In general, semantic expansion can also lead to introduce redundant terms, i.e. terms which are *logically implied* by other terms. In the literature, this problem is generally addressed as *constraint removal*, that is the removal of the constraints which are logically implied by the query. We can then detect in the expanded query, again by subsumtion, all the eliminable factors and, eventually, eliminate them [2].

3 DL techniques for intensional query answering

In [9] Motro gives an overview of the various intensional query answering techniques. On the basis of that classification, intensional query answering can be evaluated according to three main features: intensional-only (*pure*) versus intensional/extensional (*mixed*); *independence* from the database instance versus dependence; *completeness* of the characterization of the extensional answer.

In general, a query is expressed as a class of the schema (target class) restricted with additional selection predicates, which include conditions on objects of the aggregation rooted at the target class. The many queries obtained by semantic expansion will differ from the original one either for the target class or for the predicates. Each transformed query is a possible intensional answer, which is *pure*, since it does not contain reference to any extensional element, and also *independent*, since it is computed according to general IC rules which hold in any database state. Thus it is also *intension-equivalent*.

For example, in a database with an integrity constraint stating that all employees who lead a department are managers, a query on the employees who lead a department and earn more than $ 50000 is equivalent to a query on the managers who earn more than $ 50000. Conversely, in a database with an integrity constraint stating that all engineers earns over $ 40000, a query on the engineers who earn over $ 30000 is equivalent to a query on all the engineers.

When several different intensional answers are available, a main issue is to determine which answer is the "best". We give the following criteria for the best answer:

- the target class is the most specialized among the ones satisfying the query
- the classes included in the query predicates are the most specialized satisfying the query
- redundant predicates are removed.

The three above criteria are satisfied by the application of semantic expansion and constraint removal.

As far as completeness of intensional characterization is concerned, pure independent intensional answers, including ours, are intrinsically incomplete. The only exception is [6], but it imposes the strong limitation of queries corresponding only to classes of the taxonomy.

With reference to the completeness of our method, which is based on subsumption, it is well known that it is greatly influenced by the complexity of the knowledge representation model or, in our case, of the data and integrity constraint definition language. If the language does not allow completeness of subsumption, the intensional answer we get is correct, but not necessarily the most concise.

A different approach could be the generation of an intensional answer which is equivalent to the original one only for the present database instance. For example, let us suppose that, for a given database state, a query on the employees who earn between \$30000 and \$50000 return only employees who are engineers. In this case, the answer "all the engineers" is *pure* and *dependent*, i.e. it is *extension-equivalent* to the original one. Unlike the previous case, this method does not avoid data access, but can be driven by schema knowledge. For example, given the classes A and B, the query Q and the subsumption hierarchy $A \sqsupseteq Q \sqsupseteq B$, the query Q is extensionally-equivalent to B if $A - B = \emptyset$. This result can be obtained without accessing the extension if the database system provides an efficient way to deal with classes cardinalities.

With respect to a relational framework, in [9] the concept of view is central, views are used both for expressing queries and also for expressing integrity constraints (in particular, an integrity constrains is rewritten as an empty view). The problem becomes: given a query and a set of constraints, what are the constraints which apply to the answer? The author's solution is to introduce meta-relations, each of one stores the meta-tuples which synthesizes the views on a relation and whose structure is identical to the one of the given relation. In this way intensional answers characterize the extensional answers both with constraints and containments, i.e. views that are contained entirely in the intensional answer.

Note that, our approach can be easily exploited to achieve the same results of the above work in an extended data model framework. In fact, the first result can be easily achieved by selecting the integrity constraints which refers to a given query (it can be done by tracing the applicable integrity constraints during the query semantic expansion activity). The latter result easily follows from the application of DL techniques and query semantic expansion: by computing the maximal classes subsumed by the query and the integrity constraints antecedents subsumed by the query, we can give the list of computed containments. In this way, in analogy with [9] we give a partial intensional answer.

References

[1] S. Abiteboul and P. Kanellakis. Object identity as a query language primitive. In *SIGMOD*, pages 159–173. ACM Press, 1989.

[2] J.P. Ballerini, D. Beneventano, S. Bergamaschi, C. Sartori, and M. Vincini. A semantics driven query optimizer for oodbs. In *DL95*, June 1995.

[3] D. Beneventano, S. Bergamaschi, S. Lodi, and C. Sartori. Using subsumption in semantic query optimization. In A. Napoli, editor, *IJCAI Workshop on Object-Based Representation Systems - Chambery, France*, August 1993. DFKI Technical Report D-94-10.

[4] D. Beneventano, S. Bergamaschi, S. Lodi, and C. Sartori. Terminological logics for schema design and query processing in oodbs. In *KRDB'94 - Reasoning about Structured Objects, Knowledge Representation Meets Databases*, Saarbruecken, Sept. 1994, 1994.

[5] S. Bergamaschi and B. Nebel. Acquisition and validation of complex object database schemata supporting multiple inheritance. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks and Complex Problem Solving Technologies*, 4:185–203, 1994.

[6] F. Corella. Semantic retrieval and levels of abstraction. In *1st Int. Wrokshop on Expert Database Systems*, pages 91–114, 1984.

[7] W. Kim. A model of queries for object-oriented database systems. In *Int. Conf. on Very Large Databases*, Amsterdam, Holland, August 1989.

[8] C. Lecluse and P. Richard. Modelling complex structures in object-oriented databases. In *Symp. on Principles of Database Systems*, pages 362–369, Philadelphia, PA, 1989.

[9] A. Motro. Intensional answers to database queries. *IEEE Trans. on Knowledge and Data Engineering*, 6(3):444–454, 1994.

[10] S. Shenoy and M. Ozsoyoglu. A system for semantic query optimization. *ACM-SIGMOD*, pages 181–195, May 1987.