DL Techniques for Intensional Query Answering in OODBs

Sonia Bergamaschi* and **Claudio Sartori°** and **Maurizio Vincini***
CIOC - CNR
* Dipartimento di Scienze dell'ingegneria, Università di Modena
Via G. Campi 213/B, I-41100 Modena, Italy
°Dipartimento di Elettronica, Informatica e Sistemistica, Università di Bologna
Viale Risorgimento 2, I-40136 Bologna, Italy
e_mail: { sbergamaschi, csartori}@deis.unibo.it

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[Shenoy and Ozsoyoglu, 1987, Beneventano *et al.*, 1993, Beneventano *et al.*, 1994, Ballerini *et al.*, 1995], 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 [Beneventano *et al.*, 1993, Beneventano *et al.*, 1994], 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

Actual database schemata are, in fact, given in terms of base classes (i.e. primitive concepts) while further knowledge is expressed with *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 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 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 [Beneventano *et al.*, 1993, Beneventano *et al.*, 1994] 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 is the subsumption inference technique 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 [Bergamaschi and Nebel, 1994] 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 [Abiteboul and Kanellakis, 1989] and object oriented databases [Lecluse and Richard, 1989]. In particular, class types and complex value-types are differentiated. They are based on base types: integers, strings, reals, and are constructed with the recursive use of the set and record constructors. The present version of ODL allows the declarative formulation of a relevant set of database integrity constraints. 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.

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 [Kim, 1989], can be expressed as *virtual* ODL types. Subsumption computation, incoherence detection and canonical form generation proposed in [Bergamaschi and Nebel, 1994] can be used to produce the *semantic expansion* $EXP(Q)$ of a query Q. Following the approach of [Shenoy and Ozsoyoglu, 1987], 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* [Bergamaschi and Nebel, 1994].

Semantic expansion is an iterative process which produces, at any step, a query which is semantically equivalent to the original one. During the transformation, we compute and substitute in the query, at each step, the maximal subsumed classes, among the classes of the schema, 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 [Beneventano *et al.*, 1993].

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 subsumption, all the eliminable factors and, eventually, eliminate them [Ballerini *et al.*, 1995].

3 DL techniques for intensional query answering

An overview of the various intensional query answering techniques is given in [Motro, 1994]. 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:

1. the target class is the most specialized among the classes of the schema that can be substituted for the original one in the query, therefore it gives a concise description of the answer which is more informative than the original query;
2. the classes included in the query predicates are the most specialized satisfying the query, giving a more significant, though semantically equivalent, predicate;
3. redundant predicates are removed as a contribution to conciseness.

The three above criteria are satisfied by the application of semantic expansion and constraint removal. In particular, according to the criterion 1 a query like *which are the X such that p_1 and ... and p_n* gets the answer *all the X' such that p_1 and ... and p_m*, where X' is subsumed by X and $m \leq n$. If we consider the first example above, we substitute the target class "employees", which can contain many thousands of items, with the class "managers" which can contain few hundreds of items and the answer, though purely intensional characterizes the result in terms of a more restricted class than the original query.

As far as completeness of intensional characterization is concerned, our rewriting method is exact, therefore each rewrited answer is a complete characterization of the original query.

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 not necessarily the most concise.

Given a query Q, subsumption can also be used to compute its Greatest Lower Bound (GLB) and Least Upper Bound (LUB) among the classes of the schema. For simplicity, let us suppose in the following that the two bounds are unique. In this case $LUB_Q \sqsupseteq Q \sqsupseteq GLB_Q$ and each bound can be seen as a *partial* intensional answer to the query.

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 query Q and its bounds LUB and GLB, the query Q is extensionally-equivalent to B if $LUB_Q - GLB_Q = \emptyset$. This result can be obtained without accessing the extension if the database system provides an efficient way to deal with classes cardinalities.

Hybrid reasoning can be used to obtain *mixed* intensional answers. In this case, the aswer contains intensional concepts and lists of positive and negative extensional items. Given an algorithm for the *instance problem*, which can decide if an object belogns to a given class, the answer to a query Q can one of the following:

$$\begin{array}{lcl} LUB_A & - & \{i_1, i_2, \ldots, i_n\} \\ & \textit{where} & i_j \in \{LUB_Q - GLB_Q\} \wedge i_j \notin Q \\ LUB_A & \cup & \{i_1, i_2, \ldots, i_n\} \\ & \textit{where} & i_j \in \{LUB_Q - GLB_Q\} \wedge i_j \in Q \end{array}$$

For instance, the query "who earns more than \$30000" could get the answer "all the engineers extept John Smith".

The usability of this technique is obviously related to the efficiency of the algorithm for the instance problem, since it has to be computed many times.

As a final remark, we mentioned in the beginning that the rewriting activity is based on a schema including integrity rules. Of course, if more integrity rules are available more rewritings are possible. For the sake of intensional answers, one could apply *data mining* techniques to discover new rules [Cercone and Tsuchiya, 1993]. The rewritings made possible by these rules give answers which are *dependent* from the present database state.

References

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

[Ballerini *et al.*, 1995] J.P. Ballerini, D. Beneventano, S. Bergamaschi, C. Sartori, and M. Vincini. A semantics driven query optimizer for oodbs. In A. Borgida, M. Lenzerini, D. Nardi, and B. Nebel, editors, *DL95 - Intern. Workshop on Description Logics*, pages 59–62, Roma, June 1995.

[Beneventano *et al.*, 1993] 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*, 1993.

[Beneventano *et al.*, 1994] 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.

[Bergamaschi and Nebel, 1994] 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.

[Cercone and Tsuchiya, 1993] Nick Cercone and Mas Tsuchiya, editors. *IEEE Transactions on Knowledge and Data Engineering, Vol.5, N.6.* IEEE, 1993. Special issue on Learning and Discovery in Knowledge-Based Databases.

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

[Lecluse and Richard, 1989] 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.

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

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