Testing Containment of Object-Oriented Conjunctive Queries is \( \Pi_2^p \)-hard

Edward P.F. Chan* Ron van der Meyden
Department of Computer Science Information Science Laboratory
University of Waterloo NTT Basic Research Laboratories
Waterloo, Ontario, Canada N2L 3G1 Atsugi-shi, Kanagawa 243-01, Japan

May 27, 1995

Abstract

We study the complexity of testing containment for a class of object-oriented conjunctive queries. We show that the containment problem is \( \Pi_2^p \)-hard. Together with a previous result, the containment problem is complete in \( \Pi_2^p \).

1 Introduction

The problem of query optimization is difficult in an object-oriented database system (OODB). A natural first step is to use the typing constraints imposed by the schema to transform a query into an equivalent one that logically accesses a minimal set of objects. In an OODB, classes are named collections of similar objects. A class \( C \) could be refined into subclasses. Conversely, the class \( C \) is said to be a superclass of its subclasses. Subclasses are specialization of their superclasses. Consequently, objects in a subclass are also objects in its superclasses. Specialization of a class is often achieved by refining and/or adding properties to its superclasses. Since properties of a superclass are also properties of its subclasses, a subclass is said to inherit the properties of its superclasses. Classsubclass relationships form an acyclic directed graph called inheritance or generalization hierarchy.

Inheritance is a powerful modeling tool, because it gives a better structured and more concise description of the schema and it helps in factoring out shared implementations in applications. Objects belonging to the same class share some common properties. Properties are attributes or methods defined on types; they are applicable only to the instance of the types. In effect, therefore, types are constraints imposed on objects in the classes. Properties are formally denoted as attribute-type pairs in this paper. A natural first step in query optimization is to use the typing constraints implied by the schema to minimize the search space for variables involved in the query.

Example 1.1 The following is a schema for a vehicle rental database. It keeps track of all rental transactions for vehicles in the company. In this application, Auto, Trailer and

*Part of this work was done while the author visited NTT Basic Research Laboratories in Tokyo.
Truck are subclasses of the superclass Vehicle. There are clients, called discount customers, who are known to the company and receive special treatments. Discount customers receive a special rate and are not required to have deposit on the vehicles rented. However, discount customers only allow to rent automobiles, and not other types of vehicles. Let us assume further that all superclasses are partitioned by their respective subclasses.

Suppose we want to find out all those vehicles that have been rented to a discount client. Express in a calculus-like language, the query looks like:

$$Q_1: \{ x \mid \exists y \ (x \in \text{Vehicle} \land y \in \text{Discount} \land x \in y \cdot \text{VehRented}) \}.$$  

Since discount clients are allowed to rent Auto only, the above query is equivalent to the following query:

$$Q_2: \{ x \mid \exists y \ (z \in \text{Auto} \land y \in \text{Discount} \land x \in y \cdot \text{VehRented}) \}.$$  

$$Q_2$$ is considered to be more optimal since the number of variables as well as their search spaces are minimal, given the typing constraints implied by the schema.

In this paper, we shall define a class of queries called conjunctive queries for an object-oriented database. In general, variables in an object-oriented query range over heterogeneous sets of objects. This constitutes a significant divergence from its relational counterpart. As shown in [3], to solve the equivalence and optimization problems, we need first to resolve the satisfiability and containment problems. Let us consider the following example to see how the containment problem in our case is different from the containment problem in the relational model.