The Many Faces of Query Monotonicity*

Catriel Beeri‡ Yoram Kornatzky‡
The Hebrew University
Jerusalem 91904, ISRAEL

Abstract

Monotonicity, based on the partial order defined by the 'is a subset of' relation, is a well understood property of queries. For nested relations, other partial orders leading to different notions of monotonicity are possible. Monotonicity can be used for simple negative comparison of the expressive power of two languages by showing that one is monotone and the other is not. Using this approach we study three questions related to the expressive power of practically useful subsets of well known programming languages for nested relations. First, we show that logic programming languages over nested relations can be regarded as Datalog with user-defined algebraic expressions. This leads to a modular integration of recursion with the monotone subset of the algebra. Second, we prove that the equivalence of the powerset algebra and the complex object Datalog breaks down for their monotone subsets. Third, for the class of positive existential queries over nested relations, which generalize the relational tableau set queries, we show that the use of intermediate types does not enhance their expressive power, in contrast to the known result for general existential queries. We also show that this class does not contain the powerset operator, hence it is a candidate for a tractable tableau query system for nested relations. Finally, the (monotone) Bancilhon-Khoshafian calculus for complex objects is shown to be incomparable to the monotone subsets of most known languages.

1 Introduction

New database applications have emerged in the last decade in fields such as CAD and software engineering. The deficiencies of the relational model for supporting these applications are well documented. This has led to the development of more general data models, in particular the nested relational model that generalizes the flat relational model by allowing relation-valued attributes. The languages proposed for this model are either algebraic [1,24], calculus-like [1,15], or deductive [7,9,17,18].

While a hierarchy of database queries in terms of complexity and expressive power is well established for the relational model [10], the power of query languages over nested relations and complex objects is less well understood. Results reported so far are mainly concerned with the high end of the complexity hierarchy. Two important classes of queries from flat databases to flat relations with intermediate nested types were identified: the computable queries (C) [11], and the elementary queries (E), computable in hyper-exponential time and space respectively [15]. Most of the languages for nested relations suggested in the literature were shown to be equivalent to the class of elementary queries when typed sets are used. These languages include the safe complex

*Research partially supported by grant 2545/3/89 of Israel National Council for Research and Development.
‡Department of Computer Science
‡Leibniz Center for Research in Computer Science
object calculus [1], the powerset algebra and its augmentations by programming constructs such as loops [14], COL [2], stratified Datalog over sets [9] and ELPS [19].

Practically, the lower end of the complexity and expressiveness hierarchy, consisting of limited hence efficiently tractable sublanguages, is more important. It is well known that the restricted classes of conjunctive, existential, and tableau queries form the backbone of the query processing and optimization strategies of most relational systems [26]. Complex objects database systems built so far, are also founded on efficiently implemented subsets of the nested relational algebra [8,12,22]. This paper is concerned with the expressive power of such limited, hence practically interesting, subsets of the full languages mentioned above, especially those that generalize practically useful relational languages. We avoid measuring expressive power through queries over flat relations, as this blurs the essential issues in manipulating nested relations especially for the sublanguages of interest here.

A relational query is monotone if increasing its input relations produces an equal or larger result. Thus, the partial order of set inclusion underlies the monotonicity definition. However, for nested relations or more general complex objects, other partial orders naturally arise. Indeed, deductive query languages for complex objects use a variety of lattices for defining their operational semantics [4,7,17]. In this paper, we investigate the expressive power of languages using monotonicity concepts defined by several partial orders. It turns out that for these partial orders monotonicity is preserved under two common operations for constructing queries: composition and least fixed point. Languages whose basic operations define monotone queries and in which queries are constructed using only composition or least fixed point, define only monotone queries and cannot express any non-monotone query. This argument provides a simple technique for deriving a negative comparison of the expressive power of languages.

In both algebraic and logic programming languages, queries can be applied to relation valued attributes of tuples. We show that nesting of queries yields only monotone queries. It follows that one can view Datalog over nested relations as the incorporation of user-defined algebraic expressions into Datalog. We study this combination of the procedural (algebraic) and declarative views of manipulating nested relations, and prove that the expressive power of Datalog is indeed enhanced by the introduction of user-defined expressions. This contrasts the fact that, for the powerset algebra, adding the nested application of functions is redundant [1]. We also show that the equivalence of the powerset algebra and the stratified complex object Datalog breaks down for their monotone subsets: the monotone algebra is strictly contained in the complex object Datalog.

Positive existential queries over nested relations generalize the useful relational tableau set queries [23]. These queries can involve quantification of variables ranging over intermediate types not appearing in the database or the output. For both general calculus queries and for existential ones, it is known that the set-height of intermediate types induces a strict hierarchy of query classes in terms of expressive power and computational complexity [15,20]. We prove that for positive existential queries the hierarchy collapses to its lowest level (not using intermediate types), and it cannot express the powerset operation. Hence, this class is a tractable useful sublanguage that can serve the same role for nested relations as that of tableau queries for flat ones.

The Bancilhon-Khoshafian calculus [7] is a deductive language for manipulating untyped complex objects. It can be regarded as the bare core of many others based on the notion of an object lattice, e.g. [4,17]. Its exact expressive power is unknown except that it is data-complete for the class of Turing computable data functions [16]. Its queries are monotone w.r.t. one of the partial orders we consider. We prove that it is incomparable to the monotone subsets of the languages mentioned above. This suggests that a straightforward compilation of this language into algebraic