Proceedings of the 8th Panhellenic Logic Symposium

July 4-8, 2011
Ioannina Greece

Hosted by the
Department of Computer Science
University of Ioannina

Rewriting real conjunctive queries using viewsets of minimal-size

Foto Afrati, Matthew Damigos
School of Electrical & Computing Engineering
National Technical University of Athens (NTUA)
15773 Athens, Greece
{afrati,mgdamig}@softlab.ece.ntua.gr

Manolis Gergatsoulis
Department of Archives and Library Science
Ionian University
Ioannou Theotoki 72, 49100 Corfu, Greece
manolis@ionio.gr

Abstract

In this paper we study the problem of finding sets of views of minimal size which can be used to compute the answers to a given set of queries. We consider *bag-set semantics* (i.e. duplicate tuples are allowed in the answers of the queries but not in the database) and focus on the class of *conjunctive queries* (i.e., a query given by a single non-recursive Datalog rule) which represents the class of select-project-join SQL queries with equality comparisons. The main results presented in this paper are: a) we show how to restrict the space of useful views, i.e., views that may participate to a solution of the problem, and b) we show that the problem of finding a minimal viewset is decidable, by giving a bounded space of viewsets in which there exists at least one minimal.

1 Introduction

In many database research senarios [1, 2, 3, 6], we want to find a set of views of minimal size (also called *minimal viewset*) that can be used to compute the answers to a given set of queries. This problem appears in many environments such as distributed databases [6], data integration [6], database security [3] and query optimization [9]. The problem of finding views to materialize in order to efficiently answer queries (using equivalent rewritings) has been extensively studied in the past under the name of *view selection problem* [1, 2].

Considering bag-set semantics (i.e., the base relations do not contain duplicate tuples, while the answers of queries may contain duplicates), the view selection problem has been solved only for conjunctive queries without self-joins (i.e., each relation name appears at most once in each query of the given set of queries) [1]. Bag-set semantics [4] practically constitute the best theoretical approximation of SQL semantics due to basic database design principles which do not recommend duplicate tuples in base relations (normalization rules) [6]. The problem of finding minimal viewsets has also been studied for set semantics in [5], where the authors prove that the problem is decidable for both conjunctive and disjunctive viewsets. Considering bag semantics (i.e., duplicate tuples are allowed in both base relations and queries answers), the decidability of the problem can be easily derived from [2].

In this paper we study this problem by considering bag-set semantics, that is, we investigate the problem of finding a viewset for a given set of queries and a given database instance such that there exist equivalent rewritings for all queries in the given set using this viewset, and the number of tuples resulted by applying the views on the given database instance are minimal (among all possible viewsets that give equivalent rewritings to the given queries). We focus on the class of *conjunctive queries* (i.e., a query given by a single non-recursive Datalog rule) which represents the class of select-project-join SQL queries with equality comparisons.

The main results presented in this paper are: a) we show how to restrict the space of useful views, i.e., views that may participate in a solution of the problem, and b) we show that the problem of finding a minimal viewset is decidable, by giving a bounded space of viewsets in which there exists at least one minimal.

2 Preliminaries

A *relation schema* is a named relation defined by its name R (called *relation name*) and a set A of *attributes*. A *relation instance* r for a relation schema is a collection of tuples over its attribute set. The schemas of the relations in a database constitute its *database schema*. A *relational database instance* (*database*, for short) is a collection of relation instances. A relation instance can be viewed either as a *set* or as a *bag* (or *multiset*) of tuples. In a *set-valued database*, all relations are sets; in a *bag-valued database*, multiset relations are allowed. The *bag-operators* [6] are similar to the set-operators. However, using bag-operators instead of set-operators, we have the ability to manipulate duplicate tuples as in bag-operators all occurrences of a certain tuple are treated as distinct tuples. Generalizing the conventional subset operator in order to handle bags, we say that a bag B_1 is a *subbag* of a bag B_2, denoted as $B_1 \subseteq_b B_2$, if every tuple $t \in B_1$ with multiplicity n_1, is also contained in B_2 with multiplicity n_2, where $n_1 \leq n_2$. Similarly, we say that two bags B_1 and B_2 are *equal*, denoted $B_1 =_b B_2$, if $B_1 \subseteq_b B_2$ and $B_2 \subseteq_b B_1$. Depending on whether a database is bag or set-valued and the operators are set or bag operators, the queries may be computed under *set semantics* (considering set-valued databases and set-operators), *bag semantics* (considering bag-valued databases and bag-operators), or *bag-set semantics* (considering set-valued databases and bag-operators). In this paper, we consider bag-set semantics, unless stated otherwise.

A *query* is a mapping from databases to databases, usually specified by a logical formula on the schema $\mathcal{S}$ of the input databases. Typically, the output database (called *query answer*) is a database with a single relation. In this paper we focus on the class of select-project-join SQL queries with equality comparisons, a.k.a. safe *conjunctive queries* (CQs for short). Formally, a CQ definition [6] is a rule of the form $Q : q(\overline{X}) :- g_1(\overline{X}_1), \ldots, g_n(\overline{X}_n)$, where $g_1, \ldots, g_n$ are database relations, $\overline{X}$ is a vector of variables and $\overline{X}_1, \ldots, \overline{X}_n$ are vectors of variables or constants. The atom $q(\overline{X})$ is the *head* of Q, denoted as $head(Q)$, while the atoms on the right of :- are said to be the *body* of Q, denoted as $body(Q)$. Each $g_i(\overline{X}_i)$ is also called a *subgoal* of Q. The variables in $\overline{X}$ are called *distinguished variables* of Q, whereas the variables appearing in the body of Q and not appearing in the head of Q are called *non-distinguished variable* of Q. We assume that each distinguished variable of a CQ also occurs in its body (a.k.a., *safe* CQs). A *view* refers to a named query. In this paper, we are restricted to the use of views defined by CQs. We refer to a conjunction of atoms as an *expression*.

A *substitution* θ [7] is a finite set of the form $\{X_1/Y_1, \ldots, X_n/Y_n\}$, where each Y_i is a variable or a constant, and $X_1, \ldots, X_n$ are distinct variables. When $Y_1, \ldots, Y_n$ are distinct variables, θ is called *renaming substitution*. Applying θ on an expression E (resp. a CQ Q), denoted as $\theta(E)$ (resp. $\theta(Q)$), we simultaneously replace each occurrence of X_i in E (resp. Q) by Y_i for all $i = 1, \ldots, n$. The expression $\theta(E)$ (resp. the CQ $\theta(Q)$) is called an *instance* of E (resp. Q). Finally, we say that two CQs Q_1 and Q_2 are *isomorphic* if there is a renaming substitution Q_1 such that $\theta(Q_1)$ and Q_2 are identical.

An expression E is a *generalization* of an expression E' if E' is an instance of E. E is a *common generalization* of $E_1, \ldots, E_n$, with $n > 1$, if E is a generalization of each expression E_i, with $1 \leq i \leq n$. A common generalization E of $E_1, \ldots, E_n$ is a *least common generalization* (or a *least general generalization* - lgg [8]) of $E_1, \ldots, E_n$, with $n > 1$, if there is no other common generalization G of $E_1, \ldots, E_n$ such that E is a generalization of G.

Query equivalence [5] enable comparison between different reformulations of queries. The formal definitions of the query equivalence is given as follows. Let Q_1 and Q_2 be two queries over a database schema $\mathcal{S}$ (under bag-set semantics). We say that Q_2 is *equivalent* in Q_1, denoted $Q_2 \equiv Q_1$, if for every set-valued database instance $\mathcal{D}$ of $\mathcal{S}$, we have that $Q_2(\mathcal{D}) =_b Q_1(\mathcal{D})$.

Given, now, a set of views (also, called *viewset*) $\mathcal{V}$, under bag-set semantics, defined on a database schema $\mathcal{S}$, and a database $\mathcal{D}$ on the schema $\mathcal{S}$, then by $\mathcal{V}(\mathcal{D})$ we denote the database obtained by computing, using bag-operators, all the view relations in $\mathcal{V}$ on $\mathcal{D}$. Considering bag-set semantics, and given a query Q defined on $\mathcal{S}$, we say that a query R is an *equivalent rewriting* (or simply *rewriting*) of Q using the views in $\mathcal{V}$ if all subgoals of R are view atoms (called *view-subgoals* of R) defined in $\mathcal{V}$, and $Q(\mathcal{D}) =_b R(\mathcal{V}(\mathcal{D}))$. By *definition* of a view-subgoal we mean to the definition of the corresponding view in $\mathcal{V}$. Moreover, we say that the *view-expansion* of a view-subgoal v of R is the CQ obtained by applying a substitution θ over the definition V of v such that the head of $\theta(V)$ and the view-subgoal v are identical, and the non-distinguished variables of $\theta(V)$ are fresh variables. The *expansion* of R, denoted as R^{exp}, is a CQ obtained by replacing every view-subgoal of R with the body of its view-expansion. Using the expansion of a rewriting R, we can decide whether or not R is an equivalent rewriting of a CQ Q by checking the equivalence of R^{exp} and Q under the corresponding semantics [1].

3 Useful view for rewriting CQs under bag-set semantics

In this section, a) we show (see Proposition 2) that each view participating in any rewriting can be defined as a generalization of a *duplicate extension* (see Definition 1) of a subexpression of the body of the query and show (see Proposition 3) how the set of the distinguished variables of such views should be selected, and b) we show (see Theorem 1 and Proposition 1) that the view atoms used in a rewriting R should be selected in such a way that the canonical representations (i.e. the rules obtained by removing duplicate subgoals) of Q and R^{exp} are isomorphic.

In [4], the authors give a necessary and sufficient condition for deciding equivalence between CQs under bag-set semantics. Formally, this condition is given by the following theorem (Theorem 7.11 in [4]). In the following, a query (resp. expression) constructed by removing all duplicate atoms from a query Q (resp. an expression E) is called *canonical representation* of Q (resp. E) and is denoted as Q^c (resp. E^c).

Theorem 1. *Let Q_1 and Q_2 be two CQs defined over a database schema $\mathcal{S}$. Then $Q_2 \equiv Q_1$ if and only if Q_1^c and Q_2^c are isomorphic.*

As a consequence of Theorem 1, to test if R is a rewriting of a CQ Q using a viewset $\mathcal{V}$, it is sufficient to test if $(R^{exp})^c$ is isomorphic to Q^c. From this we derive two significant conditions that R^{exp} should satisfy; which are formally given by the following proposition. In the following, we say that a V-subgoal v of a rewriting R of a CQ Q, that uses the view V, *covers* a subgoal g of Q if g is the image of an atom in the view-expansion of v under the corresponding renaming substitution from $(R^{exp})^c$ to Q^c. In this case we say that V covers g.

Proposition 1. *Let Q be a CQ, $\mathcal{V}$ be a viewset and R be a rewriting of Q using $\mathcal{V}$. Then the following hold: (1) there is a subset of view-subgoals of R^c that cover all the subgoals of Q and no view-subgoal in this subset is redundant, and (2) for every view-subgoal v of R, the body of the canonical representation of the view-expansion of v is isomorphic to a subexpression of body(Q^c).*

Intuitively, each rewriting may have a set of view-subgoals that provide the equivalence between its expansion and the query, and several other view-subgoals that provide duplications of the subgoals in its expansion. The second kind of view-subgoals is redundant since none of them affects the existence of the rewriting. Considering now the first condition of the above proposition we conclude that the body of the view-expansion of each view-subgoal of a rewriting

can be constructed by either adding or removing duplicate subgoals to a subexpression of the query. The notion of *duplicate-extension* of a query captures this.

Definition 1. A CQ Q_2 is a *duplicate-extension* of a CQ Q_1 if Q_2 can be obtained by a sequence of additions and deletions of duplicate subgoals from the body of Q_1.

The notion of *duplicate-extension* can be similarly defined for expressions.

Consequently, considering a rewriting R of a CQ Q using V and a subgoal v of R, the body of the view-expansion of v is a duplicate-extension of a subexpression of Q. The following proposition immediately follows from the definition of the view-expansion.

Proposition 2. *If R is a rewriting of a CQ Q using a viewset $\mathcal{V}$, then the body of the definition of each view V of $\mathcal{V}$ used in R is a generalization of a duplicate-extension of a subexpression of Q's body.*

We now define the minimum set of body variables that appear in the head of the definition of a *useful view* (i.e. a view that can be used in a rewriting). This set is based on the set of *linking variables* [2] of a query Q and a subexpression S of Q defined as $lvars(Q, S) = vars(Q - S) \cap vars(S)$.

Proposition 3. *Let Q be a CQ and V be a view. Then, there is a V-subgoal which covers a subexpression S of Q in a rewriting R of Q using a viewset in which V appears,* ***if and only if*** *there is a substitution θ over V such that the following hold: (1) $body(\theta(V)^c)$ is identical to S^c, (2) if X is a variable of V such that $\theta(X)$ belongs to $lvars(Q', S)$, where Q' is obtained from Q by adding a copy of each subgoal g of S covered by another view-subgoal of R, then X is a distinguished variable in the definition of V, (3) if Y and Z are variables of V such that $Y \neq Z$ and $\theta(Y) = \theta(Z)$, then Y and Z are distinguished variables in the definition of V.*

Notice that a view V may cover multiple subexpressions of a CQ Q. In this case, the distinguished variables of V are given by the union of the sets of variables computed by considering that V covers one subexpression at a time.

4 Space of minimal viewsets

In this section we investigate the following problem: Considering a set of conjunctive queries $\mathcal{Q}$ over a relational database schema $\mathcal{S}$, and a database instance $\mathcal{D}$ of $\mathcal{S}$, we want to find a viewset $\mathcal{V}$ over $\mathcal{S}$ such that a) there is a rewriting of each Q in $\mathcal{Q}$ using $\mathcal{V}$, and b) the view set $\mathcal{V}$ is a *minimal viewset* for $\mathcal{Q}$ and $\mathcal{D}$ (i.e. the total number of tuples of $\mathcal{V}(\mathcal{D})$ is minimal). The main result of this section is that we show the decidability of this problem (see Theorem 2) by giving a bounded space of viewsets in which there is at least one minimal viewset.

In Section 3 we showed the form of each useful view. Here, we have a set of CQs and infinite viewsets that give rewritings to all queries in this set (because a duplicate-extension of a subexpression may have arbitrary many subgoals). In addition, if a view covers more than one subexpression (of the same or of different queries) then the body of its definition constitutes a common generalization of duplicate-extensions of the covered subexpressions (Proposition 2). Searching for minimal viewset, it is not useful to generalize the subexpressions more than needed as the overgeneralized view definitions may increase the number of tuples in the instance of viewset. In this perspective, we extend the concept of lgg to duplicate-extensions of expressions.

Definition 2. Let $\mathcal{N}$ be a set of relation names and $\mathcal{E}$ be a set of expressions whose atoms have relation names in $\mathcal{N}$, and such that for each $g \in \mathcal{N}$ there is at least one g-atom (i.e., atom with relation name g) in every expression in $\mathcal{E}$. Suppose $\mathcal{E}'$ be the set of duplicate-extensions of the expressions in $\mathcal{E}$, such that for each $g \in \mathcal{N}$ the number of g-atoms is the same in all expressions of $\mathcal{E}'$. Then we say that an lgg E of the expressions in $\mathcal{E}'$ is an *d-lgg* of the expressions in $\mathcal{E}$ if E does not have duplicate atoms.

Intuitively, the d-lgg is an lgg of duplicate-extensions of a set of expressions without duplicate atoms. Although a duplicate-extension of an expression may have arbitrary many atoms, the d-lggs of a set of expressions have bounded number of atoms.

Proposition 4. *Let $\mathcal{E}$ be a set of n expressions such that there is a d-lgg of $\mathcal{E}$. Then each d-lgg E of the expressions in $\mathcal{E}$, has at most $\Pi_{i=1}^{n}(n_i)$ atoms, where the expression E_i of $\mathcal{E}$ has n_i atoms.*

Now, we show that the space of viewsets which is defined by subexpressions of queries bodies and d-lggs of these subexpressions constitute a bounded search space for the problem of finding a minimal viewset. The following theorem proves this result.

Theorem 2. *Let $\mathcal{Q}$ be a set of CQs over a database schema $\mathcal{S}$ and $\mathcal{D}$ be an instance of $\mathcal{S}$. Then there is a minimal viewset $\mathcal{V}$ for $\mathcal{Q}$ and $\mathcal{D}$ such that for each view V in $\mathcal{V}$ we have that (1) body(V) is either a subexpression S of the body of a CQ in $\mathcal{Q}$ or a d-lgg of subexpressions $S_1, \ldots S_n$ of the bodies of CQs in $\mathcal{Q}$, and (2) considering that V covers either S or $S_1, \ldots S_n$, respectively, head(V) contains the minimum set of variables of the body of V such that the Proposition 3 holds. The problem of finding a minimal viewset for $\mathcal{Q}$ and $\mathcal{D}$ is decidable.*

Future Work: Studying the exact complexity of the problem and especially finding tractable special cases is open. In addition, finding efficient algorithms by further restricting the search space of the problem is also an interesting problem.

References

[1] Foto Afrati, Rada Chirkova, Manolis Gergatsoulis, and Vassia Pavlaki. View selection for real conjunctive queries. *Acta Inf.*, 44(5):289–321, 2007.

[2] Foto N. Afrati, Matthew G. Damigos, and Manolis Gergatsoulis. On solving efficiently the view selection problem under bag-semantics. In *BIRTE '08, LNBIP*, volume 27, pp. 12-28, Springer 2008.

[3] Silvana Castano, Maria Grazia Fugini, Giancarlo Martella, and Pierangela Samarati. *Database Security.* Addison-Wesley & ACM Press, 1995.

[4] Surajit Chaudhuri and Moshe Y. Vardi. Optimization of real conjunctive queries. In *PODS '93, pp. 59-70*, 1993.

[5] Rada Chirkova and Chen Li. Materializing views with minimal size to answer queries. In *PODS '03, pp. 38-48*, 2003.

[6] Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom. *Database Systems: The Complete Book.* Prentice Hall Press, Upper Saddle River, NJ, USA, 2008.

[7] J. W. Lloyd. *Foundations of logic programming.* Springer-Verlag New York, Inc., 1984.

[8] G.D. Plotkin. A note on inductive generalization. *Machine Intelligence*, 5:153–163, 1970.

[9] Dimitri Theodoratos and Timos Sellis. Data warehouse configuration. In *VLDB '97, pp. 126-135*, 1997.