Designing Views to Answer Queries under Set, Bag,and BagSet Semantics

Rada Chirkova[*]
Department of Computer Science, North Carolina State University
Raleigh, NC 27695-7535
chirkova@csc.ncsu.edu

Foto Afrati
Electrical and Computing Engineering
National Technical University
157 73 Athens, Greece
afrati@cs.ece.ntua.gr

Manolis Gergatsoulis
Department of Archive and Library Sciences
Ionian University
Palea Anaktora, Plateia Eleftherias, 49100 Corfu
manolis@ionio.gr

Vassia Pavlaki
Electrical and Computing Engineering
National Technical University
157 73 Athens, Greece
vpavlaki@mail.ntua.gr

ABSTRACT

A lot of work has been done recently on optimizing queries in the presence of materialized views. However the majority of the research assumes set-theoretic semantics while SQL queries have bag-theoretic semantics (duplicates are not eliminated unless explicitly requested). This paper presents results on designing views to answer queries in relational databases under set, bag and bag-set semantics. The results can be used in finding sound and complete algorithms for designing views and rewriting queries under each of the three assumptions.

1. INTRODUCTION

A lot of work has been done recently on optimizing queries in the presence of materialized views. In this context, problems such as definition of views, composition of views, maintenance of views have been researched. However the majority of the research assumes set semantics while SQL queries have bag semantics (duplicates are not eliminated unless explicitly requested). As SQL is the query language used in most commercial database management systems, results on rewriting queries under bag or bag-set semantics are useful in practice.

Database relations are often duplicate-free. However the problem really stems from the bag semantics of the queries and views. More precisely, database relations are often sets, while views and queries are often bags, defined without using the DISTINCT keyword (bag-set semantics).

The bag-set semantics case is arguably more practical than the bag-semantics case, as relational database-management systems typically compute query answers using operators with bag-valued outputs on set-valued databases. At the same time, studying the bag-semantics case is important not just from theoretical but also from practical perspective, as in view selection it is possible to design and materialize bag-valued views and thus to obtain bag-valued databases of stored data. Computing query answers on such databases using the rules of evaluating SQL queries on relational databases obeys the laws of bag, rather than bag-set, semantics.

The current work presents results on designing views to answer queries and rewriting queries in relational databases under set, bag and bag-set semantics which are useful in practice. The work is interesting because it is difficult to apply results on optimization of conjunctive queries to query optimization in real-life database management systems.

To the best of our knowledge, limited related work has been done. [4] studies the containment problem of conjunctive queries under bag semantics which is proved to be $\prod_2^p$-hard, whereas equivalence under bag semantics has the same complexity as the graph - isomorphism problem which is in NP. In [7] techniques for bag seman-

[*] This material is based upon work supported by the National Science Foundation under Grant No. 0307072.

tics, bag-specific constraints (UWDs), and for handling bag queries over arbitrary mixes of bag and set schema elements and views are presented. The problem of optimizing queries with materialized views under bag semantics is studied in [3] and under set semantics in [9]. Finally, [8] studies conjunctive queries over generalized databases in order to achieve an examination of relations as multisets given their importance in SQL.

In this work we study the problem of designing views to answer queries without self-joins under set semantics and queries with and without self-joins under bag and bag-set semantics. The contributions are the following. 1) a bound for the number of subgoals in the optimal viewsets is given and 2) we study the computational complexity of the view selection problem. The results can be used in finding sound and complete algorithms for designing views and rewriting queries under each of the three semantics.

2. PRELIMINARIES

2.1 Basic Definitions

A *relational database* is a collection of stored relations. Each relation R is a collection of tuples, where each tuple is a list of values of the attributes in the *relation schema* of R.

A relation can be viewed either as a *set* or as a *bag* (another term is *multiset*) of tuples. A bag can be thought of as a set of elements (we call it the *core-set* of the bag) with multiplicities attached to each element. In a *set-valued database*, all stored relations are sets; in a *bag-valued database*, multiset stored relations are allowed.

In this paper we focus on safe *conjunctive queries.* A conjunctive query is a rule of the form: Q : $ans(\bar{X}) \leftarrow e_1(\bar{X}_1), \ldots, e_n(\bar{X}_n)$, where $e_1, \ldots e_n$ are database relations. A query has *self-joins* if the minimized query definition [2] has at least two subgoals with the same relation name. A *view* refers to a named query. A view is said to be *materialized* if its answer is stored in the database.

We say that a bag B is a subbag [4] of a bag B' (we write $B \subseteq_b B'$) if each element of B is contained also in B' with a greater than or equal multiplicity. The bag union ($\sqcup$) [4] of two bags is obtained by adding the multiplicity factors for each tuple in either of the bags.

2.2 Query containment and equivalence

A query Q_1 is *set-contained* in a query Q_2, denoted by $Q_1 \sqsubseteq_s Q_2$, if for any database $\mathcal{D}$, the result of the query Q_1 over $\mathcal{D}$ under set semantics is a subset of the result of the query Q_2 over $\mathcal{D}$ under set semantics. A query Q_1 is *bag-contained* (or *bag-set contained*) in a query Q_2, denoted by $Q_1 \sqsubseteq_b Q_2$ (or $Q_1 \sqsubseteq_{bs} Q_2$, respectively), if for any bag-valued (set-valued, respectively) database $\mathcal{D}$, the result of the query Q_1 over $\mathcal{D}$ under bag semantics (bag-set semantics, respectively) is a subbag of the result of the query Q_2 over $\mathcal{D}$ under bag semantics (bag-set semantics, respectively). Two queries are equivalent under set/bag/bag-set semantics if they are contained in each other under the same semantics.

2.3 Equivalent rewritings and the view selection problem

Let $\mathcal{V}$ be a set of views defined on a database schema $\mathcal{S}$, and $\mathcal{D}$ be a database with the schema $\mathcal{S}$. Then by $\mathcal{D}_{\mathcal{V}}$ we denote the database obtained by computing all the view relations in $\mathcal{V}$ on $\mathcal{D}$. Let Q be a query defined on a database schema $\mathcal{S}$, and $\mathcal{V}$ be a set of views defined on $\mathcal{S}$. A query R is a *rewriting of the query Q using the views in $\mathcal{V}$* if all subgoals of R are view predicates defined in $\mathcal{V}$ or interpreted predicates.

The *expansion* of a rewriting R of a query Q on a set of views $\mathcal{V}$, denoted by R^{exp}, is obtained from R by replacing all the view atoms in the body of R by their definitions in terms of the base relations. A rewriting R of a query Q on a set of views $\mathcal{V}$ is a *contained rewriting* of Q using $\mathcal{V}$ under set semantics if $R(\mathcal{D}_{\mathcal{V}})$ is a subset of $Q(\mathcal{D})$. A rewriting R of a query Q on a set of views $\mathcal{V}$ is an *equivalent rewriting* under set semantics if for every database $\mathcal{D}$, $Q(\mathcal{D}) = \mathcal{R}(\mathcal{D}_{\mathcal{V}})$.

The definition of the notion *contained rewriting* for *bag* or *bag-set* semantics is analogous. The only difference is that we now require that $R(\mathcal{D}_{\mathcal{V}})$ is a subbag of $Q(\mathcal{D})$. The definition of the notion of *equivalent rewriting* for the *bag* and *bag-set* semantics is the same as above (in this case, however, the symbol $=$ stands for bag equality). A conjunctive equivalent (under some semantics) rewriting Q' of a conjunctive query Q is *locally minimal* [9] if we cannot remove any literals from Q' and still retain equivalence to Q.

Some results in this paper are given for a special type of constraints $\mathcal{L}$ on materialized views: In those results, $\mathcal{L}$ is a singleton set $\mathcal{L} = \{C\}$, $C \epsilon \mathbf{N}$. The *storage-limit* C means that the total size $size(\mathcal{V}(\mathcal{D}))$ of the relations for the views in $\mathcal{V}$ on $\mathcal{D}$ must not exceed C. If the storage limit is sufficiently large then we can materialize all query answers and this is the optimal viewset. The problem becomes interesting however when the storage limit is less than that. Clearly, if the storage limit is too small then there is no viewset that can rewrite all queries.

In the rest of this papers we consider *problem inputs* that are 4-tuples $(\mathcal{S}, \mathcal{Q}, \mathcal{D}, \mathcal{L})$, where $\mathcal{S}$ is a database schema, $\mathcal{Q}$ is a workload of queries defined on $\mathcal{S}$, $\mathcal{D}$ is a database with schema $\mathcal{S}$ and $\mathcal{L}$ is a collection of constraints on sets of materialized views. A problem input $\mathcal{P}$ = ($\mathcal{S}$, $\mathcal{Q}$, $\mathcal{D}$, $\mathcal{L}$), is said to be *set-oriented* (*bag-oriented*, *bag-set-oriented*, respectively) if we consider set-semantics (bag-semantics, bag-set semantics, respectively) for computing query answers, whereas a problem input is said to be *conjunctive* if we consider the conjunctive language for queries, views and rewritings.

DEFINITION 2.1. *(***Candidate and optimal rewritings***) For a given query Q, semantics (set, bag, or bag-set) for evaluating the query on the database, a viewset $\mathcal{V}$, a database $\mathcal{D}$ and a cost model for query evaluation:*
1) R is a candidate rewriting of Q in terms of $\mathcal{V}$ if R is an equivalent rewriting of Q under the given semantics, and
2) R is an optimal rewriting of Q in terms of $\mathcal{V}$ on a database $\mathcal{D}$ if R is a candidate rewriting and minimizes

the cost of computing the answer to Q on $\mathcal{D}_{\mathcal{V}}$ among all candidate rewritings of Q in terms of $\mathcal{V}$. □

Definition 2.2. *(**Admissible viewset**) Let $P = (\mathcal{S}, \mathcal{Q}, \mathcal{D}, \mathcal{L})$ be a problem input. A set of views $\mathcal{V}$ is said to be an* admissible viewset *for the problem input P, if the following three conditions hold:*
*1) $\mathcal{V}$ gives equivalent (*candidate*) rewritings of all the queries in $\mathcal{Q}$*
2) for every view $V \in \mathcal{V}$, there exists an equivalent rewriting of a query in $\mathcal{Q}$ that uses V
3) $\mathcal{V}$ satisfies the constraints $\mathcal{L}$. □

Definition 2.3. *(**Optimal viewset**) For a problem input $P = (\mathcal{S},\ \mathcal{Q}, \mathcal{D}, \mathcal{L})$, an* optimal viewset *is a set of views $\mathcal{V}$ defined on $\mathcal{S}$, such that:*
1) $\mathcal{V}$ is an admissible viewset *for P, and*
2) $\mathcal{V}$ minimizes the total cost of evaluating the queries in $\mathcal{Q}$ on the database $\mathcal{D}_{\mathcal{V}}$, among all admissible sets of views for P. □

Definition 2.4. *(**Nonredundant viewset**) For any problem input $\mathcal{P}$, a viewset $\mathcal{V}$ is said to be* nonredundant *for $\mathcal{P}$, if $\mathcal{V}$ is admissible for $\mathcal{P}$ and there is no proper subset $\mathcal{V}'$ of $\mathcal{V}$ such that $\mathcal{V}'$ is also admissible for $\mathcal{P}$.* □

In some results of this paper, instead of a database $\mathcal{D}$ in the definition of a problem input, we will use the notion of an *oracle* $\mathcal{O}$. An oracle is supposed to give, instantaneously, exact relation sizes for all views defined on the schema $\mathcal{S}$. In this case a problem input is written as $(\mathcal{S}, \mathcal{Q}, \mathcal{O}, \mathcal{L})$. The notion of an optimal viewset is defined analogously to the case of problem inputs of the form $(\mathcal{S}, \mathcal{Q}, \mathcal{D}, \mathcal{L})$, where $\mathcal{D}$ is a database. The results in the remainder of this paper are given for problem inputs that include a fixed database, but can be extended in a straightforward manner to problem inputs that include an oracle.

2.4 Different Types of Views

There are two types of conjunctive views that can be used in a conjunctive rewriting of a conjunctive query: (1) containment-target views, and (2) filtering views. A conjunctive view V is a *containment-target view* for a conjunctive query Q if there exists a conjunctive rewriting P of Q (P uses V), and there is a containment mapping (for the set-semantics case, or bijective mappings for the bag and bag-set semantics case) from Q to the expansion P^{exp} of P, such that V provides the image of at least one subgoal of Q under the mapping. Intuitively, in a rewriting, a *containment-target view* "covers" at least one query subgoal. Covering all query subgoals is enough to produce a rewriting of the query. A view is a *filtering view* for a query if it is not a containment-target view. For more details see [6].

3. QUERIES WITHOUT SELF-JOINS UNDER SET SEMANTICS

3.1 View definitions without self-joins

In this section, we show that for workloads of queries without self-joins and under set semantics, there exist optimal viewsets whose view definitions do not have self-joins. As a consequence, view definitions in such viewsets have no more subgoals than any query in the workload.

Theorem 3.1. *Given a conjunctive set - oriented problem input $\mathcal{P} = (\mathcal{S}, \mathcal{Q}, \mathcal{D}, \mathcal{L})$, where $\mathcal{L}$ represents a single storage-limit and all queries in $\mathcal{Q}$ are conjunctive queries without self-joins, if there is an optimal viewset $\mathcal{V}$ for $\mathcal{P}$ under the storage limit constraint $\mathcal{L}$, then there exits an optimal viewset $\mathcal{V}'$ under $\mathcal{L}$ such that every view in $\mathcal{V}'$ can be defined as a conjunctive query without self-joins.* □

Corollary 3.1. *Given a conjunctive set - oriented problem input $\mathcal{P} = (\mathcal{S}, \mathcal{Q}, \mathcal{D}, \mathcal{L})$ where $\mathcal{L}$ represents a single storage - limit constraint and assuming that all queries in $\mathcal{Q}$ are without self-joins, and the number of (relational) subgoals in any query does not exceed an integer n, then if there exists an optimal viewset for $\mathcal{P}$ under the storage limit constraint $\mathcal{L}$, then there exists an optimal viewset $\mathcal{V}$ of $\mathcal{P}$ under $\mathcal{L}$, such that for every view V in $\mathcal{V}$, the number of subgoals of V is bounded from above by n.* □

The optimal viewset stipulated in Corollary 3.1 may include filtering views alongside containment - target views. Moreover, even an exponential number of filtering views may be necessary under set semantics; see [5]. Another interesting observation is that we cannot strengthen the Corollary 3.1 to state that under the premises of the corollary there exists an optimal viewset $\mathcal{V}$, in which each view is a subexpression of some query in the input query workload. Finally, when queries have self-joins, the number of subgoals of views can be up to a product of the number of subgoals of the queries. See also Example 1 in [5].

3.2 Rewritings without self-joins

In this section we show that under set semantics and for workloads $\mathcal{Q}$ of queries without self-joins, there exist optimal viewsets $\mathcal{V}$, such that rewriting any query in $\mathcal{Q}$ does not require self-joins of containment-target views in $\mathcal{V}$. We can also show that queries with self-joins may necessitate the use of nontrivial self-joins of containment-target views.

Theorem 3.2. *Given a conjunctive set-oriented problem input $\mathcal{P} = (\mathcal{S}, \mathcal{Q}, \mathcal{D}, \mathcal{L})$ and assuming that the queries do not have self-joins, if there exists an optimal viewset $\mathcal{V}$ for $\mathcal{P}$ under the storage limit constraint $\mathcal{L}$, then it is possible to rewrite each query in the input workload $\mathcal{Q}$ without using self-joins of containment-target views.* □

We can show that the lack of self-joins in workload queries is an essential condition in Theorem 3.2. Example 1 in [5] shows that input queries with self-joins may necessitate the use of nontrivial self-joins of *filtering* views.

3.3 The problem is in NP

The decision version of the view-selection problem is shown in this section to be in NP for workloads of queries without self-joins, provided that filtering views are not used in query rewritings. We prove the result for the *oracle* version of problem inputs, that is, we show that the size of a witness is polynomial in the size of the following components of the problem input: database schema, query workload, and constraints on the materialized views. This result is stronger than proving that the size of a witness is polynomial in the size of the above components plus the size of an input database, because database schemas, query workloads, and constraints on the materialized views are typically small in size compared to the size of possible databases conforming to the schemas. To prove the main result, we first establish an upper bound on the number of containment-target views in query rewritings.

Lemma 3.1. *[9] Let Q be a conjunctive query and $\mathcal{V}$ be a set of views, both Q and $\mathcal{V}$ without built-in predicates. If the body of Q has p relational subgoals and Q' is a locally minimal equivalent conjunctive rewriting of Q using $\mathcal{V}$, then Q' has at most p relational subgoals.* □

Corollary 3.2. *For any conjunctive query Q with p relational subgoals and for any locally minimal conjunctive rewriting Q' of Q in terms of views such that $Q' \equiv_s Q$, the number of containment-target views in Q' does not exceed p.* □

This result follows immediately from the observation that any locally minimal rewriting does not contain filtering views.

Corollary 3.3. *For any set-oriented problem input $\mathcal{P} = (\mathcal{S}, \mathcal{Q}, \mathcal{D}, \mathcal{L})$ with any set of constraints $\mathcal{L}$, and for any optimal nonredundant viewset $\mathcal{V}$ for $\mathcal{P}$, the number of containment-target views in $\mathcal{V}$ does not exceed p where p is the total number of relational subgoals in all the queries in the query workload $\mathcal{Q}$ in $\mathcal{P}$.* □

Theorem 3.3. *Given an* oracle version *of a conjunctive set-oriented problem input $\mathcal{P}$ whose queries are without self-joins, the decision version of the view-selection problem is in NP, provided query rewritings do not include filtering views.* □

Note that if filtering views are allowed in query rewritings, then the view-selection problem under set semantics has an exponential-time lower bound even when none of the workload queries have self-joins; see [5].

4. QUERIES UNDER BAG SEMANTICS

Before proceeding to the main results of this section note that under bag semantics, any candidate query rewriting lacks any filtering views, as well as any redundant containment-target views [4].

4.1 Number of Subgoals is Bounded

In this section, we show that under bag semantics and for workloads of queries without *or with* self-joins, each view definition in any admissible viewset (and thus in any optimal viewset) has no more subgoals than some query in the workload. Furthermore, each view definition in any admissible viewset is a subexpression of some definition in the input query workload. One consequence of these results is that for workloads of queries *without* self-joins, all view definitions in all admissible viewsets do not have self-joins. All the results hold for problem inputs with arbitrary sets of constraints on materialized views.

Lemma 4.1. *Let $\mathcal{P} = (\mathcal{S}, \mathcal{Q}, \mathcal{D}, \mathcal{L})$ be a conjunctive bag-oriented problem input, where $\mathcal{L}$ is a set of* any *constraints and all queries in $\mathcal{Q}$ are conjunctive queries. Let $\mathcal{V}$ be any* admissible *viewset for $\mathcal{P}$, and let Q be any query in $\mathcal{Q}$. Suppose $\mathcal{V}' \subseteq \mathcal{V}$ is the set of all views in the equivalent rewriting R of Q in terms of $\mathcal{V}$. Then the definitions of views $\mathcal{V}'$ in the expansion of R form a partition of the definition of Q.* □

The remaining results in Section 4.1 follow trivially from Lemma 4.1.

Corollary 4.1. *Assume a conjunctive bag-oriented problem input $\mathcal{P}$, and let $\mathcal{V}$ be any admissible viewset for $\mathcal{P}$. Then every view in $\mathcal{V}$ can be defined as a subexpression of some query in the input workload $\mathcal{Q}$.* □

Theorem 4.1. *Given a conjunctive bag-oriented problem input $\mathcal{P}$, the following holds. Suppose n is the number of subgoals in the longest query in $\mathcal{Q}$. Then, for any admissible viewset $\mathcal{V}$ for $\mathcal{P}$, every view in $\mathcal{V}$ can be defined using at most n subgoals.* □

We can make a more precise statement about the number of subgoals in view definitions for views in admissible viewsets:

Corollary 4.2. *Given a conjunctive bag - oriented problem input $\mathcal{P}$ and let V be any view in any* admissible *viewset $\mathcal{V}$ for $\mathcal{P}$. Suppose V is used in rewriting queries $Q_{i_1}, \ldots, Q_{i_k}$ in $\mathcal{Q}$; let m be the number of subgoals in the* shortest *definition among the definitions of $Q_{i_1}, \ldots, Q_{i_k}$. Then V can be defined using at most m subgoals.* □

Corollary 4.3. *Let $\mathcal{P}$ be a conjunctive bag-oriented problem input and $\mathcal{V}$ an admissible viewset for $\mathcal{P}$. Assuming that queries in $\mathcal{Q}$ do not have self-joins, then every view in $\mathcal{V}$ can be defined as a conjunctive query without self-joins.* □

Note that given a problem input $\mathcal{P}$ whose query workloads $\mathcal{Q}$ do not have self-joins, and for any admissible viewset $\mathcal{V}$ for $\mathcal{P}$, rewriting any query in $\mathcal{Q}$ does not require self-joins of views in $\mathcal{V}$.

4.2 The Problem is in NP

In this section we show that the decision version of the view-selection problem is in NP for a single storage-limit constraint on materialized views (see also [1]). Similarly to Section 3, we prove the result for the *oracle* version of problem inputs. At the same time, unlike the results in Section 3, the NP results for bag semantics hold for workloads of queries without *or with* self-joins.

We first establish an analog of Corollary 3.2 in Section 3:

Corollary 4.4. *For any conjunctive query Q with p relational subgoals and for any conjunctive rewriting Q' of Q in terms of views, such that $Q' \equiv_b Q$, the number of views in Q' does not exceed p.* □

This result follows immediately from the fact that for any rewriting that is equivalent to a query under bag semantics, the rewriting does not contain filtering views or "unnecessary" containment - target views, and is thus locally minimal.

Corollary 4.5. *Under bag semantics, for any problem input $\mathcal{P} = (\mathcal{S}, \mathcal{Q}, \mathcal{D}, \mathcal{L})$ with any set of constraints $\mathcal{L}$, and for any admissible viewset $\mathcal{V}$ for $\mathcal{P}$, the number of views in $\mathcal{V}$ does not exceed p, where p is the total number of relational subgoals in all the queries in the query workload $\mathcal{Q}$ in $\mathcal{P}$.* □

Theorem 4.2. *Given an* oracle version *of a conjunctive, bag-oriented problem input $\mathcal{P}$ and assuming additionally that the input set of constraints $\mathcal{L}$ represents a single storage-limit, the decision version of the view-selection problem is in NP.* □

Unlike Theorem 3.3, Theorem 4.2 holds for workloads of queries without *or with* self-joins.

5. QUERIES UNDER BAG-SET SEMANTICS

Before proceeding to the main results of this section note that filtering views are not needed under bag-set semantics. The proof is left to the extended version of the current paper due to space limit. In bag-semantics case also filtering views can be used, but removing them generally results in more efficient query evaluation.

5.1 Number of Subgoals is Bounded

In the following paragraphs, we show that under bag-set semantics and for workloads of queries without *or with* self-joins, each view definition in any admissible viewset (and thus in any optimal viewset) can be defined using no more subgoals than for some query in the workload. Furthermore, each view definition in any admissible viewset can be defined as a subexpression of some definition in the input query workload. One consequence of these results is that for workloads of queries *without* self-joins, all view definitions in all admissible viewsets can be defined without using self-joins. All the results hold for problem inputs with arbitrary sets of constraints on materialized views.

Lemma 5.1. *Given a conjunctive, bag - set oriented problem input, let $\mathcal{V}$ be any* admissible *viewset for $\mathcal{P}$. Then each view in $\mathcal{V}$ can be defined as a subexpression of some query in $\mathcal{Q}$.* □

Note that in Lemma 5.1, there is no requirement on left-linear query plans or on local minimality of rewritings. In addition, all results in Section 5.1 hold for optimal viewsets in particular and follow trivially from Lemma 5.1.

Corollary 5.1. *Given a conjunctive bag-set-oriented problem input $\mathcal{P}$, let $\mathcal{V}$ be any admissible viewset for $\mathcal{P}$. Then every view in $\mathcal{V}$ can be defined as a subexpression of some query in the input workload $\mathcal{Q}$.* □

Theorem 5.1. *Given a conjunctive bag-set-oriented problem input $\mathcal{P}$, the following holds. Suppose n is the number of subgoals in the longest query in $\mathcal{Q}$. Then, for any admissible viewset $\mathcal{V}$ for $\mathcal{P}$, every view in $\mathcal{V}$ can be defined using at most n subgoals.* □

We can make a more precise statement about the number of subgoals in view definitions for views in admissible viewsets:

Corollary 5.2. *Given a conjunctive bag-set-oriented problem input $\mathcal{P}$, let V be any view in any* admissible *viewset $\mathcal{V}$ for $\mathcal{P}$. Suppose V is used in rewriting queries $Q_{i_1}, \ldots, Q_{i_k}$ in $\mathcal{Q}$; let m be the number of subgoals in the* shortest *definition among the definitions of $Q_{i_1}, \ldots, Q_{i_k}$. Then V can be defined using at most m subgoals.* □

Corollary 5.3. *Given a conjunctive bag-set - oriented problem input $\mathcal{P}$, and assuming additionally that queries in $\mathcal{Q}$ do not have self-joins, let $\mathcal{V}$ be any admissible viewset for $\mathcal{P}$. Then every view in $\mathcal{V}$ can be defined as a conjunctive query without self-joins.* □

It is interesting to note that for problem inputs $\mathcal{P}$ whose query workloads $\mathcal{Q}$ do not have self-joins and for any admissible viewset $\mathcal{V}$ for $\mathcal{P}$, rewriting any query in $\mathcal{Q}$ does not require self-joins of views in $\mathcal{V}$.

5.2 The Problem is in NP

We now show that the decision version of the view-selection problem is in NP for a single storage-limit constraint on materialized views. Similarly to the previous sections, we prove the result for the *oracle* version of problem inputs. The NP results hold for workloads of queries without *or with* self-joins.

Corollary 5.4. *For any conjunctive query Q with p relational subgoals and for any conjunctive locally minimal rewriting Q' of Q in terms of views, such that $Q' \equiv_{bs} Q$, the number of views in Q' does not exceed p.* □

This result follows immediately from the definition of a locally minimal rewriting that is equivalent to a query under bag-set semantics. By definition, the rewriting does not contain filtering views or "unnecessary" containment - target views.

Corollary 5.5. *Under bag-set semantics, for any problem input $\mathcal{P} = (\mathcal{S}, \mathcal{Q}, \mathcal{D}, \mathcal{L})$ with any set of constraints $\mathcal{L}$, and for any nonredundant viewset $\mathcal{V}$ for $\mathcal{P}$, the number of views in $\mathcal{V}$ does not exceed p, where p is the total number of relational subgoals in all the queries in the query workload $\mathcal{Q}$ in $\mathcal{P}$.* □

Theorem 5.2. *Given an* oracle version *of a conjunctive bag-set-oriented problem input $\mathcal{P}$, and assuming additionally that the input set of constraints $\mathcal{L}$ represents* *a single storage-limit, the decision version of the view-selection problem is in NP. (Similarly to Theorem 4.2, Theorem 5.2 holds for workloads of queries without* or with *self-joins):* □

6. CONCLUSIONS AND FUTURE WORK

This paper presented results on designing views to answer queries in relational databases under set, bag and bag-set semantics. Some complexity results were also given. On the practical side, we are currently working on designing sound and complete algorithms for designing views under each of the three assumptions. In future work we would like to extend this method to include, in a systematic way, queries with arithmetic comparisons. On a different direction, applying our results to XQuery is also the object of future research.

7. REFERENCES

[1] F. Afrati and R. Chirkova. Selecting and using views to compute aggregate queries. Available at `http://dbgroup.ncsu.edu/aggregAquv.pdf`, 2003.

[2] A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational databases. In *Proc. 9th ACM Symposium on Theory of Computing*, pages 77–90, 1977.

[3] S. Chaudhuri, R. Krishnamurty, S. Potamianos, and K. Shim. Optimizing queries with materialized views. Proceedings of ICDE, 1995.

[4] S. Chaudhuri and M. Y. Vardi. Optimization of real conjunctive queries. In *Proc. 12th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 59–70. ACM Press, 1993.

[5] R. Chirkova, A. Y. Halevy, and D. Suciu. A formal perspective on the view selection problem. *The VLDB Journal*, 11(3):216–237, 2002.

[6] R. Chirkova and C. Li. Materializing views with minimal size to answer queries. In *Proc. 22th ACM SIGACT-SIGMOD-GIGART Symposium on Principles of Database Systems*, pages 38–48. ACM Press, 2003.

[7] A. Deutsch. *XML Query Reformulation over Mixed and Redundant Storage*. PhD thesis, University of Pennsylvania, 2002. Available at `http://www.db.ucsd.edu/People/alin/thesis/thesis.pdf`.

[8] Y. Ioannidis and R. Ramakrishnan. Containment of conjunctive queries: Beyond relations as sets. *ACM Transactions on Database Systems*, 20(3):288–324, 1995.

[9] A. Y. Levy, A. O. Mendelzon, Y. Sagiv, and D. Srivastava. Answering queries using views. In *Proc. 14th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 95–104. ACM Press, 1995.