On the Decidability and Finite Controllability of Query Processing in Databases with Incomplete Information

Riccardo Rosati
Dipartimento di Informatica e Sistemistica
Università di Roma “La Sapienza”
Via Salaria 113, I-00198 Roma, Italy
rosati@dis.uniroma1.it

ABSTRACT

In this paper we study queries over relational databases with integrity constraints (ICs). The main problem we analyze is *OWA query answering*, i.e., query answering over a database with ICs under open-world assumption. The kinds of ICs that we consider are functional dependencies (in particular key dependencies) and inclusion dependencies; the query languages we consider are conjunctive queries (CQs), union of conjunctive queries (UCQs), CQs and UCQs with negation and/or inequality. We present a set of results about the decidability and finite controllability of OWA query answering under ICs. In particular: (i) we identify the decidability/undecidability frontier for OWA query answering under different combinations of the ICs allowed and the query language allowed; (ii) we study OWA query answering both over finite databases and over unrestricted databases, and identify the cases in which such a problem is *finitely controllable*, i.e., when OWA query answering over finite databases coincides with OWA query answering over unrestricted databases. Moreover, we are able to easily turn the above results into new results about implication of ICs and query containment under ICs, due to the deep relationship between OWA query answering and these two classical problems in database theory. In particular, we close two long-standing open problems in query containment, since we prove finite controllability of containment of conjunctive queries both under arbitrary inclusion dependencies and under key and foreign key dependencies. Besides their theoretical interest, we believe that the results of our investigation are very relevant in many research areas which have recently dealt with databases under an incomplete information assumption: e.g., view-based information access, ontology-based information systems, data integration, data exchange, and peer-to-peer information systems.

1. INTRODUCTION

The problem In this paper we study queries and integrity constraints (ICs) over relational databases. The main problem studied in this paper is *OWA query answering*, which corresponds to query answering over a database with integrity constraints under open-world assumption (OWA), i.e., under the assumption that the facts stored in the database are only an *incomplete* specification of the data [35, 21, 34]. Under this assumption, the actual meaning of a database $\mathcal{D}$ with integrity constraints $\mathcal{C}$ is represented by the set of all databases $\mathcal{B}$ such that $\mathcal{B}$ contains all the facts in $\mathcal{D}$ and $\mathcal{B}$ satisfies the integrity constraints in $\mathcal{C}$.

The significance of the OWA query answering problem is witnessed by the following considerations:

- As we will explain in Section 3, OWA query answering is deeply related to several classical problems in database theory, in particular: *implication of integrity constraints* (a.k.a. database dependencies) [7]; *OWA-consistency*, i.e., consistency of a database instance with respect to a set of ICs under open-world assumption [35, 21]; and *query containment under integrity constraints* [26].
- Many research areas are currently studying problems that involve databases with incomplete information: e.g., view-based information access [23], ontology-based information systems [25], data integration [28], data exchange [17], mapping composition [18], consistent query answering [3], and peer-to-peer information systems [22]. In all such scenarios, the problem of OWA query answering (or problems very closely related to it) is studied under various forms. Therefore, results about OWA query answering are in principle very relevant in all these areas.

We recall that, even in the absence of integrity constraints, the problem of OWA-answering first-order queries is undecidable both over finite and over unrestricted databases [31], while, under the standard closed-world assumption (CWA) adopted by databases, any (domain-independent) first-order query can be answered in polynomial time with respect to data complexity, even in the presence of the typical relational integrity constraints [2]. That is, OWA query answering is generally much harder than CWA query answering.

From queries to ICs We consider the most common form of relational queries, i.e., *conjunctive queries* (CQs), and the most important forms of relational integrity constraints, i.e.: (i) *functional dependencies* (FDs), and in particular *key dependencies* (KDs); (ii) *inclusion dependencies* (IDs); (iii) *exclusion dependencies* (EDs).

PODS'06, June 26–28, 2006, Chicago, Illinois, USA.

We exploit the tight correspondence between the notion of conjunctive query and each of the above forms of integrity constraint. Informally, the correspondence between a class of queries $\mathcal{Q}$ and a class of integrity constraints $\mathcal{C}$ is based on the fact that (boolean) queries in $\mathcal{Q}$ are able to express the negation of each integrity constraint in $\mathcal{C}$, i.e., a database violates an integrity constraint $c \in \mathcal{C}$ if and only if the corresponding query $q_c \in \mathcal{Q}$ is true in the database. More specifically, we establish the following correspondences:

- conjunctive queries can be put in correspondence with exclusion dependencies;
- conjunctive queries with inequalities can be put in correspondence with functional and key dependencies;
- conjunctive queries with negation can be put in correspondence with inclusion dependencies and tuple-generating dependencies (TGDs).

Of course, the above relationship between queries and ICs is not a result *per se*, since it is not surprising and is not totally new: indeed, some of the above correspondences have been used, in different forms, in previous studies concerning query containment and query answering under ICs (among the most recent examples, see e.g. [24, 19, 16]).

However, through the systematic use of such correspondences, we are actually able to "jump" from the notion of integrity constraint to the notion of query, and we are able to provide a unified view of the problems mentioned above, i.e., implication of integrity constraints, OWA-consistency, OWA query answering, and query containment under ICs. Moreover, this allows us, for instance, to use a number of known results about implication of ICs to prove properties for OWA query answering and query containment under ICs.

Results for OWA query answering We develop our analysis both under the assumption that a database must be a *finite* structure, and under the assumption of *unrestricted* databases (i.e., a database may be infinite). In this respect, we identify the cases when the OWA query answering problem is *finitely controllable*, i.e., when OWA query answering over finite databases coincides with OWA query answering over unrestricted databases.

We present a set of decidability and finite controllability results for OWA query answering under ICs. More precisely: (i) we identify the cases in which such a problem is finitely controllable; (ii) we identify the decidability/undecidability frontier for the query languages and the ICs above mentioned; (iii) for the decidable cases, we establish the computational complexity of OWA query answering (both data complexity and combined complexity). The summary of the results obtained is reported in Figure 1 (see Section 9 for an explanation of the table).

In a nutshell, our results provide a clear picture of the frontier between decidability and undecidability of OWA-answering of conjunctive queries (and unions of conjunctive queries) under key dependencies and inclusion dependencies. In particular, our results show that:

- OWA-answering of (unions of) conjunctive queries under IDs (Theorems 3 and 4) and under keys and foreign keys (Theorem 5) is finitely controllable;
- OWA-answering of conjunctive queries under so-called *non-conflicting* KDs and IDs is not finitely controllable and is undecidable over finite databases (Theorems 6 and 7);
- adding (even safe) negation to unions of conjunctive queries makes OWA-answering under inclusion dependencies non-finitely controllable (Theorem 9) and undecidable (Theorem 10);
- adding (even safe) negation to conjunctive queries under combinations of KDs and IDs in which OWA-answering of conjunctive queries is decidable, makes OWA-answering non-finitely controllable and undecidable (Theorem 11);
- adding a form of negation with universal quantification to unions of conjunctive queries makes OWA-answering non-finitely controllable and undecidable *even in the absence of ICs* (Theorem 13). Therefore, even for very small fragments of first-order queries with unsafe negation, and even in the absence of ICs, OWA-answering is not finitely controllable and is undecidable.
- adding inequalities to conjunctive queries also makes OWA-answering non-finitely controllable and undecidable in the presence of IDs (Theorem 8).

Results for query containment Moreover, we are able to easily turn the above results into new results about implication of ICs and query containment under ICs, due to the deep relationship between OWA query answering and these two classical problems in database theory. In particular, we close two long-standing open problems in query containment which date back to the mid-80's [26]:

- we prove finite controllability of containment of conjunctive queries under arbitrary inclusion dependencies (Corollary 1);
- we prove finite controllability of containment of conjunctive queries under keys and foreign keys (Corollary 2).

Relevance of our results Besides its theoretical interest, we believe that the analysis presented in this paper is very relevant in all the above mentioned areas dealing with data under an incomplete information assumption.

In general, our results show that it is very easy to get to undecidability of query processing in databases under incomplete information, as long as the IC/schema language and/or the query language are sufficiently expressive. Consequently, such results provide a set of coordinates which may help in the process of choosing a reasonable tradeoff between expressiveness of both schema and query languages, and decidability/complexity of query processing.

Structure of the paper In the next section, we present some preliminary definitions and define the problem studied. In Section 3 we illustrate the relationship between queries and integrity constraints which drives our analysis of OWA query answering. Then, in Section 4 we present our results for OWA-answering of CQs and UCQs, in Section 5 we analyze OWA-answering for UCQs with inequalities, and in Section 6 we study UCQs with negation. In Section 7 we describe the relationship between OWA-answering and query containment, and point out two notable consequences of our results for OWA-answering in query containment. Finally, we analyze related work in Section 8 and conclude in Section 9.

2. DEFINITIONS

We start from: (i) a relational signature, i.e., a set of relation symbols in which each relation is associated with an arity, i.e., a non-negative integer; (ii) a countably infinite alphabet of constant symbols; (iii) an alphabet of variable symbols. An *attribute* of a relation r is an integer b such that $1 \leq b \leq n$, where n is the arity of r.

A *fact* is an expression of the form $r(t)$, where r is a relation symbol and t is a tuple of constants. An *atom* is an expression of the form $r(t)$, where r is a relation symbol and t is a tuple of constants or variables. A *substitution* is a function mapping variables to constants.

A *database instance* (or simply *database*) is a set of facts.

Given an n-tuple $t = \langle v_1, \ldots, v_n \rangle$ and a sequence of integers $A = i_1, \ldots, i_k$ where each $1 \leq i_j \leq n$ for each j, we denote by $t[A]$ the projection of t over A, i.e., the k-tuple $\langle v_{i_1}, \ldots, v_{i_k} \rangle$.

2.1 Integrity constraints

Key dependencies A key dependency (KD) is an expression of the form $key(r) = A$, where r is a relation symbol and A is a non-empty sequence of attributes, i.e., a sequence of integers ranging from 1 to the arity of r. The number of attributes in A is called the *arity* of the KD. We say that a set of KDs $\mathcal{K}$ is a set of *single* KDs if, for each relation r, there is at most one KD for r in $\mathcal{K}$. (We can assume that a KD for a relation is always present, since in the case when there are no KDs for r we can consider the trivial KD $key(r) = U$ where U is the set of all attributes of r.) A database $\mathcal{D}$ satisfies a KD $key(r) = A$ if, for every pair of facts of the form $r(t)$, $r(t')$ in $\mathcal{D}$, if $t[A] = t'[A]$ then $t = t'$.

Inclusion dependencies An inclusion dependency (ID) is an expression of the form $r[A] \subseteq s[B]$, where r and s are relation symbols and A and B are sequences of attributes, i.e., sequences of integers ranging from 1 to the arity of the respective relations. The number of attributes in A (which is the same as the number of attributes in B) is called the *arity* of the ID. We do not allow multiple occurrences of the same attribute in A and in B. A database $\mathcal{D}$ satisfies an ID of the form $r[A] \subseteq s[B]$ if, for every fact in $\mathcal{D}$ of the form $r(t)$, there exists a fact in $\mathcal{D}$ of the form $s(t')$ such that $t[A] = t'[B]$.

Other dependencies KDs and IDs are the main ICs studied in this paper. However, we also introduce functional dependencies, single-head tuple-generating dependencies, and exclusion dependencies, which will be used in the following. A *functional dependency* (FD) is an expression of the form $r : A \rightarrow b$ where r is a relation, A is a set of attributes of r and b is an attribute of r. A database $\mathcal{D}$ satisfies a FD $r : A \rightarrow b$ if, for every pair of facts of the form $r(t)$, $r(t')$ in $\mathcal{D}$, if $t[A] = t'[A]$ then $t[b] = t'[b]$.

An *exclusion dependency* (ED) is an expression of the form $\forall \vec{x}.g_1, \ldots, g_n \rightarrow \bot$, where g_i is an atom. A database $\mathcal{D}$ satisfies such ED if there exists no substitution σ of the variables $\vec{x}$ such that $\{\sigma(g_1), \ldots, \sigma(g_n)\} \subseteq \mathcal{D}$.

We then recall *single-head tuple-generating dependencies*, also known as *template dependencies* [6, 35], a generalization of IDs. A single-head tuple-generating dependency (STGD) is an expression of the form $\forall \vec{x}.r_1(\vec{x}), \ldots, r_n(\vec{x}) \rightarrow \exists \vec{y}.r(\vec{x}, \vec{y})$. A database $\mathcal{D}$ satisfies an STGD of the above form if, for every substitution σ of the variables $\vec{x}$ such that $\{\sigma(r_1(\vec{x})), \ldots, \sigma(r_n(\vec{x}))\} \subseteq \mathcal{D}$, there exists a substitution σ' of $\vec{y}$ such that $r(\sigma(\vec{x}), \sigma'(\vec{y})) \in \mathcal{D}$. An STGD is called *safe* if there are no existential variables in the right-hand side of the implication, i.e., if it is of the form $\forall \vec{x}.r_1(\vec{x}), \ldots, r_n(\vec{x}) \rightarrow r(\vec{x})$.

Classes of KDs and IDs Given a set of KDs $\mathcal{K}$ and a set of IDs $\mathcal{I}$, we say that $\mathcal{I} \cup \mathcal{K}$ is a set of *non-conflicting* KDs and IDs if $\mathcal{K}$ is a set of single KDs and for each ID in I of the form $r[A] \subseteq s[B]$, B is not a *proper* superset of the key of s, i.e., if $\mathcal{K}$ contains the KD $key(s) = C$, then $B \not\supset C$.

Moreover, given a set of single KDs $\mathcal{K}$ and a set of IDs $\mathcal{I}$, we say that $\mathcal{I}$ is a set of *foreign keys* (FKs) for $\mathcal{K}$ if for each ID in $\mathcal{I}$ of the form $r[A] \subseteq s[B]$, B is a (not necessarily strict) subset of the key of s, i.e., if $\mathcal{K}$ contains the KD $key(s) = C$, then $B \subseteq C$.[1] Of course, if $\mathcal{I}$ is a set of FKs for $\mathcal{K}$ then $\mathcal{I} \cup \mathcal{K}$ is a set of non-conflicting KDs and IDs, but not vice versa.

Finally, given a set of ICs $\mathcal{C}$, we say that a database instance $\mathcal{B}$ satisfies $\mathcal{C}$ if $\mathcal{B}$ satisfies every IC in $\mathcal{C}$.

2.2 Queries

A *union of conjunctive queries* (UCQ) is an expression of the form

$$\{\vec{x} \mid conj_1(\vec{x}, \vec{c}) \vee \ldots \vee conj_m(\vec{x}, \vec{c})\} \qquad (1)$$

where each $conj_i(\vec{x}, \vec{c})$ is an expression of the form

$$conj_i(\vec{x}, \vec{c}) = \exists \vec{y}.a_1 \wedge \ldots \wedge a_n$$

in which each a_i is an atom whose arguments are terms from the sets of variables $\vec{x}$, $\vec{y}$, and from the set of constants $\vec{c}$ and such that each variable from $\vec{x}$ and $\vec{y}$ occurs in at least one atom a_i. The variables $\vec{x}$ are called the distinguished variables of the CQ.

A UCQ with negation (UCQ$^{\neg}$) is an expression of the form (1) in which each a_i is either an atom or a negated atom, and a negated atom is an expression of the form $\neg a$ where a is an atom. A UCQ with safe negation (UCQ$^{\neg s}$) is a UCQ$^{\neg}$ of the form (1) and such that in each $conj_i(\vec{x}, \vec{c})$ each variable from $\vec{x}$ and $\vec{y}$ occurs in at least one positive atom.

A UCQ with inequalities (UCQ$^{\neq}$) is an expression of the form (1) in which each $conj_i(\vec{x}, \vec{c})$ is a conjunction $\exists \vec{y}.a_1 \wedge \ldots \wedge a_n$ where each a_i is either an atom or an expression of the form $z \neq z'$, where z and z' are variables.

A UCQ with universally quantified negation (UCQ$^{\neg\forall}$) is a UCQ$^{\neg}$ in which the variables that only appear in negated atoms are universally quantified. Formally, a UCQ$^{\neg\forall}$ is an expression of the form (1) in which each $conj_i(\vec{x}, \vec{c})$ is of the form

$$\exists \vec{y}.\forall \vec{z}.conj(\vec{x}, \vec{y}, \vec{z}, \vec{c})$$

where $conj$ is a conjunction of literals (atoms and negated atoms) whose arguments are terms from the sets of variables $\vec{x}$, $\vec{y}$, $\vec{z}$ and from the set of constants $\vec{c}$, and in which *each variable in $\vec{z}$ only occurs in negated atoms.* An example of a UCQ$^{\neg\forall}$ is the following:

$$\{x \mid (\exists y, z.\forall v, w.r(x, y, z) \wedge \neg s(x, w, y) \wedge \neg t(v, w, x)) \vee (\exists y.\forall u.r(x, x, y) \wedge \neg s(x, u, u))\}$$

[1] Notice that this definition of foreign key generalizes the more common assumption in which a foreign key refers exactly to a key, which corresponds to $B = C$ (or even $B \supseteq C$) in the above definition.

Notice that all the classes of queries above considered are *domain-independent* first-order queries [2].

We call a UCQ a *conjunctive query* (CQ) when $m = 1$. Analogously, we define the notions of CQ with negation (CQ$^{\neg}$), safe negation (CQ$^{\neg s}$), inequalities (CQ$^{\neq}$), and universally quantified negation (CQ$^{\neg\forall}$).

A *boolean* CQ is a CQ without distinguished variables, i.e., an expression of the form $conj_1(\vec{x}, \vec{c}) \vee \ldots \vee conj_m(\vec{x}, \vec{c})$. Being a sentence, i.e., a closed first-order formula, such a query is either true or false in a database. In the same way, we define the boolean version of the other kinds of queries introduced above. Given a non-boolean query q and a tuple of constants t, we denote by $q(t)$ the boolean query obtained from q by replacing the distinguished variables of q with the corresponding constants in t. Finally, the *size* of a CQ q is the number of atoms in the body of q.

2.3 Problems studied

Given a query q and a database $\mathcal{D}$, we denote by $q^{\mathcal{D}}$ the set of tuples corresponding to the standard evaluation of the query over $\mathcal{D}$ (under CWA). Moreover, given a set of ICs $\mathcal{C}$, we denote by $sem(\mathcal{C}, \mathcal{D})$ the set of databases $sem(\mathcal{C}, \mathcal{D}) = \{\mathcal{B} \mid \mathcal{B} \supseteq \mathcal{D}$ and $\mathcal{B}$ satisfies $\mathcal{C}\}$, while $sem_f(\mathcal{C}, \mathcal{D})$ denotes the subset of *finite* databases contained in $sem(\mathcal{C}, \mathcal{D})$.

Then, we define $ans(q, \mathcal{C}, \mathcal{D})$ and $ans_f(q, \mathcal{C}, \mathcal{D})$ as follows:

$$ans(q, \mathcal{C}, \mathcal{D}) = \{t \mid t \in q^{\mathcal{B}} \text{ for every } \mathcal{B} \in sem(\mathcal{C}, \mathcal{D})\}$$
$$ans_f(q, \mathcal{C}, \mathcal{D}) = \{t \mid t \in q^{\mathcal{B}} \text{ for every } \mathcal{B} \in sem_f(\mathcal{C}, \mathcal{D})\}$$

The above definition of $ans(q, \mathcal{C}, \mathcal{D})$ corresponds to the notion of *certain answers* in indefinite databases.

We now introduce the main problems studied in the paper, i.e., implication of ICs [2], OWA-consistency and OWA-answering [36, 21].

Implication of ICs For unrestricted databases: given a set of ICs $\mathcal{C}$ and an IC I, we say that $\mathcal{C}$ implies I (and write $\mathcal{C} \models I$) if for every database $\mathcal{D}$ such that $\mathcal{D}$ satisfies $\mathcal{C}$, $\mathcal{D}$ satisfies I. For finite databases: $\mathcal{C}$ finitely implies I (and write $\mathcal{C} \models_f I$) if for every finite database $\mathcal{D}$ such that $\mathcal{D}$ satisfies $\mathcal{C}$, $\mathcal{D}$ satisfies I.

OWA-consistency For unrestricted databases: given a set of ICs $\mathcal{C}$ and a database $\mathcal{D}$, we say that $\mathcal{D}$ is OWA-consistent with $\mathcal{C}$ if $sem(\mathcal{C}, \mathcal{D}) \neq \emptyset$. For finite databases: $\mathcal{D}$ is OWA$_f$-consistent with $\mathcal{C}$ if $sem_f(\mathcal{C}, \mathcal{D}) \neq \emptyset$.

OWA-answering For unrestricted databases: given a set of ICs $\mathcal{C}$, a database $\mathcal{D}$, and a query q, compute $ans(q, \mathcal{C}, \mathcal{D})$. For finite databases (finite OWA-answering): compute $ans_f(q, \mathcal{C}, \mathcal{D})$.

The *decision problem* associated with OWA-answering is the following: given a query q and a tuple t, decide whether $t \in ans(q, \mathcal{C}, \mathcal{D})$, i.e., decide whether the boolean query $q(t)$ is true in all databases in $sem(\mathcal{C}, \mathcal{D})$ (resp., $sem_f(\mathcal{C}, \mathcal{D})$). In the following, when we talk about (un)decidability of OWA-answering we actually refer to (un)decidability of the decision problem associated with OWA-answering.

3. FROM INTEGRITY CONSTRAINTS TO QUERIES

In this section we prove some preliminary results that highlight the correspondences among implication of ICs, OWA-consistency, and OWA-answering. We start by showing a property that relates undecidability of IC implication to OWA-consistency.

Theorem 1. *Let $\mathcal{C}$ be a set of ICs, and let φ be either an ED or a FD. If the problem of deciding whether $\mathcal{C} \models \varphi$ (respectively, $\mathcal{C} \models_f \varphi$) is undecidable, then OWA-consistency (respectively, OWA$_f$-consistency) under $\mathcal{C}$ is undecidable.*

Proof. We reduce the problem of deciding whether $\mathcal{C} \models \varphi$ to OWA-consistency. More precisely, starting from the IC φ, we construct an instance $\mathcal{D}_\varphi$ as follows:

1. if φ is an ED of the form $\forall \vec{x}. g_1, \ldots, g_n \rightarrow \bot$, we first "freeze" φ, i.e., we define a substitution σ of the variables $\vec{x}$ appearing in φ with constant symbols, such that each distinct variable is replaced by a distinct constant, and define $\mathcal{D}_\varphi = \{\sigma(g_1), \ldots, \sigma(g_n)\}$;
2. if φ is a FD of the form $r : 1, \ldots, k \rightarrow i$, again we "freeze" φ as above, and obtain $\mathcal{D}_\varphi = \{r(c_1, \ldots, c_n), r(c_1, \ldots, c_k, c'_{k+1}, \ldots, c'_n)\}$ where each distinct c_i is a distinct constant symbol.

We now prove that $\mathcal{C} \not\models \varphi$ iff $\mathcal{D}_\varphi$ is OWA-consistent with $\mathcal{C}$. Indeed, if $\mathcal{D}_\varphi$ is OWA-consistent with $\mathcal{C}$, then there exists a database $\mathcal{B}$ containing $\mathcal{D}_\varphi$ and satisfying $\mathcal{C}$: therefore, in both the above cases φ is not satisfied in such a $\mathcal{B}$, which in turn implies that $\mathcal{C} \not\models \varphi$. Conversely, if $\mathcal{C} \not\models \varphi$, then there exists a database $\mathcal{B}$ that satisfies $\mathcal{C}$ ando does not satisfy φ. In the case when φ is an ED (the case when φ is a FD is analogous), this implies that there exists a substitution σ of the variables in $g_1, \ldots, g_n$ such that there exists a set of facts $S = \{g'_1, \ldots, g'_n\}$ contained in $\mathcal{B}$ such that $\sigma(g_i) = g'_i$ for each i. This immediately implies that $\mathcal{D}_\varphi$ is OWA-consistent with $\mathcal{C}$.

The proof for the case of finite databases is analogous. □

All the ICs presented in this paper and studied in the relational setting belong to two well-known general classes of ICs, called tuple-generating depndencies (TGDs) and equality-generating dependencies (EGDs). Such kinds of ICs correspond to sentences (in particular, implications) in first-order logic. Consequently, the negation of an IC corresponds to a sentence, i.e., a boolean first-order query.

Interestingly, it turns out that some of the ICs above presented are such that their negation corresponds to a boolean conjunctive query of the kinds introduced above. In particular, it is immediate to verify that:

- if φ is an ED, the negation of φ corresponds to a CQ. In particular, if φ is the ED $\forall \vec{x}. a_1 \wedge \ldots \wedge a_n \rightarrow \bot$, we denote by $\tau_q(\varphi)$ the boolean CQ $\exists \vec{x}. a_1 \wedge \ldots \wedge a_n$;
- if φ is a FD, the negation of φ corresponds to a CQ$^{\neq}$. In particular, if φ is the FD $r : 1, \ldots, k \rightarrow i$ (where r has arity n), we denote by $\tau_q(\varphi)$ the boolean CQ$^{\neq}$ $\exists x_1, \ldots, x_n. r(x_1, \ldots, x_n) \wedge r(x_1, \ldots, x_k, x'_{k+1}, \ldots, x'_n) \wedge x_i \neq x'_i$;
- if φ is a STGD, the negation of φ corresponds to a CQ$^{\neg\forall}$. In particular, if φ is the STGD $\forall \vec{x}. a_1 \wedge \ldots \wedge a_{n-1} \rightarrow \exists \vec{y}. a_n$, we denote by $\tau_q(\varphi)$ the boolean CQ$^{\neg\forall}$ $\exists \vec{x}. a_1 \wedge \ldots \wedge a_{n-1} \wedge \forall \vec{y}. \neg a_n$.
- if φ is a safe STGD, the negation of φ corresponds to a CQ$^{\neg s}$. In particular, if φ is the safe STGD $\forall \vec{x}. a_1 \wedge \ldots \wedge a_{n-1} \rightarrow a_n$, we denote by $\tau_q(\varphi)$ the boolean CQ$^{\neg s}$ $\exists \vec{x}. a_1 \wedge \ldots \wedge a_{n-1} \wedge \neg a_n$.

Finally, given a set of ICs Ψ such that each φ in Ψ is either an ED or a FD or a (safe) STGD, we denote by $\tau_q(\Psi)$ the boolean query corresponding to the union of the queries obtained by negating each IC in Ψ, i.e., $\tau_q(\Psi) = \bigvee_{\varphi\in\Psi} \tau_q(\varphi)$.

The following theorem establishes the correspondence between OWA-answering and OWA-consistency (w.l.o.g., we can assume that the query is boolean):

Theorem 2. *Let $\mathcal{C}, \Psi$ be sets of ICs, such that each $\varphi \in \Psi$ is either an ED or a FD or a STGD or a safe STGD, and let $\mathcal{D}$ be a database. Then, $\mathcal{D}$ is not OWA-consistent with $\mathcal{C} \cup \Psi$ iff $\tau_q(\Psi)$ is true in all databases in $sem(\mathcal{C}, \mathcal{D})$ (and $\mathcal{D}$ is not OWA$_f$-consistent with $\mathcal{C} \cup \Psi$ iff $\tau_q(\Psi)$ is true in all databases in $sem_f(\mathcal{C}, \mathcal{D})$).*

Proof. If it is not the case that $\tau_q(\Psi)$ is true in all databases in $sem(\mathcal{C}, \mathcal{D})$, then there exists a database $\mathcal{B}$ that satisfies $\mathcal{C}$, contains $\mathcal{D}$ and is such that $\tau_q(\Psi)$ is false in $\mathcal{B}$. Then, it follows immediately that $\mathcal{B}$ satisfies Ψ. Consequently, $\mathcal{D}$ is OWA-consistent with $\mathcal{C} \cup \Psi$. Conversely, suppose $\mathcal{D}$ is OWA-consistent with $\mathcal{C} \cup \Psi$. Then, there exists a database $\mathcal{B}$ that satisfies $\mathcal{C}\cup\Psi$ and contains $\mathcal{D}$. Now suppose that $\tau_q(\Psi)$ is true in $\mathcal{B}$: this implies that Ψ is not satisfied in $\mathcal{B}$, thus contradicting the hypothesis. Consequently, $\tau_q(\Psi)$ is false in $\mathcal{B}$. In the same way, we prove the thesis for finite databases. □

4. RESULTS FOR (UNIONS OF) CONJUNCTIVE QUERIES

In this section we analyze OWA-answering of CQs. We start by studying finite controllability of this problem in the presence of IDs.

Theorem 3. *OWA-answering CQs under IDs is finitely controllable.*

Proof (sketch). The proof is rather involved and requires several preliminary definitions and lemmas. In the following, we assume that m is the number of relation symbols in the database, and denote by k the maximum arity of such relations.

In order to prove finite controllability of CQs under IDs, we modify the *chase* procedure of [26] for inclusion dependencies, which, given a set of IDs $\mathcal{I}$ and a database instance $\mathcal{D}$, produces a (in general infinite) database $can(\mathcal{I}, \mathcal{D})$ (called the canonical chase). Our modified version always produces a finite database. However, differently from the canonical chase of [26], in this case we have to preliminarily fix the maximum size of the CQs. In other words, the finite chase will constitute a correct model of $\mathcal{I}$ and $\mathcal{D}$ only for CQs of size less or equal to n.

Definition 1. *Given a set of IDs $\mathcal{I}$, a database instance $\mathcal{D}$, and an integer $n \geq 1$, we denote by $fchase(\mathcal{I}, \mathcal{D}, n)$ the database obtained starting from $\mathcal{D}$ and closing the database with respect to the following ID-chase rule:*

if *$I \in \mathcal{I}$, with $I = r[A] \subseteq s[B]$*
and $r(t) \in fchase(\mathcal{I}, \mathcal{D}, n)$
and there is no fact in $fchase(\mathcal{I}, \mathcal{D}, n)$ of the form $s(t')$ such that $t'[B] = t[A]$,
then *add to $fchase(\mathcal{I}, \mathcal{D}, n)$ a fact $s(t')$ such that $t'[B] = t[A]$ and for each attribute p of s and such that $p \notin B$,*

$$t'[p] = f^{(j)}_{I,p}(trunc_n(t'[B]))$$

where:

- *$trunc_n(t)$ denotes the tuple obtained from t by eliminating the terms occurring at nesting level $\ell(n) + 1$;*
- *j is an integer such that $0 \leq j \leq \ell(n)$ and the function symbol $f^{(j)}_{I,p}$ does not occur in $trunc_n(t'[B])$.*

We call *existential value* every value $t'[p]$ introduced in $fchase(\mathcal{I}, \mathcal{D}, n)$ by an application of the ID-chase rule in Definition 1.

From Definition 1 it immediately follows that $fchase(\mathcal{I}, \mathcal{D}, n)$ satisfies the IDs in $\mathcal{I}$.

Now consider a specific construction of $fchase(\mathcal{I}, \mathcal{D}, n)$, i.e., starting from $fchase_0(\mathcal{I}, \mathcal{D}, n) = \mathcal{D}$, at each step i of the construction we nondeterministically choose a particular application of the ID-chase rule to a fact in $fchase_{i-1}(\mathcal{I}, \mathcal{D}, n)$, thus obtaining $fchase_i(\mathcal{I}, \mathcal{D}, n)$. This construction ends after a finite number of steps, because the nesting level of the functions in the terms representing existential values is bound to $\ell(n) + 1$, and the number of function and constant symbols used is finite, thus the number of distinct existential values introduced by the ID-chase rule is finite, and therefore the number of values involved in the construction of $fchase(\mathcal{I}, \mathcal{D}, n)$ is finite. Consequently, $fchase(\mathcal{I}, \mathcal{D}, n)$ is always a finite database.

Moreover, we can consider every $fchase(\mathcal{I}, \mathcal{D}, n)$ thus generated as a forest, where each node is a fact, the roots are the facts in $\mathcal{D}$, and there is an edge from a fact f to the fact f' iff, in the construction of $fchase(\mathcal{I}, \mathcal{D}, n)$, the fact f' has been obtained by applying the ID-chase rule to f.

Definition 2. *Let $f \in fchase(\mathcal{I}, \mathcal{D}, n)$. We denote by $history_n(f)$ the set of facts corresponding to the branch of the chase from the $\ell(n)$-th predecessor of f to f.*

The following lemma is an immediate consequence of the definitions of $fchase(\mathcal{I}, \mathcal{D}, n)$ and $can(\mathcal{I}, \mathcal{D})$.

Lemma 1. *There exists a homomorphism from $can(\mathcal{I}, \mathcal{D})$ to $fchase(\mathcal{I}, \mathcal{D}, n)$.*

In the following, given a boolean CQ q $\exists\vec{y}.a_1 \wedge \ldots \wedge a_n$, we call *image* of q a set of facts F such that there exists a substitution σ of the variables $\vec{y}$ such that $\{\sigma(a_1), \ldots, \sigma(a_n)\} = F$.

The next property derives from an analogous result in [26].

Lemma 2. *For each set of IDs $\mathcal{I}$, for each database instance $\mathcal{D}$, for each query q of size less or equal to n, and for each n-tuple t, if $t \in q^{fchase(\mathcal{I},\mathcal{D},n)}$ then there exists an image Im of $q(t)$ in $fchase(\mathcal{I}, \mathcal{D}, n)$ such that, for each pair of facts f, f' in Im, $history_n(f) \cap history_n(f') \neq \emptyset$.*

The following lemma is actually the key property for showing correctness of the "reuse" of existential values done by the ID-chase rule in the construction of $fchase(\mathcal{I}, \mathcal{D}, n)$. The lemma can be immediately proved from the construction mechanism of the skolem terms denoting the existential values.

Lemma 3. *For every fact f in $\mathit{fchase}(\mathcal{I},\mathcal{D},n)$, for every existential value v introduced by the ID-chase rule in f, and for every fact f' of $\mathit{fchase}(\mathcal{I},\mathcal{D},n)$ in which v is introduced by the ID-chase rule and such that $f \neq f'$, $\mathit{history}_n(f) \cap \mathit{history}_n(f') = \emptyset$.*

In words, the above lemma guarantees that in $\mathit{fchase}(\mathcal{I},\mathcal{D},n)$ the reuse of the same existential value v by the ID-chase rule is always done at a distance from the other occurrences of v that is sufficient to avoid that the "incorrect" (or unnecessary) equalities implied by the reuse of v change the evaluation of any conjunctive query of size n (or less).

We are now ready to prove correctness of the reuse of existential values in $\mathit{fchase}(\mathcal{I},\mathcal{D},n)$.

Lemma 4. *For each set of IDs $\mathcal{I}$, for each database instance $\mathcal{D}$, for each query q of size less or equal to n, and for each n-tuple t, $t \in q^{\mathit{fchase}(\mathcal{I},\mathcal{D},n)}$ iff $t \in q^{\mathit{can}(\mathcal{I},\mathcal{D})}$.*

We are finally ready to prove Theorem 3. Indeed, given a query q of size n, from Lemma 4 it follows that the database $\mathit{fchase}(\mathcal{I},\mathcal{D},n)$ constitutes a canonical model for q, i.e., $t \in \mathit{ans}(q,\mathcal{I},\mathcal{D})$ iff $t \in q^{\mathit{fchase}(\mathcal{I},\mathcal{D},n)}$. In particular, if $t \notin \mathit{ans}(q,\mathcal{I},\mathcal{D})$ then $t \notin q^{\mathit{fchase}(\mathcal{I},\mathcal{D},n)}$, and thus, since $\mathit{fchase}(\mathcal{I},\mathcal{D},n)$ is a finite database, $t \notin \mathit{ans}_f(q,\mathcal{I},\mathcal{D})$. On the other hand, the fact that $t \in \mathit{ans}(q,\mathcal{I},\mathcal{D})$ implies $t \in \mathit{ans}_f(q,\mathcal{I},\mathcal{D})$ trivially follows from definition of *ans* and ans_f. Consequently, $t \in \mathit{ans}(q,\mathcal{I},\mathcal{D})$ iff $t \in \mathit{ans}_f(q,\mathcal{I},\mathcal{D})$, which proves the thesis. □

The above theorem can be easily extended to unions of conjunctive queries.

Theorem 4. *OWA-answering UCQs under IDs is finitely controllable.*

It is then possible to prove the analogous of Theorem 4 for the case of KDs and FKs.

Theorem 5. *OWA-answering UCQs under single KDs and FKs is finitely controllable.*

Then, we prove that, as soon as we extend the ICs beyond single KDs and FKs, finite controllability of OWA-answering of CQs does not hold anymore.

Theorem 6. *OWA-answering CQs under non-conflicting KDs and IDs is not finitely controllable.*

Proof (sketch). Let $\mathcal{C}$ be the set of non-conflicting KDs and IDs constituted by the ID $r[2] \subseteq r[1]$ and the KD $\mathit{key}(r) = 2$. It is immediate to verify that $\mathcal{C}$ implies the ID $I = r[1] \subseteq r[2]$ over finite databases, while $\mathcal{C}$ does not imply I over unrestricted databases.

Consequently, given an instance $\mathcal{D} = \{r(a,b)\}$, the query $\exists x.r(x,a)$ is true over finite databases while it is false over unrestricted databases. □

Then, we recall a result presented in [9] for OWA-answering UCQs under non-conflicting KDs and IDs over unrestricted databases.

Proposition 1. *[9, Theorem 3.9] OWA-answering UCQs under non-conflicting KDs and IDs is decidable, in particular it is in PTIME in data complexity and in PSPACE in combined complexity.*

Finally, we prove that the above property cannot be extended to the case of finite databases.

Theorem 7. *Finite OWA-answering CQs under non-conflicting KDs and IDs is undecidable.*

Proof (sketch). We prove the theorem by reducing implication of IDs from FDs and IDs (which is not finitely controllable [11], and is undecidable both for finite databases and for unrestricted databases [30, 12]) to OWA-answering of CQs under non-conflicting KDs and IDs. Given a set of FDs $\mathcal{F}$ which contains m FDs, a set of IDs $\mathcal{I}$, and an ID I, we define a set of KDs $\mathcal{K}'$ and a set of IDs $\mathcal{I}'$ as follows: we start from $\mathcal{K} = \emptyset$ and $\mathcal{I}' = \mathcal{I}$. Then, for each FD in $\mathcal{F}$: if the i-th FD in $\mathcal{F}$ is of the form $r : i_1,\ldots,i_k \rightarrow b$ (such a FD is denoted in the following by F_i), we use an auxiliary relation r_i (i.e., a new relation symbol that does not already occur in $\mathcal{F} \cup \mathcal{I}' \cup \{I\}$) of arity $2k+1$, add to $\mathcal{K}'$ the KD $\mathit{key}(r_i) = k+1,\ldots,2k$, and add to $\mathcal{I}'$ the IDs

$$\begin{array}{l} r_i[k+1,\ldots,2k] \subseteq r_i[1,\ldots,k] \\ r[i_1,\ldots,i_k,b] \subseteq r[1,\ldots,k,2k+1] \end{array}$$

Finally, if the ID I has the form $I = r[l_1,\ldots,l_h] \subseteq s[j_1,\ldots,j_h]$ (where r has arity n and s has arity p), we define $\mathcal{D}(I)$ as the database $\mathcal{D} = \{r(t)\}$ with $t = c_1,\ldots,c_n$, and define $q(I)$ as the boolean CQ $\exists x_1,\ldots,x_p.s(v_1,\ldots,v_p)$ where each v_i is such that $v_i = c_{l_k}$ if $i = j_k$ for some k s.t. $1 \leq k \leq h$, while $v_i = x_i$ otherwise. Notice that the set $\mathcal{K}' \cup \mathcal{I}'$ thus constructed is a set of non-conflicting KDs and IDs. It is now possible to show that $\mathcal{F} \cup \mathcal{I} \models_f I$ iff the CQ $q(I)$ is true in all databases of $\mathit{sem}_f(\mathcal{K}' \cup \mathcal{I}', \mathcal{D}(I))$. □

Observe that the above results identify the first combination of ICs and query language (CQs under non-conflicting KDs and IDs) in which OWA-answering is decidable for unrestricted databases and is undecidable over finite databases.

5. QUERIES WITH INEQUALITIES

In this section we analyze UCQ$^{\neq}$, i.e., UCQs where the presence of the inequality predicate $\neq$ is allowed. As shown by the following theorem, the possibility of expressing inequalities changes drastically the finite controllability and decidability properties of OWA-answering for UCQs.

Theorem 8. *OWA-answering UCQ$^{\neq}$s under IDs is not finitely controllable, and is undecidable both for finite databases and for unrestricted databases.*

Proof (sketch). We prove the theorem by reducing implication of IDs from FDs and IDs (which is not finitely controllable [11], and is undecidable both for finite databases and for unrestricted databases [30, 12]) to OWA-answering of UCQ$^{\neq}$s under IDs. First, observe that, given a set of FDs $\mathcal{F}$, the query $\tau_q(\mathcal{F})$ (see Section 3) is a boolean UCQ$^{\neq}$.

Now, given a set of FDs $\mathcal{F}$, a set of IDs $\mathcal{I}$, and an ID $I = r[l_1,\ldots,l_h] \subseteq s[j_1,\ldots,j_h]$ (where r has arity n and s has arity p), we define $\mathcal{D}(I)$ as the database $\mathcal{D} = \{r(t)\}$ with $t = c_1,\ldots,c_n$, and define $q(I)$ as the boolean CQ $\exists x_1,\ldots,x_p.s(v_1,\ldots,v_p)$ where each v_i is such that $v_i = c_{l_k}$ if $i = j_k$ for some k s.t. $1 \leq k \leq h$, while $v_i = x_i$ otherwise.

We now prove that $\mathcal{F} \cup \mathcal{I} \models I$ iff the UCQ$^{\neq}$ $\tau_q(\mathcal{F}) \vee q(I)$ is true in all databases of $\mathit{sem}(\mathcal{I},\mathcal{D}(I))$. The proof follows immediately from the fact that, for each database $\mathcal{B} \in \mathit{sem}(\mathcal{I},\mathcal{D}(I))$:

- $\mathcal{F}$ is not satisfied in $\mathcal{B}$ iff $\tau_q(\mathcal{F})$ is true in $\mathcal{B}$;
- if $\mathcal{F}$ is satisfied in $\mathcal{B}$ and I is not satisfied in $\mathcal{B}$, then there are no facts of the form $s(t')$ such that $t'[B] = t[A]$, which implies that the query $q(I)$ is false in $\mathcal{B}$;
- if $\mathcal{F}$ is satisfied in $\mathcal{B}$ and I is satisfied in $\mathcal{B}$, then there is a fact of the form $s(t')$ such that $t'[B] = t[A]$, which implies that the query $q(I)$ is true in $\mathcal{B}$.

Consequently, if $\tau_q(\mathcal{F}) \vee q(I)$ is true in $\mathcal{B}$ then either $\mathcal{B}$ does not satisfy $\mathcal{F}$ or $\mathcal{B}$ satisfies $\mathcal{F}$ and I, while if $\tau_q(\mathcal{F}) \vee q(I)$ is false in $\mathcal{B}$ then either $\mathcal{B}$ satisfies $\mathcal{F}$ or does not satisfy I. The above proof also holds for finite databases, i.e., it also shows that $\mathcal{F} \cup \mathcal{I} \models_f I$ iff the UCQ$^{\neq}$ $\tau_q(\mathcal{F}) \vee q(I)$ is true in all databases of $sem_f(\mathcal{I}, \mathcal{D}(I))$. □

6. QUERIES WITH NEGATION

We now turn our attention to the classes of queries previously introduced which allow for the presence of negated atoms. As in the case of inequalities, we will show in this section that adding negation makes it very easy to lose finite controllability and decidability of OWA-answering, even under quite simple forms of ICs. We start our analysis from queries with safe negation.

Theorem 9. *OWA-answering UCQ$^{\neg s}$s under IDs is not finitely controllable.*

Proof (sketch). Let I be the ID $r[2] \subseteq r[1]$, let $\mathcal{I} = \{I\}$, let q be the UCQ$^{\neg s}$ $(\exists x, y, z.r(x, y) \wedge r(y, z) \wedge \neg r(x, z)) \vee (\exists x.r(x, x))$, and let $\mathcal{D}$ be the database $\{r(a, b)\}$. First, q is true in all databases in $sem_f(\mathcal{I}, \mathcal{D})$: indeed, for each finite database $\mathcal{B}$ in $sem_f(\mathcal{I}, \mathcal{D})$, if r is not transitive in $\mathcal{B}$, then q is true in $\mathcal{B}$ (since the first disjunct of q corresponds to the negation of the transitivity property), while if r is transitive in $\mathcal{B}$, then, due to the seriality of r imposed by the ID I, it follows that r has a cycle in $\mathcal{B}$, and therefore by transitivity of r in $\mathcal{B}$ there exists a constant c such that $r(c, c) \in \mathcal{B}$, which implies that the second disjunct of q is true in $\mathcal{B}$. On the other hand, it is immediate to see that there exists an infinite database in $sem(\mathcal{I}, \mathcal{D})$ in which q is false, i.e., in which r is both serial and transitive but has no cycles. □

Theorem 10. *Both OWA-answering and finite OWA-answering of UCQ$^{\neg s}$s under IDs are undecidable.*

Proof (sketch). First, OWA-consistency (and OWA$_f$-consistency) under STGDs is undecidable, which easily follows from undecidability of implication (and finite implication) of STGDs [6, 35]. Then, we reduce OWA-consistency under STGDs to OWA-consistency under IDs and safe STGDs. The reduction is very simple: starting from a set of STGDs, we replace every STGD of the form

$$\forall \vec{x}.a_1 \wedge \ldots \wedge a_n \rightarrow \exists \vec{y}.r(\vec{x}, \vec{y})$$

(where w.l.o.g. we assume that the variables from $\vec{x}$ occur in the first k arguments of $r(\vec{x}, \vec{y})$) with the following two ICs: the safe STGD

$$\forall \vec{x}.a_1 \wedge \ldots \wedge a_n \rightarrow aux(\vec{x})$$

(where $aux(\vec{x})$ is an atom with k arguments which are exactly the first k arguments of the atom $r(\vec{x}, \vec{y})$) and the ID

$$aux[1, \ldots, k] \subseteq r[1, \ldots, k]$$

in which aux is a new auxiliary relation of arity k (so we introduce one new auxiliary relation for each STGD). We have thus constructed a set of safe STGDs and IDs. Correctness of the reduction is straighforward (the above reduction is also correct for OWA$_f$-consistency). Finally, we reduce OWA-consistency under safe STGDs and IDs to OWA-answering of UCQ$^{\neg s}$s under IDs. Given a set of safe STGDs $\mathcal{S}$, a set of IDs $\mathcal{I}$, the query $\tau_q(\mathcal{S})$ (see Section 3) is a boolean UCQ$^{\neg s}$, and from Theorem 2 it follows that, for each database $\mathcal{D}$, $\mathcal{D}$ is not OWA-consistent with $\mathcal{S} \cup \mathcal{I}$ iff $\tau_q(\mathcal{S})$ is true in all databases in $sem(\mathcal{I}, \mathcal{D})$ (and in the same way we reduce OWA$_f$-consistency under safe STGDs and IDs to finite OWA-answering of UCQ$^{\neg s}$s under IDs). □

We now turn our attention to OWA-answering of CQ$^{\neg s}$.

Theorem 11. *OWA-answering of CQ$^{\neg s}$ under single KDs and FKs is not finitely controllable, and is undecidable both for finite and for unrestricted databases.*

Finally, we prove that OWA-answering UCQ$^{\neg\forall}$s is undecidable, *even in the absence of integrity constraints.* We start from two auxiliary lemmas about implication of IDs and EDs from STGDs.

Lemma 5. *Let $\mathcal{S}$ be a set of STGDs, and let I be a ID. The problem of establishing whether $\mathcal{S} \models I$ is not finitely controllable and is undecidable both for finite and for unrestricted databases.*

Proof. Follows immediately from the results on implication of STGDs from STGDs in [6, 35], which prove that the implication problem for STGDs (called template dependencies) is not finitely controllable and undecidable both for finite and for unrestricted databases. □

Lemma 6. *Let $\mathcal{S}$ be a set of STGDs and EDs, and let I be an ED. The problem of establishing whether $\mathcal{S} \models I$ is not finitely controllable and is undecidable both for finite and for unrestricted databases.*

Proof. We reduce implication of IDs from STGDs to implication of EDs from STGDs and EDs. Let $\mathcal{S}$ be a set of STGDs and let I_1 be a generic ID of the form $r_1[i_1, \ldots, i_m] \subseteq r_2[j_1, \ldots, j_m]$. Let s be a new relation symbol (not appearing in $\mathcal{S}$). Now let E be the following ED: $s[1, \ldots, m] \cap r_1[i_1, \ldots, i_m] = \emptyset$, and let E' be the following ED: $s[1, \ldots, m] \cap r_2[j_1, \ldots, j_m] = \emptyset$. It is immediate to verify that $\mathcal{S} \models I$ iff $\mathcal{S} \cup \{E\} \models E'$. Consequently, by Lemma 5 the thesis follows. □

Theorem 12. *OWA-consistency under STGDs and EDs is undecidable.*

Proof. The proof follows from Theorem 1 and from Lemma 6. □

Theorem 13. *OWA-answering of UCQ$^{\neg\forall}$s (even in the absence of ICs) is not finitely controllable, and is undecidable both for finite and for unrestricted databases.*

Proof. We reduce OWA-consistency under STGDs and EDs to OWA-answering of UCQ$^{\neg\forall}$s. Let $\mathcal{S}$ be a set of STGDs and EDs, and let $\mathcal{D}$ be a database. From Theorem 2, it follows that $\mathcal{D}$ is not OWA-consistent with $\mathcal{S}$ iff

$\tau_q(\mathcal{S})$ is true in all databases in $sem(\emptyset, \mathcal{D})$ (and $\mathcal{D}$ is not OWA$_f$-consistent with $\mathcal{S}$ iff $\tau_q(\mathcal{S})$ is true in all databases in $sem_f(\emptyset, \mathcal{D})$). Then, it is immediate to verify that $\tau_q(\mathcal{S})$ is a UCQ$^{\neg\forall}$: indeed, $\tau_q(\mathcal{S}) = \bigvee_{I \in \mathcal{S}} \tau_q(I)$, and, if I is an ED, then $\tau_q(I)$ is a CQ (i.e., a special case of a CQ$^{\neg\forall}$), while if I is a STGD, then $\tau_q(I)$ is a CQ$^{\neg\forall}$. □

Therefore, the above theorem shows that even for very small fragments of domain-independent first-order queries with unsafe negation, e.g., UCQ$^{\neg\forall}$s, and even in the absence of ICs, OWA-answering is not finitely controllable and is undecidable.

7. FROM OWA-ANSWERING TO QUERY CONTAINMENT

In this section we introduce query containment under ICs and relate the results for OWA-answering presented above to query containment.

Given two queries q_1 and q_2 and a set of ICs $\mathcal{C}$, we say that q_1 *is contained in* q_2 *under* $\mathcal{C}$ (denoted by $q_1 \subseteq_{\mathcal{C}} q_2$) if, for each database $\mathcal{B}$ that satisfies $\mathcal{C}$, $q_1^{\mathcal{B}} \subseteq q_2^{\mathcal{B}}$.

When the query q_1 is a CQ, the relationship between OWA-answering and query containment can be informally explained as follows (for more details see e.g. [26]). In the absence of ICs, we "freeze" q_1 by replacing each distinct variable with a distinct constant in q_1 through a substitution σ, thus obtaining a set of facts, i.e., a database $\mathcal{D}(q_1)$. Then, it can be shown that $q_1 \subseteq_{\mathcal{C}} q_2$ iff $t \in ans(q_2, \mathcal{C}, \mathcal{D}(q_1))$, where $t = \sigma(\vec{x})$. In the presence of a set of ICs $\mathcal{C}$, we must add a unification phase to the above procedure, since the ICs may imply equalities on the constants used for freezing the query q_1 (so the terms used for freezing q_2 are now "soft" constants): if $\mathcal{C}$ is such that implication of ICs under $\mathcal{C}$ is decidable, then also this unification is computable in a finite amount of time.

As a consequence of the above reduction, all the decidability and finite controllability results for OWA-answering presented in this paper can be easily extended to the corresponding query containment problems.

Due to lack of space, in the present version of the paper we omit details and comments on the results for query containment thus derived. We only point out the two following results.

The following property immediately follows from Theorem 3.

Corollary 1. *Containment between CQs under IDs is finitely controllable.*

Analogously, the following corollary follows from Theorem 5.

Corollary 2. *Containment between CQs under single KDs and FKs is finitely controllable.*

The above two properties close two problems left open in [26], which established finite controllability of containment between CQs under *unary* IDs (i.e., IDs with arity 1) and under the so-called *key-based* dependencies, which constitute a combination of KDs and IDs much more restricted than single KDs and FKs, and left open the problem of finite controllability under arbitrary IDs and under more expressive combinations of KDs and IDs.

8. RELATED WORK

Query answering and containment under ICs With respect to query containment, the most closely related work is certainly [26], which shows decidability of containment of CQs under IDs (which immediately implies decidability of OWA-answering of CQs under IDs) and under the class of key-based dependencies (that has already been introduced in Section 7). These results have been extended in [9] to containment (and OWA-answering) of CQs under non-conflicting KDs and IDs for unrestricted databases.

The work in [5, 4] present results on undecidability of first-order query answering using unary conjunctive views. This setting is quite different from the one studied in the present paper, which actually cannot be reduced to the framework of unary conjunctive views (and vice versa). However, although in different settings, some of the results (in particular with respect to the use of negation and inequality) are similar.

View-based query processing is also closely related to OWA query answering. We only mention the approach presented in [1, 15], which studies query answering using views. In particular, [15] analyzes the presence of ICs, in particular functional dependencies, in this setting.

Many decidability results have been established for classes of ICs which admit a finite *chase*, i.e., a finite "canonical model" for the database and the ICs (see [8, 2]). For instance, [37] studies containment of conjunctive queries under (a generalized form of) acyclic IDs and FDs (whose chase is finite). Moreover, the approach presented in [14] studies containment of conjunctive queries under Datalog ICs, i.e., ICs that can be expressed in terms of a Disjunctive Datalog program. Again, Disjunctive Datalog programs cannot express arbitrary IDs, so the kinds of ICs analyzed in the present paper are not covered by the results in [14]. A similar setting is studied in [32, 33] (although under a least-fixpoint-based semantics that differs from the one presented in this paper), which also present results about conjunctive queries with inequality predicates which extend the one in [27]. Also, [16, 17] present results about query answering in a combination of dependencies for which the chase is finite, although in the different setting of *data exchange*. In particular, conjunctive queries and conjunctive queries with inequalities are studied.

Instead, in the present paper we have studied classes of ICs for which the chase is in general infinite, since we admit IDs with arbitrary cycles. This is the main technical difficulty of our work, and one of the main differences with respect to the above mentioned studies.

Finally, we point out that the results presented in this paper complement a previous result [10] which states that OWA-answering of positive Datalog queries both under IDs and under single KDs and FKs is undecidable.

Implication of ICs Many studies have dealt with the implication problem for FDs and IDs. Besides the "classical" results already cited in the previous sections, below we briefly describe some works which have a close relation to the present paper.

In [13] the authors identify one of the first combinations of ICs (namely, unary FDs and unary IDs) for which implication is not finitely controllable, although decidable both for the finite and for the unrestricted case. In this respect, our results about CQs under non-conflicting KDs and IDs

(Theorem 7) identify the first (to our knowledge) class of FDs and IDs under which finite model reasoning is undecidable while unrestricted model reasoning is decidable.

The work presented in [29] defines a notion of non-conflicting FDs and IDs and proves decidability of implication from such ICs. Our notion of non-conflicting KDs and IDs is significantly different, because we take into account cyclic IDs, which cause the chase to be infinite, while in [29] only proper-circular IDs are considered (i.e., a class of IDs that has a finite chase).

Finally, [20, 19] have studied integrity constraints for XML. To this aim, they have shown that the implication problem for KDs and FKs is undecidable, which apparently contradicts our decidability results for KDs and FKs. However, we point out that the notion of foreign key in [20, 19] is different from ours: actually, since in [20, 19] a FK may involve a superset of a key, it follows that a set of keys and foreign keys according to [20, 19] is a set of conflicting KDs and IDs according to our classification, and hence OWA-answering under such ICs is undecidable, which agrees with the results in [20, 19].

9. CONCLUSIONS

The table displayed in Figure 1 summarizes the results presented in this paper. In the table, each column corresponds to a different query language, while each row corresponds to a different class of ICs. Each class of IC and each query language identifies a cell, which is divided into three (or two) sub-cells. The first sub-cell, whose corresponding sub-row is labeled by FC, indicates whether OWA-answering for the corresponding combination of ICs and queries is finitely controllable, while the two subsequent sub-cells indicate whether OWA-answering is undecidable (over unrestricted databases and over finite databases, respectively). If it is not the case, the sub-cell displays the complexity of OWA-answering (the first class refers to data complexity, the second one to combined complexity).[2] Some of the results reported in the table are already known or follow trivially from known results: the new results are printed in boldface type.

As for further developments of the present work, we believe that one of the most interesting aspects to investigate is the extension of the analysis presented in this paper towards different kinds of IC/schema languages (data models, ontology languages, etc.) and query languages. In particular, we conjecture that our results may imply interesting results for different schema languages that have the ability of expressing forms of key dependencies, inclusion dependencies, and exclusion dependencies among data, with special regard to the design of decidable query languages for such schemas.

Acknowledgments

This research has been partially supported by the projects TONES (FP6-7603) and INTEROP Network of Excellence (IST-508011) funded by the EU, by project HYPER, funded by IBM through a Shared University Research (SUR) Award grant, and by MIUR FIRB 2005 project “Tecnologie Orientate alla Conoscenza per Aggregazioni di Imprese in Internet” (TOCAI.IT). The author wishes to thank Maurizio Lenzerini for useful discussions on the subject of the paper.

[2] Due to space limitations, in the present version of the paper we omit details about the complexity results displayed in the table: however, all the new complexity results follow from previous results and from the theorems presented in this paper. We point out that, for the classes different from P (which states a membership result), the complexity of the problem is complete with respect to the class reported.

10. REFERENCES

[1] S. Abiteboul and O. Duschka. Complexity of answering queries using materialized views. In *Proc. of PODS'98*, pages 254–265, 1998.

[2] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison Wesley Publ. Co., 1995.

[3] M. Arenas, L. E. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *Proc. of PODS'99*, pages 68–79, 1999.

[4] J. Bailey and G. Dong. Decidability of first-order logic queries over views. In *Proc. of ICDT'99*, pages 83–99, 1999.

[5] J. Bailey, G. Dong, and K. Ramamohanarao. Decidability and undecidability results for the termination problem of active database rules. In *Proc. of PODS'98*, pages 264–273, 1998.

[6] C. Beeri and M. Y. Vardi. The implication problem for data dependencies. In *Proc. of ICALP'81*, pages 73–85, 1981.

[7] C. Beeri and M. Y. Vardi. Formal systems for tuple and equality generating dependencies. *SIAM J. on Computing*, 13(1):76–98, 1984.

[8] C. Beeri and M. Y. Vardi. A proof procedure for data dependencies. *J. of the ACM*, 31(4):718–741, 1984.

[9] A. Calì, D. Lembo, and R. Rosati. On the decidability and complexity of query answering over inconsistent and incomplete databases. In *Proc. of PODS 2003*, pages 260–271, 2003.

[10] D. Calvanese and R. Rosati. Answering recursive queries under keys and foreign keys is undecidable. In *Proc. of KRDB 2003*. CEUR Electronic Workshop Proceedings, `http://ceur-ws.org/Vol-79/`, 2003.

[11] M. A. Casanova, R. Fagin, and C. H. Papadimitriou. Inclusion dependencies and their interaction with functional dependencies. *J. of Computer and System Sciences*, 28(1):29–59, 1984.

[12] A. K. Chandra and M. Y. Vardi. The implication problem for functional and inclusion dependencies is undecidable. *SIAM J. on Computing*, 14(3):671–677, 1985.

[13] S. S. Cosmadakis, P. C. Kanellakis, and M. Vardi. Polynomial-time implication problems for unary inclusion dependencies. *J. of the ACM*, 37(1):15–46, Jan. 1990.

[14] G. Dong and J. Su. Conjunctive query containment with respect to views and constraints. *Information Processing Letters*, 57(2):95–102, 1996.

[15] O. M. Duschka, M. R. Genesereth, and A. Y. Levy. Recursive query plans for data integration. *J. of Logic Programming*, 43(1):49–73, 2000.

[16] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: Semantics and query answering. In *Proc. of ICDT 2003*, pages 207–224, 2003.

		CQ/UCQ	$\text{CQ}^{\neg s}$	$\text{UCQ}^{\neq}$	$\text{UCQ}^{\neg s}$	$\text{UCQ}^{\neg\forall}$
no ICs	FC	yes	yes	yes	yes	**no**
	unres.	P/NP	P/NP	P/NP	P/EXPTIME	**undecid.**
	finite					**undecid.**
KDs (FDs)	FC	yes	yes	yes	yes	**no**
	unres.	P/NP	P/NP	P/NP	P/EXPTIME	**undecid.**
	finite					**undecid.**
IDs	FC	**yes**	**yes**	**no**	**no**	**no**
	unres.	P/PSPACE	**P/PSPACE**	**undecid.**	**undecid.**	**undecid.**
	finite	**P/PSPACE**	**P/PSPACE**	**undecid.**	**undecid.**	**undecid.**
KDs+FKs	FC	**yes**	**no**	**no**	**no**	**no**
	unres.	P/PSPACE	**undecid.**	**undecid.**	**undecid.**	**undecid.**
	finite	**P/PSPACE**	**undecid.**	**undecid.**	**undecid.**	**undecid.**
non-conflicting KDs+IDs	FC	**no**	**no**	**no**	**no**	**no**
	unres.	P/PSPACE	**undecid.**	**undecid.**	**undecid.**	**undecid.**
	finite	**undecid.**	**undecid.**	**undecid.**	**undecid.**	**undecid.**
KDs+IDs	FC	no	no	no	no	no
	unres.	undecid.	undecid.	undecid.	undecid.	undecid.
	finite	undecid.	undecid.	undecid.	undecid.	undecid.

Figure 1: Complexity of OWA-answering (decision problem) under ICs

[17] R. Fagin, P. G. Kolaitis, and L. Popa. Data exchange: Getting to the core. In *Proc. of PODS 2003*, pages 90–101, 2003.

[18] R. Fagin, P. G. Kolaitis, L. Popa, and W.-C. Tan. Composing schema mappings: Second-order dependencies to the rescue. In *Proc. of PODS 2004*, 2004.

[19] W. Fan and L. Libkin. On XML integrity constraints in the presence of DTDs. *J. of the ACM*, 49(3):368–406, 2002.

[20] W. Fan and J. Siméon. Integrity constraints for XML. In *Proc. of PODS 2000*, pages 23–34, 2000.

[21] M. H. Graham, A. O. Meldelzon, and M. Y. Vardi. Notions of dependency satisfaction. *J. of the ACM*, 33(1):105–129, 1986.

[22] A. Halevy, Z. Ives, D. Suciu, and I. Tatarinov. Schema mediation in peer data management systems. In *Proc. of ICDE 2003*, pages 505–516, 2003.

[23] A. Y. Halevy. Answering queries using views: A survey. *VLDB Journal*, 10(4):270–294, 2001.

[24] A. Y. Halevy, I. S. Mumick, Y. Sagiv, and O. Shmueli. Static analysis in Datalog extensions. *J. of the ACM*, 48(5):971–1012, 2001.

[25] U. Hustadt, B. Motik, and U. Sattler. Data complexity of reasoning in very expressive description logics. In *Proc. of IJCAI 2005*, 2005.

[26] D. S. Johnson and A. C. Klug. Testing containment of conjunctive queries under functional and inclusion dependencies. *J. of Computer and System Sciences*, 28(1):167–189, 1984.

[27] A. C. Klug. On conjunctive queries containing inequalities. *J. of the ACM*, 35(1):146–160, 1988.

[28] M. Lenzerini. Data integration: A theoretical perspective. In *Proc. of PODS 2002*, pages 233–246, 2002.

[29] M. Levene and G. Loizou. How to prevent interaction of functional and inclusion dependencies. *Information Processing Letters*, 71(4):115–125, 1999.

[30] J. C. Mitchell. The implication problem for functional and inclusion dependencies. *Information and Control*, 56:154–173, 1983.

[31] B. Trahktenbrot. Impossibility of an algorithm for the decision problem in finite classes. *Transactions of the American Mathematical Society*, 3:1–5, 1963.

[32] R. van der Meyden. *The Complexity of Querying Indefinite Information.* PhD thesis, Rutgers University, 1992.

[33] R. van der Meyden. The complexity of querying indefinite data about linearly ordered domains. *J. of Computer and System Sciences*, 54(1):113–135, 1997.

[34] R. van der Meyden. Logical approaches to incomplete information. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*, pages 307–356. Kluwer Academic Publisher, 1998.

[35] M. Y. Vardi. The implication and finite implication problems for typed template dependencies. In *Proc. of PODS'82*, pages 230–238, 1982.

[36] M. Y. Vardi. On the integrity of databases with incomplete information. In *Proc. of PODS'82*, pages 252–266, 1982.

[37] X. Zhang and Z. M. Ozsoyoglu. Implication and referential constraints: A new formal reasoning. *IEEE Trans. on Knowledge and Data Engineering*, 9(6):894–910, 1997.