Query Answering Under Exact View Assumption in Local As View Data Integration System.*

Andrei Lopatenko[1,2]

[1] Free University of Bozen–Bolzano, Faculty of computer Science, Italy,
lopatenko@inf.unibz.it
[2] University of Manchester, Department of Computer Science, UK

Abstract. In this paper we thoroughly analyze a problem of query answering under Exact View Assumption (EVA) in a data integration system. We propose an algorithm to build a representation of all legal global database instances using conditional tables. Other approaches for query answering as well data and program complexity are analyzed. We found a few tractable cases of answering under EVA using reduction to Constraint Satisfaction Problem.

1 Introduction

There is an increasing number of available information sources, many of them providing access to data in different formats. Data Integration systems (DIS) aim to provide a uniform query interface to all information sources. One of the views is to view such system as Local As View approach: 1) to postulate a global schema that provides a unifying data model for all information sources, 2) to model each sources as a materialized view defined in terms of global schema, which are virtual. Such approach defines *a Query Answering* problem: given a query in terms of global schema, answer it using data from data sources.

There are different assumptions maybe given in data integration systems about the quality of information contained in data sources. One of the criteria of the quality of information is a completeness and correctness of information: under Sound View Assumption (**SVA**) it is assumed that data source contains only correct information, but it maybe incomplete; under Exact View Assumption (**EVA**) it is assumed that a data source contains complete and correct information (EVA = SVA + Closed World Assumption).

In this paper we will work on the query answering under Exact View Assumption, where information contained in views is complete and correct. At first step, we provide a reduction from query answering under EVA to query answering under SVA, and under special kind of integrity constraints.

At next step, we will construct a chase procedure, which is being applied a representation of all databases legal under SVA, constructs a database representation of all databases valid under EVA. Using this chase procedure we will get else one proof of upper bounds for query answering under EVA.

At next step we consider a logical programming approach for solving query answering

* This work is done under supervision by Prof. Enrico Franconi, Free University of Bozen-Bolzano

under EVA problem. Developing this approach, we get some characterization of legal databases under EVA
And finally, we will find some tractable cases of query answering problem using reduction to Constraint Satisfaction Problem.

1.1 Preliminaries

To formalize a notion of Data Integration System we use a framework proposed in [Lenzerini, 2002], [Abiteboul & Duschka, 1998].

Definition 1 Data integration system. *A data integration system* $\mathcal{I}$ *is a triple* $\langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, *where*

- $\mathcal{G}$ *is the global schema expressed in the relational model with integrity constraints. Henceforth, we indicate with* $\mathcal{A}_\mathcal{G}$ *the alphabet of the relation symbols of* $\mathcal{G}$ *and with* $\mathcal{C}_\mathcal{G}$ *the set of integrity constraints specified on* $\mathcal{A}_\mathcal{G}$.
- $\mathcal{S}$ *is the source schema, constituted by the schemata of the various sources that are part of the integration system. Henceforth, we indicate with* $\mathcal{A}_\mathcal{S}$ *the alphabet of the relation symbols (also sometimes called materialized views or simply views) on of* $\mathcal{S}$. *Is is disjoint of* $\mathcal{A}_\mathcal{G}$.
- $\mathcal{M}$ *is the mapping between the global and source schema. The mapping is a set of assertions of the form* $\langle \mathcal{Q}_\mathcal{G} \ \zeta \ \mathcal{Q}_\mathcal{S} \rangle$, *where* $\mathcal{Q}_\mathcal{G}$ *and* $\mathcal{Q}_\mathcal{S}$ *are queries of the same arity, respectively over the global schema* $\mathcal{G}$ *and over the source schema* $\mathcal{S}$. *Queries* $Q_\mathcal{G}$ *are expressed in a query language* $\mathcal{L}_{\mathcal{M},\mathcal{G}}$ *and queries* $\mathcal{Q}_\mathcal{S}$ *are expressed in a query language* $\mathcal{L}_{\mathcal{M},\mathcal{S}}$. *The symbol* ζ *is a symbol from the alphabet* $\{\subset, \subseteq, \equiv, \supseteq, \supset\}$.

Definition 2 Source database. *A source database* $\mathcal{D}_\mathcal{S}$ *for* $\mathcal{I}$ *is a database* $\mathcal{D}_\mathcal{S}$ *that conforms to the source schema* $\mathcal{S}$ *and satisfies all constraints in* $\mathcal{S}$.

We will often use a term *view* to denote source database.

Definition 3 Global database. *A global database* $\mathcal{B}$ *for* $\mathcal{I}$ *is any database for* $\mathcal{G}$. *A global database* $\mathcal{B}$ *is legal with respect to* $\mathcal{D}_\mathcal{S}$, *if:*

- $\mathcal{B}$ *is legal with respect to* $\mathcal{G}$, *i.e.* $\mathcal{B}$ *satisfies all the constraints in* $\mathcal{G}$*;*
- $\mathcal{B}$ *satisfies the mappings* $\mathcal{M}$ *with respect to* $\mathcal{D}_\mathcal{S}$.

A global database which is legal with respect to given data integration system may be also called retrieved *database.*

Definition 4 Local-As-View approach (LAV). *The mapping in data integration system is given by associating each source relation symbol with a view over global relation symbols. In LAV the mapping given as a set of triples* $\langle \mathcal{A}_\mathcal{S} \ Q(\mathcal{A}_\mathcal{G}) \ \zeta \rangle$ *where* ζ *is a symbol of the alphabet* $\{sound, complete, exact\}$ *and* ζ *means which assumption is valid for given view.*

Sound views assumption, SWA. The extension of the view contains a subset of tuples derived from database. $E_{V_i} \subseteq Q_{\mathcal{G}}^{V_i}(\mathcal{D})$ or, in other words, $\forall \bar{x}\ s(\bar{x}) \ \rightarrow\ q_{\mathcal{G}}^{s}(\bar{x})$, where *s* is a source relation symbol, $q_{\mathcal{G}}^{s}$ is a query over global schema relation symbols, associated with the view *s*, $\bar{x}$ denotes a set of variables of arity of *s*.

Complete views assumption, CWA. The extension of the view contains all tuples derived from database. $E_{V_i} \supseteq Q_{\mathcal{G}}^{V_i}(\mathcal{D})$ or, in other words, $\forall \bar{x}\ s(\bar{x}) \ \leftarrow\ q_{\mathcal{G}}^{s}(\bar{x})$.

Exact views assumption, EVA. The extention of view contains all and only tuples derived from database. Both sound and complete assumptions. $E_{V_i} = Q_{\mathcal{G}}^{V_i}(\mathcal{D})$ or, in other words, $\forall \bar{x}\ s(\bar{x}) \ \leftrightarrow\ q_{\mathcal{G}}^{s}(\bar{x})$.

Definition 5 *A tuple t is* a certain answer *for a query Q in respect to DIS $\mathcal{I}$ if t is an answer to Q for every legal global database of $\mathcal{I}$. This term is synonym to 'a sceptical answer' term used in AI community.*
A tuple t is a possible answer for a query Q in respect to DIS $\mathcal{I}$, if t is an answer to Q for at least one legal global database of $\mathcal{I}$. "A possible answer" is a synonym for "a credulous answer" in AI.

Clearly there exist databases providing different answers under EVA and SVA,

Example 1. A data integration system with global schema containing one relation $student(name, supervisor)$ and two data sources with data mapping $student_1(X) : -student(X, Y)$ and $professor_2(Y) : -student(X, Y)$. Data source $student_1$ contains data $\{\langle john\rangle\}$, and data source $professor_2$ contains data $\{\langle magda\rangle\}$.
Then for a query Q(X,Y) :- student(X,Y), the query answer under EVA is $\{\langle john, magda\rangle\}$, but the query answer under SVA is an empty set.

1.2 Contributions and related work

A problem of query answering under Exact View Assumption was analyzed in [Abiteboul & Duschka, 1998]. In that article it was proofed that the problem is coNP-hard for conjunctive queries and views defined as conjunctive queries. The problems is in coNP for view as union of conjunctive queries and DATALOG-defined queries.a The description of the algorithm given in that article is very incomplete and lacks many important details, actually, instead of algorithm, the example of algorithm running for simple database is given.In this paper we provide a detailed description of the algorithm and another definition of semantic for it. Our algorithm is based on a chase procedure, which simplifies proof techniques and gives complexity results for the algorithm . Reduction to SVA query answering with additional constraints simplifies 'high-level' reasoning problems like 1) composition of mappings, which are of interest for peer-to-peer community and 2) query containment, which is important for query optimization.
In [Grahne & Mendelzon, 1999] an algorithm based on tableaux representation of a global database is given. The algorithm, described in that article handles only conjunctive queries and views. Our and [Abiteboul & Duschka, 1998] algorithms handle positive (disjunctive) queries. In case of Exact Views Assumption algorithm [Grahne & Mendelzon, 1999] provides only possible answers instead of certain.
[Flesca & Greco, 2001] suggests algorithm for query answering in the presence of negative goals (both a query and a view definitions may contain negative atoms). They

propose a method of a query answering, which works only under Exact View Assumption and contains some specific features of EVA like derivation of negative information, but this fact is not explicitly mentioned in the article. In case of algorithm of [Flesca & Greco, 2001] not all information is extracted in general case (for Example 1 algorithm provides empty answer)

In [Imielinski & Lipski, 1983] a problem of "inverting relational expressions", which is equivalent to the problem of reconstructing global database is considered. The algorithm for reconstructing database instance representing all legal databases under OWA is given. The problem of reconstructing a database instance under EVA, which is in the original article called as CWA(Closed World Assumption) is defined. In [Fagin *et al.*, 2003] a problem of "data exchange" is considered. [Fagin *et al.*, 2003] construct "an universal solution" which is conditional table representation of all databases legal under SVA.

In [Bravo & Bertossi, 2004] an answer set approach is proposed for query answering under EVA. Closed World assumption is implemented as denial constraints in logic program. Proposed approach is hard to extend for disjunctive queries because it construct models using new additional null values. The predefined number of null values makes results of [Bravo & Bertossi, 2004] valid only for conjunctive queries.

2 A representation of legal databases by conditional tables

In this section we use such tool as *conditional tables*[Imielienski & Witold Lipski, 1984][Abiteboul, Hull, & Vianu, 1995] to represent a set of databases. Conditional tables are relations with marked null values and local (at the level of a tuple) and global (relational level) conditions. It is powerful enough tool to represent disjunctive and some kinds of negative information.

Let us try to reduce a problem of query answering under EVA to query answering under SVA with some special conditions. In general, query answering under EVA is not reducible to query answering under SVA. It can be proofed by 1) complexity arguments, query answering under EVA is coNP-complete, but query answering under SVA is in PTIME in data complexity; or by 2) monotonicity arguments: query answering under EVA is not monotonic in respect to extension of data sources, while query answering under SVA is monotonic.

We introduce *conjunctive implication constraints*, which are of them form

$$r_1(\boldsymbol{x_1}), r_2(\boldsymbol{x_2}), ..., r_n(\boldsymbol{x_n}) \subseteq r(\boldsymbol{y})$$

$$I \models r_1(\boldsymbol{x_1}), r_2(\boldsymbol{x_2}), ..., r_n(\boldsymbol{x_n}) \subseteq r(\boldsymbol{y}) \;\; iff \; I(r_1)(\boldsymbol{x_1}) I(r_2)(\boldsymbol{x_2}) \cdots I(r_n)(\boldsymbol{x_n}) \subseteq I(r)(\boldsymbol{y})$$

Definition 6 *A sound view analog of DIS $\mathcal{I}$ is a DIS denoted as $SVA(\mathcal{I})$, constructed by the following procedure:*

- *copy to $SVA(\mathcal{I})$ all global and local schemata from $\mathcal{I}$, with all view definitions and view extensions*

– *for each view V under EVA of $\mathcal{I}$, create in the global schema of $SVA(\mathcal{I})$ a relation symbol r_V and in mapping schema of $SVA(\mathcal{I})$, create a LAV mapping $V(X) \subset r_V(X)$*
– *for each view V under EVA, for a view definition $V(X) = r_1(X), ..., r_n(X)$ of $\mathcal{I}$, create in the global schema of $SVA(\mathcal{I})$ an integrity constraint $r_V(X) \supset r_1(X), ..., r_n(X)$*

Example 2. Continue example 1. A transformed DIS is

$$
\begin{array}{rl}
Global\ schema: & student(X,Y), student_1^R(X), professor_2^R(Y) \\
Global\ IC: & student(X,Y) \subseteq student_1^R(X), student(X,Y) \subseteq professor_2^R(Y) \\
Local\ schemata: & student_1(X), professor_2(Y), student_1^A(X), professor_2^A(Y) \\
Mappings: & student_1(X) : -student(X,Y), professor_2(Y) : -student(X,Y) \\
 & student_1^A(X) : -student_1^R(X), professor_2^A(Y) : -professor_2^R(Y) \\
Data: & student_1 = < john >, student_1^A = < john >, \\
 & professor_2 = < magda >, professor_2^A = < magda >
\end{array}
$$

Definition 7 *A database D is legal in respect to DIS $\mathcal{I}$ with conjunctive implication dependencies, if 1) it is legal in respect to DIS $\mathcal{I}$ without constraints; 2) for every relation symbol r contained at the head of on the constraints an extension of in D is equal to related view extention*

Remark 1 *This definition of database legal in respect to mapping and integrity constraints is like definition provided in [Cali, Lembo, & Rosati, 2003], with the only additional condition that databases extending relations which are in the heads of integrity constraints formula are not considered as legal.*

Lemma 1 *For any DIS $\mathcal{I}$ its set of legal global instances is co-initially equivalent to a set of legal global instances represented its sound view analog $SVA(\mathcal{I})$*

Remark 2 *As in [Imielienski & Witold Lipski, 1984], two sets of databases are co-initially equivalent, if they have the same set of minimal (in respect to set inclusion) elements*

So, now we have to solve a problem: once we have a DIS under SVA and integrity constraints, how we can construct a representation of all possible global instances.

Assume that we are given a set of relation symbols $\mathcal{R}$, a set of conjunctive implication integrity constraints $\mathcal{IC}$, such that all relation symbols in $\mathcal{IC}$ belongs to $\mathcal{R}$, and an incomplete database instance *D* which represents all databases legal under SVA. We assume that as incomplete database formalism conditional tables of [Imielienski & Witold Lipski, 1984] (with marked nulls and conditions) are used. As such database instance an *universal solution* from [Fagin *et al.*, 2003] can be taken.

Definition 8 An exact view chase *is called the described below procedure. A result of application of an exact view chase to a database D (a set of conditional tables) is denoted by Ch(D).*

A chase: for each integrity constraint $r \supset \pi(\sigma(r_1 \bowtie \cdots \bowtie r_n))$, do (till the end of the chase description we use a view instead of integrity constraint language):

Each view definition is represented in a canonical form $v_i = \pi_{\overline{X}_i}\sigma_{F_i}(R_{i_1} \bowtie ... \bowtie R_{i_k})$. For each view v_i the sequence of actions $\bowtie -step \rightarrow \sigma - step \rightarrow \pi - step$ is performed.

Definition 9 *Two values x and y are called unifiable, iff both of them are the same constants or at least on of them is a variable. Tuples $t_1 \in R_i, and t_2 \in R_j$ are called unifiable in respect to $\bowtie$-expression $R_i \bowtie_G R_j$, if for each condition $X_1 = X_2 \in G$, values $\pi_{X_1}t_1$ and $\pi_{X_2}t_2$ are unifiable. A tuple t_1 is unifiable with a tuple t_2 in respect to π-expession $\pi_{\overline{X}}$, if each projects attribute of t is unifiable with adequate attribute of t_1. A tuple t_1 is unifiable with tuple t_2 in respect to $\pi\sigma R_1 \bowtie ... \bowtie R_n$ expression if*

Definition 10 *A tuple $t \in R$ is satisfiable by σ_F-expression, if for each attribute X of R, which is in F, $\pi_X t$ is unifiable with F condition.*

A $\bowtie$**-step**.

Input: a database instance and $\bowtie$ expression.

Create R_i^c as a relation representing maximal view extention possible to get from a database instance I. For any set of tuples $t_1 \in R_1, ..., I_m \in R_m$ if they are unifiable in respect to $\bowtie$ $condition$, create a tuple t. Associate with a new tuple t, condition $c_t^{\bowtie} = c_{t_1} \wedge ... \wedge c_{t_m} \bigwedge(x_i = x_j)$, whenever x_i and x_j participate in a join.

A σ**-step**.

Input: a table R_i^c and σ-expression.

With each tuple of R_i^c a view associate σ-condition as a set of equality conditions for its attributes. A new condition c_t^{σ} for a tuple t is $c_t^{\bowtie} \wedge c_F$.

A π**-step**.

Input. A table R_i^c, π-expression and view extention E_i

For each tuple $t \in R_i^c$ assume the projection $\pi_{\overline{X}R_I^c}$ has m variables $x_{i_1}, ..., x_{i_m}$. Then create a condition for a tuple t as the following $c_t = c_t^{\pi} \wedge c_t^{\sigma}$, where

$$c_t^{\pi} = \bigvee_{t_e \in E, t_e \leq t} \pi_{\overline{X}}t = t_e, \tag{1}$$

,

Then in a relation R_i^c, for each tuple t, a condition c_t is replaced by $\neg c_t = \neg c_t^{\pi} \vee \neg c_t^{\sigma} \vee \neg c_t^{\bowtie}$.

Then condition $\neg c_t$ is distributed to original database instance by the following procedure. To each original tuple a set of atoms which contains original variebles is distributed.

Assume the projection is the one column projection $v = \pi_X R$. Let $\mathcal{X} = x_1, ..., x_m$ is a set of variables in projected attribute of database instance, abd $\mathcal{A} = a_1, ..., a_n$ is a set of constants in a view extention.

If $\mathcal{X} = \emptyset$ and $\pi_X(R) \supset \mathcal{A}$, then **inconsistent**.

If $m = n$, then create a global condition (of the size $O(m^2 \times n)$) $(x_1 = a_1 \vee ... \vee x_m = a_n) \wedge ... \wedge (x_1 = a_n \vee ... \vee x_m = a_n)$.

If $m < n$, then for each $i \in \{n - m, ..., n\}$, create a new variable x_i, and for each tuple $\langle \overline{a} x \overline{b} \rangle$ unifiable with view, create a new tuple $\langle \overline{a} x_i \overline{b} \rangle$. Disjoint all tuples for a variable x_i by disjoint condition. Then apply π**-step** as for $m = n$.

In case when the projection is the multicolumn projection $v = \pi_{\overline{X}} R$ then use step is done over tuples of varieables and vectors. Denote each tuple t_i from a view as a variable $\overline{t^i}$, and each unifiable with some tuple from view a tuple from database instance as $\overline{x^j}$. Then perform chase as for one attribute projection case using new variables. Finally rewrite

$\overline{x^j} = \overline{t^i} \rightarrow (x_1^j = t_1^i \wedge ... \wedge x_k^j = t_k^i)$

If for a tuple $t \in e$, unifiable tuple t in database instance does not exists, then **inconsistent**.

For each tuple $t \in R_i^c$, check if $\pi_{\overline{X}} R_i^c \subseteq E_i$. If not then **inconsistent**.

Theorem 2 *A chase will always terminate for any conditional table D and a set of constraints IC. A result of a chase is a conditional table D' which represents a full subset of database instances represented by D, which satisfy IC.*

Corollary 3 *For monotone queries a set of certain answers in respect to DIS $\mathcal{I}$ will be equivalent to a set of certain answers in respect to a database instance represented by $Ch(SVA(\mathcal{I}))$*

Theorem 4 *A size of $Ch(SVA(\mathcal{I}))$is polynomial in respect to a size of $\mathcal{I}$ and chase procedure is PTIME in data complexity*

And then using results [Grahne, 1989] about complexity of certain answering using conditional tables.

Corollary 5 *The answering is in NP for DIS under Exact View assumption*

3 Query Answering using Logical Programming paradigm

There are two paradigms in methods of data integration: query answering and query rewriting. **Query answering** assumes that it is possible to retrieve all data from data sources into one global database, then process them by some algorithm like our chase, and finally get an answer to the query. Another method is **a query rewriting** is a reformulation of a query with predicates in a global database into a query with predicates of local databases.

Definition 11 *A view operator $\mathcal{V}$ for DIS $\mathcal{I}$ is defined in the following way. The domain of $\mathcal{V}$ is a set of databases with schema of the global database of $\mathcal{I}$. The range of $\mathcal{V}$ is a set of databases with a schema as union of all schemata of views in $\mathcal{I}$. $\mathcal{V}(D_1) = D_2$, if D_2 is a result of application of queries of the mappings of $\mathcal{I}$ to D_1*
For DIS $\mathcal{I}$, as $E(\mathcal{I})$ we denote an union of extensions of views in $\mathcal{I}$.

Lemma 6 *Let a database D is an union of all legal databases for DIS $\mathcal{I}$. A tuple t is a certain iff* $E(\mathcal{I}) \setminus \mathcal{V}(D \setminus \{t\}) \neq \emptyset$

Remark 3 *In other words, a tuple t is a certain iff after removal of it from a database which is union of all legal databases, some tuples in computed view extensions will be lost in respect to original extensions*

Remark 4 *It is crucial that a tuple is deleted from an union of all databases which are legal under EVA. In case if all legal databases for SVA are taken (they are easy to construct), then lemma is not valid.*

Assuming that a logical program P_{poss} which answers all possible answers, it is possible to construct a program P_{cert} which selects a certain answers from possible. A P_{cert} is not given here, because of strong limitations on a length of the article.

4 Tractable instances of Query Answering under EVA problem

As it was proofed by [Abiteboul & Duschka, 1998] the Query Answering under EVA is NP-complete in data complexity problems. It makes the problem in general intractable. It would be interesting to find tractable cases. To do this we try to reduce a query answering to uniform Constraint Satisfaction Problem [Kolaitis & Vardi, 1998], for which a number of tractable cases is known. In our work with homomorphism based definition of CSP as in [Kolaitis & Vardi, 1998]
We use a conditional table representation of legal databases received by applying a chase to universal solution of [Fagin *et al.*, 2003]. At this part we are interested in answering conjunctive queries without built-in predicates.
As a domain of CSP problem denoted as $\mathcal{D}\wr\Updownarrow$ we use a set consisting of active domain of all view enriched with one additional constant. $\mathcal{D}\wr\Updownarrow = adom(\mathcal{V}) \cup$ *null*. Let denote *D* a set of conditional tables, which are result of algorithm in ch. "A representation ...". As $\mathcal{V}\dashv\nabla$, we denote a set of variables of *D*.

Lemma 7 *For a DIS with query language is conjunctive queries without built-in predicates and the same view definition language, a set of certain answers for any query is equal to a set of certain answers in respect to a set of databases, which domain $\mathcal{D}\wr\Updownarrow$ as defined about satisfying conditional table.*

At next step we are constructing a CSP which finds all possible assignments to variables which satisfy conditional tables . A condition for variable assignment will be a conjunction of all global and local conditions in *D*, which in conjunctive normal formal is a conjunction of disjuncts of atoms in the form $X\ op\ Y$, where X is an variable from $\mathcal{V}\dashv\nabla$, *op* is "=" or "$\neq$", and Y is a variable or constant. the construction is quite straitforward and in the spirit of [Kolaitis, 2003]. We are constructing two structures $\mathcal{A}$ and $\mathcal{B}$ needed to define a CSP.

1. for each atom of the form $X \neq a$, transform it into the form $X = b_1 \vee X = b_2 \vee \cdots \vee X = b_n$ where $b_1, \cdots b_n$ are all values from $\mathcal{D}\wr\Updownarrow \setminus \{a\}$.

2. for each combination of variables $x_1, \cdots, x_n$ used in a disjunct create a relation $r_{x_1 \cdots x_n}$ in $\mathcal{A}$ and $\mathcal{B}$
3. for each disjunct
   (a) put into a relation $r_{x_1 \cdots x_n}$ of a structure $\mathcal{A}$ a tuple of variables $x_1, \cdots, x_n$.
   (b) put into a relation $r_{x_1 \cdots x_n}$ of a structure $\mathcal{B}$ a set of tuples of constants satisfying disjunct (in case if relation is not empty, delete from it all tuples not satisfying current disjunct).
4. define homomorphism requirement

Lemma 8 *For any conditional database D a relevant CSP problem will be polynomial in size of D.*

Using results of [Fagin *et al.*, 2003],

Theorem 9 *A query answering under EVA problem is in PTIME in data complexity if a tree-width of relevant CSP structures is bounded.*

5 Future work

We found a tractable case of query answering using a reduction to CSP problem. Currently tractability condition is expressed as boundness of a tree-width of a graph. Only some simple condition on a database expressed in database language were found. We are trying to find how boundness of a tree-width can be expressed as a restriction on an arity of views, view definitions, etc. Also we are trying to find tractable cases by reducing the query answering problem to other problems with known tractable cases.
We have provided a logical programming paradigm for answering with conjunctive queries and views. We are trying to find an analog for positive queries with built-in predicates.
The problem of query answering with integrity constraints was investigated at [Cali, Lembo, & Rosati, 2003] only under Sound Views Assumption. It would be interesting to investigate an interaction of conjunctive implication constraints with implication and key constraints and develop a theory of query answering under EVA and integrity constraints.
We are interested to apply our results in EVA answering to P2P and DB system, which we developed [Franconi *et al.*, 2004], [Franconi *et al.*, 2003]. Our autoepistemic semantic [Franconi *et al.*, 2003] makes computation complexity of query answering in P2P DB lower (PTIME instead of undecidable for cyclic networks), and we would like to understand complexity of query answering under EVA and cyclic dependencies. This a study of query answering under exact view assumption for a knowledge bases using Description Logic ontologies[Calvanese, Giacomo, & Lenzerini, 2000]; where the situation is more sophisticated; mainly computtaional complexity is higher and the query answering is EXPTIME-complete, and automata techniques are needed. We are going to find a rewriting techniques for answering in Description Logic using some expressive languages capturing EXPTIME data complexity.
Acknowledgment: I am grateful to my PhD supervisor Prof. Enrico Franconi for his support and technical conversations.

References

[Abiteboul & Duschka, 1998] Abiteboul, S., and Duschka, O. M. 1998. Complexity of answering queries using materialized views. In *ACM Symp. on Principles of Database Systems (PODS)*, 254–263.

[Abiteboul, Hull, & Vianu, 1995] Abiteboul, S.; Hull, R.; and Vianu, V. 1995. *Foundations of Databases*. Addison-Wesley.

[Bravo & Bertossi, 2004] Bravo, L., and Bertossi, L. 2004. Disjunctive deductive databases for computing certain and consistent answers to queries from mediated data integration systems. *Journal of Applied Logic*.

[Cali, Lembo, & Rosati, 2003] Cali, A.; Lembo, D.; and Rosati, R. 2003. On the decidability and complexity of query answering over inconsistent and incomplete databases. In *Proc. of the 22nd ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS-2003)*, 260–271.

[Calvanese, Giacomo, & Lenzerini, 2000] Calvanese, D.; Giacomo, G. D.; and Lenzerini, M. 2000. Answering queries using views over description logics knowledge bases. In *AAAI/IAAI*, 386–391.

[Fagin *et al.*, 2003] Fagin, R.; Kolaitis, P.; Miller, R.; and Popa, L. 2003. Data exchange: Semantics and query answering. In *ICDT 2003, Siena, Italy*.

[Flesca & Greco, 2001] Flesca, S., and Greco, S. 2001. Rewriting queries using views. In *Database and Expert Systems Applications, 10th International Conference, DEXA '99*, volume 10, 352–361.

[Franconi *et al.*, 2003] Franconi, E.; Kuper, G.; Lopatenko, A.; and Serafin, L. 2003. A robust logical and computational characterisation of peer-to-peer database system. In *International Workshop On Databases, Information Systems and Peer-to-Peer Computing, collocated with VLDB'03*.

[Franconi *et al.*, 2004] Franconi, E.; Kuper, G.; Lopatenko, A.; and Zaihraeu, I. 2004. A distributed algorithm for robust data sharing and updates in p2p database networks. In *"International Workshop on Peer-to-Peer Computing and Databases" (in conjunction with EDBT 2004)*.

[Grahne & Mendelzon, 1999] Grahne, G., and Mendelzon, A. O. 1999. Tableau techniques for querying information sources through global schemas. *Lecture Notes in Computer Science* 1540:332–347.

[Grahne, 1989] Grahne, G. 1989. Horn tables-an efficient tool for handling incomplete information in databases. In *Proceedings of the eighth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, 75–82. ACM Press.

[Imielienski & Witold Lipski, 1984] Imielienski, T., and Witold Lipski, J. 1984. Incomplete information in relational databases. *Journal of the ACM (JACM)* 31(4):761–791.

[Imielinski & Lipski, 1983] Imielinski, T., and Lipski, W. 1983. Inverting relational expressions - a uniform and natural technique for various database problems. In *Proceedings of the Second ACM SIGACT-SIGMOD Symposium on Principles of Database Systems, March 21-23, 1983, Colony Square Hotel, Atlanta, Georgia*, 305–311. ACM.

[Kolaitis & Vardi, 1998] Kolaitis, P. G., and Vardi, M. Y. 1998. Conjunctive-query containment and constraint satisfaction. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, 205–213. ACM Press.

[Kolaitis, 2003] Kolaitis, P. 2003. Course on constraint satisfaction, complexity, and logic. In *ESSLLI, Vienna, Austria, 2003*.

[Lenzerini, 2002] Lenzerini, M. 2002. Data integration: A theoretical perspective. In Popa, L., ed., *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 233–246.