A Representation of Certain Answers for Views and Queries with Negation

Victor Felea
Department of Computer Science
"Al. I. Cuza" University
Iasi, Romania
Email: felea@infoiasi.ro

***Abstract*—The paper is about databases content processing, namely query processing. Certain answers are very important in the study of the data complexity of the problem of answering queries using materialized views and constitute a semantics of query answers in mediated integration systems. The computing of these answers depends on database and view models, and the case when the negation occurs was not studied. In this paper, we give a representation of certain answer sets where both the query and the views are expressed in conjunctive form with negation. Using this representation, a method to compute the certain answers for the open-world and close-world assumptions is given.**

Keywords–views; queries; negation; certain answers.

I. Introduction

In many data-management applications such as data-integration from different sources, data warehousing, query optimization, the problem of view-based query processing is central. The study of this problem implies the consideration of several main notions as $answering$ and $rewriting$. $Answering$ means the computing the tuples satisfying the query in all databases consistent with the views. $Rewriting$ is a reformulating of the query in terms of the views, and then evaluating the rewriting over the views extensions. In general terms, the problem of rewriting is as follows: given a query Q on a database schema S, expressed in a language $\mathcal{L}$, and a set $\mathcal{V}$ of views on S, can we answer Q using only $\mathcal{V}$?

A lot of results have been reported in the last years and many methods have been studied(see a survey in [1]). One of the approach of view-based query processing problem is the query-answering approach, where so-called certain tuples ([2]) are computed. Certain tuples are the tuples that satisfy the query in all databases consistent with the views, on the basis of the view definitions and the view extensions.

In [2],[3], some aspects and applications of the problem of answering queries using views, and algorithms are presented. Some authors study the problem of view-based query processing in a context where databases are semistructured, and both the queries and the views are expressed as regular path queries in [4],[5]. A tableau technique is used for computing query answers in [6]. In [7], the authors study the complexity of query answering considering key and inclusion dependencies. The problem of answering queries using views, when queries and views are in conjunctive form with arithmetic comparisons, are analyzed in [8].

The structure of the paper is following: in Section II, we discuss the main papers concerning to the computing of certain answer sets, in Section III, we specify the basic definitions and notations used in the paper. In Section IV, we give a representation of certain answer sets in case of open-world assumption. A method to compute certain answer sets in cases of open-world and close-world assumptions is given in Sections V and VI, respectively. The problem of time complexity to compute certain answers is analyzed in Section VII. Finally, Section VIII concludes the paper.

II. State of the Art

In the relational model, a query Q_1 is said to be contained in the query Q_2 if Q_1 produces a subset of the answers of Q_2, for any database. In the context of data integration, we say that Q_1 is contained in Q_2 relative to a set of views $\mathcal{V}$ if, for any set I of instances of $\mathcal{V}$, the certain answers of Q_1 are a subset of the certain answers of Q_2. In [2], the authors study the complexity of computing certain answers in case when views and queries are in conjunctive form, conjunctive form with inequality, non-recursive datalog, datalog or first order formula. In case when the query is expressed in datalog, and does not contain comparison predicate, and the views are in conjunctive form, the set of certain answers can be obtained by so called a query plan, which is a datalog program whose extensional relations are the source relations. More precisely, the maximally contained query plans defined in [9] compute the certain answers of the query [2]. In [10], using certain answers, the authors define $relative\ containment$, which formalizes the notion of query containment relative to the sources that occur to the data-integration system. In [11], the containment of two queries is studied. In [5], the authors study the problem of answering queries using materialized views in the presence of negative atoms in views. By our best knowledge, the problem to compute the certain answer set in case when the negation occurs in views or query was not considered yet in literature.

Let us give a motivated example.

Example 1 Let us consider the relational schema S consisting of $\{COMP, CON, PROD, ITEM\}$,where

$COMP$ represents the companies and has *comp-id* as an identification code of a company, and *comp-name* is the name of the company,

CON represents the contracts between companies, and has

as attributes $con\text{-}id$ the identification code of the contract, $b\text{-}comp$, the beneficiary of the contract $con\text{-}id$, and $f\text{-}comp$ is the supplier of the contract $con\text{-}id$ (the values of $b\text{-}comp$ and $f\text{-}comp$ are company codes),
$PROD$ represents the products, and has $prod\text{-}id$ as the identification code of a product, and $prod\text{-}name$ the product name,
$ITEM$ represents items specified by the contracts, and has as attributes: $con\text{-}id1$, the code of a contract, $prod\text{-}id1$, the code of a product. We consider the following view: find contracts x_1, companies x_2, and products x_3 such that x_2 is the beneficiary company of x_1 and the product x_3 occurs as item of the contract x_1, and there exists a contract y_2 such that the product x_3 does not occur in the contract y_2.

Using the schema $\mathcal{S}$, we can expressed this view definition as:

$$V(x_1, x_2, x_3) : -CON(x_1, x_2, y_1), CON(y_2, y_3, x_2),$$
$$PROD(x_3, y_4), ITEM(x_1, x_3), \neg ITEM(y_2, x_3),$$

where the character ',' between two literals means logical conjunction. Let us consider the query: find all companies z such that there exist two contracts t_1 and t_3, where one of them contains z as the beneficiary and another as supplier and there exists a products t_5 such that one or another from contracts t_1 and t_3 does not contain the product t_5. We expressed this query as follows:

$$Q : q(z) : -CON(t_1, z, t_2), CON(t_3, t_4, z),$$
$$PROD(t_5, t_6), (\neg ITEM(t_1, t_5) \vee \neg ITEM(t_3, t_5))$$

It is clear that the query Q is equivalent with a union of queries in conjunctive form. Let I be an extension of V, where $I = \{\overline{w}_1, \overline{w}_2\}$, and $\overline{w}_1 = (1, 'S2', 'P2')$, and $\overline{w}_2 = (2, 'S3', 'P3')$. We are interested to compute the certain answers corresponding to I, V and Q.

In this paper, we compute the certain answer sets in two cases: under the open-world assumption (OWA) and under the closed-world assumption (CWA) in case when views are in conjunctive form, and query is a union of conjunctive form, and both can contain negative literals.

III. Basic Definitions and Notations

Let Dom be a countable infinite domain for databases. A view definition has the following form:

$$V(\overline{x}) : -R_1(\overline{u}_1), \ldots, R_k(\overline{u}_k), \neg R_{k+1}(\overline{u}_{k+1}), \ldots,$$
$$\neg R_{k+p}(\overline{u}_{k+p}), \tag{1}$$

where $\overline{x}$ is a vector of variables. These variables are called free in the view V, The vectors $\overline{u}_i$ consists of variables or constants, $1 \le i \le k+p$. All constants are considered in Dom. There are two restrictions about variables or constants that occur in the view. The first one is: each variable that occurs in $\overline{x}$, it must appear also in at least a vector $\overline{u}_i$, $1 \le i \le k$, that means it also appears in the positive part of the view definition. This is called the safe property of the view. The second one is: each variable or constant that occurs in the negated part of the view definition, must occur in its positive part. This property is called as safeness property of negation. The symbols R_i are relational symbols, $1 \le i \le k+p$. All variables from $\overline{u}_i$, $1 \le i \le k+p$, that are different from variables from $\overline{x}$, are called existentially quantified variables. Let us denote by $f_V(\overline{x}, \overline{y})$ the right part of the view definition V. A query Q in conjunction form with negation has a similar form as view definition. Let us denote by $q(\overline{z})$ the head of the query, and $f_q(\overline{z}, \overline{t})$, the right hand part of the query, where $\overline{t}$ denote all existentially quantified variables from the query. In an integration system the views are called sources. If sources have non-relational data models, we can use $wrappers$ [12] to create relational view of data.
In the following definition we present the notion of certain answer in two cases: (OWA) and (CWA).

Definition 1: Let Q be a query and $\mathcal{V} = \{V_1, \ldots, V_m\}$ be a set of view definitions over the database schema $\mathcal{S} = \{R_1, \ldots, R_s\}$ (all relational symbols from $\mathcal{S}$ are used in at least V_i). Let $\overline{w}_i$ be an extension of the view definition V_i, for each $i, 1 \le i \le m$. Let $I = \{\overline{w}_1, \ldots, \overline{w}_m\}$. The tuple t is a certain answer for I, $\mathcal{V}$, and Q under OWA if $t \in Q(D)$, for all databases D defined on Dom such that $I \subseteq \mathcal{V}(D)$. The tuple t is a certain answer for I, $\mathcal{V}$ and Q under CWA if $t \in Q(D)$ for all databases D defined on Dom such that $I = \mathcal{V}(D)$.
In an intuitive sense, a tuple is a certain answer of the query Q, if it is an answer for any of the possible database instances, which are consistent with the given extensions of the views. Concerning the number of view definitions, and the number of extensions of the views from $\mathcal{V}$ we distinguish three cases:
(I)$\mathcal{V}$ consists of a single view definition denoted V, and I consists of m extensions of V, denoted $\overline{w}_1, \ldots, \overline{w}_m$.
(II)$\mathcal{V}$ consists of multiple definitions of a view V, denoted $V(\overline{x}) : -f_V^i(\overline{x}, \overline{y})$, $1 \le i \le h$, and I consists of m extensions of the view definitions of V.
(III) $\mathcal{V}$ consists of multiple view definitions of the views $V_1, \ldots, V_q$ and I consists of m extensions, an extension corresponds to a view definition of any view V_i, $1 \le i \le q$.

For the sake of the presentation, let us consider the case (I), the approaches of the cases differ only in notation.

In the case of the open-world assumption, we express the relation $I \subseteq \mathcal{V}(D)$ that it is equivalent to: $\overline{w}_i \in V(D)$, for each $i, 1 \le i \le m$, where D is a database defined on Dom. Then there exists a mapping ν from the set of all variables from $\overline{x}$ and $\overline{y}$ into Dom such that the following relations yield:

$$\nu pos(f_V(\overline{x}, \overline{y})) \subseteq D, \nu neg(f_V(\overline{x}, \overline{y})) \cap D = \emptyset, \nu\overline{x} = \overline{w}_i, \tag{2}$$

where $\nu(c) = c$, for a constant c,
$pos(f_V(\overline{x}, \overline{y}))$= $\{R_1(\overline{u}_1), \ldots, R_k(\overline{u}_k)\}$ and
$neg(f_V(\overline{x}, \overline{y}))$= $\{R_{k+1}(\overline{u}_{k+1}), \ldots, R_{k+p}(\overline{u}_{k+p})\}$
We denote by $Rel(f_V(\overline{x}, \overline{y}))$, the set of all relational symbols that occur in the conjunction $f_V(\overline{x}, \overline{y}))$. For a mapping ν having the property from (2), we denote by $f_V(\overline{w}_i, \overline{y})$ the

result of replacing all free variables x' from $\overline{x}$ with $\nu(x')$. For two different vectors $\overline{w}_i$ and $\overline{w}_j$,the existentially quantified variables of type y are independent, hence we take the sets of variables for $\overline{y}$ disjoint. So, we denote by $f_V(\overline{w}_i,\overline{y}_i)$ orf_i, the expression $f_V(\overline{x},\overline{y})$, where $\overline{y}$ is replaced by $\overline{y}_i$ and the set of all variables from $\overline{y}_i$ is disjoint from the set from $\overline{y}_j$, that means $\overline{y}_i \cap \overline{y}_j = \emptyset$ for all $i,j, 1 \le i \ne j \le m$. Let us denote by C the set of all elements from Dom, that appear in the vectors $\overline{w}_i, 1 \le i \le m$ and in $f_V(\overline{x},\overline{y})$. Let Y be the set of all variables from $\overline{y}_1,\ldots,\overline{y}_m$. Let π be a partition of the set $C \cup Y$, and $Class_\pi$ the set of the classes defined by the partition π. We denote by $\equiv_\pi$ the congruence relation defined by the partition π, namely: we have $t \equiv_\pi t'$ if there exists a set M from $Class_\pi$ such that $t,t' \in M$. In the paper we only need partitions with the property: for two different constants c and c',we have $c \not\equiv_\pi c'$. These partitions are called C-partitions. It is clear that for a C-partition a class contains at most a constant. For a partition π on $C \cup Y$, we consider a mapping from $C \cup Y$ into $Class_\pi$ denoted φ_π called the canonical onto mapping and defined by $\varphi_\pi(t) = [t]_\pi$, where $[t]_\pi$ means the class that contains t. The mapping φ_π is extended naturally to a vector $\overline{w}' = (t_1,\ldots,t_r)$ on $C \cup Y$ by $\varphi_\pi(\overline{w}')$=$(\varphi_\pi(t_1),\ldots,\varphi_\pi(t_r))$. For an atom $R(\overline{w}')$, we consider $\varphi_\pi(R(\overline{w}'))$=$R(\varphi_\pi(\overline{w}'))$. For a set of atoms S having the form $R(\overline{w}')$, we define $\varphi_\pi(S)$=$\{\varphi_\pi(R(\overline{w}'))|R(\overline{w}') \in S\}$. Associated to a C-partition π, we define two databases having elements from $Class_\pi$, and denoted T_π^{min}, T_π^{max}, in the following manner:

$$T_\pi^{min} = \cup_{i=1}^{i=m} \varphi_\pi pos(f_V(\overline{w}_i,\overline{y}_i)) \quad (3)$$

$$T_\pi^{max} = \{\varphi_\pi R(\overline{w}) | R \in Rel(f_V(\overline{x},\overline{y})), \overline{w} \text{ on } C \cup Y\} - \cup_{i=1}^{i=m} \varphi_\pi neg(f_V(\overline{w}_i,\overline{y}_i)) \quad (4)$$

We denote by $\mathcal{M}_\pi$ the set of all databases between T_π^{min} and T_π^{max},i.e.,:

$$\mathcal{M}_\pi = \{T | T_\pi^{min} \subseteq T \subseteq T_\pi^{max}\}$$

For a conjunction of literals f_i, we need to consider a formula denoted $\phi(f_i)$, whose basic elements have the form $(t_i \ne t_j)$, where t_i and t_j are elements from $C \cup Y$. Let f_i having the form: $f_i = R_1(\overline{z}_1),\ldots,R_k(\overline{z}_k)$, $\neg R_{k+1}(\overline{z}_{k+1}),\ldots,$ $\neg R_{k+p}(\overline{z}_{k+p})$. Let $R_{k+j}(\overline{z}_{k+j})$ be an atom that occurs in the negated part of f_i. Let us consider the case when R_{k+j} occurs in the positive part of f_i, with the indexes $\alpha_1,\ldots,\alpha_q$, that means we have: R_{k+j}=R_{α_1}=...=R_{α_q}, and $R_{k+j} \ne R_\beta$ for each $\beta \in \{1,2,\ldots,k\} - \{\alpha_1,\ldots,\alpha_q\}$.
Associated to the atom $R_{k+j}(\overline{z}_{k+j})$, we consider the formula denoted ϕ_i^j and defined as follows:

$$\phi_i^j = (\overline{z}_{k+j} \ne \overline{z}_{\alpha_1}) \wedge \ldots \wedge (\overline{z}_{k+j} \ne \overline{z}_{\alpha_q})$$

where the expression $(\overline{z}_l \ne \overline{z}_s)$ denotes the following disjunction: $(t_l^1 \ne t_s^1) \vee \ldots \vee (t_l^r \ne t_s^r)$, with $\overline{z}_l = (t_l^1,\ldots,t_l^r)$, $\overline{z}_s = (t_s^1,\ldots,t_s^r)$. In case when R_{k+j} does not occur in the positive part of f_i, then we consider $\phi_i^j = TRUE$. The formula $\phi(f_i)$is defined as the conjunction of all formulas ϕ_i^j, for $1 \le j \le p$, that means $\phi(f_i) = \phi_i^1 \wedge \ldots \wedge \phi_i^p$. Now, let us consider the conjunction of all formulas $\phi(f_i)$, $1 \le i \le m$, denoted $\phi(f_V)$, that means: $\phi(f_V) = \phi(f_1) \wedge \ldots \wedge \phi(f_m)$.
Let us consider an example concerning these formulas.

Example 2 Let V and I be defined in Example 1. To be short, let us rewrite the predicates CON, $PROD$, $ITEM$ by R_1, R_2, R_3, respectively. We have: $V(x_1,x_2,x_3)$: $-$ $R_1(x_1,x_2,y_1)$, $R_1(y_2,y_3,x_2)$, $R_2(x_3,x_4)$, $R_3(x_1,x_3)$, $\neg R_3(y_2,x_3)$. f_1=$f_V(\overline{w}_1,\overline{y})$=$R_1(1,'S2',y_1)$, $R_1(y_2,y_3,'S2')$, $R_2('P2',y_4)$, $R_3(1,'P2')$, $\neg R_3(y_2,'P2')$. The formula $\phi(f_1)$ corresponds to the atom $R_3(y_2,'P2')$ and $\phi(f_1) = (y_2 \ne 1)$. For the formula f_2 we take y-variables as: $y_5, y_6, y_7.y_8$. We obtain $\phi(f_2) = (y_6 \ne 2)$. Finally, $\phi(f_V) = (y_2 \ne 1) \wedge (y_6 \ne 2)$.

We remark that the formula $\phi(f_V)$ express the satisfiability property of the formula f_V. In the following we define formally the logic value of a formula for a C-partition.

Definition 2: Let π be a C-partition defined on $C \cup Y$ and $\phi(f_V)$ the formula constructed for f_V, as we have mentioned above. We define the logic value of $\phi(f_V)$ for π, denoted $\pi(\phi(f_V))$, as follows:
(i) If $\phi = (t \ne t')$, where t and $t' \in C \cup Y$, then $\pi(\phi) = TRUE$ if there is no class E from $Class_\pi$ such that $t,t' \in E$, i.e., $[t]_\pi \ne [t']_\pi$.
(ii) $\pi(\phi_1 \wedge \phi_2) = \pi(\phi_1) \wedge \pi(\phi_2)$, $\pi(\phi_1 \vee \phi_2) = \pi(\phi_1) \vee \pi(\phi_2)$.

We remark that for a C-partition π , we have $\pi(\phi(f_V)) = TRUE$ if and only if $\varphi_\pi(f_i(\overline{w}_i,\overline{y}_i))$ is satisfiable ([12]) for each $i, 1 \le i \le m$, where φ_π is the canonical onto mapping corresponding to π. For a database D defined on Dom, we consider $val(D)$ the set of all values that occur in the atoms of D. Formally,
$val(D) = \{v | \exists R(\overline{w}) \in D, v \text{ is a component of } \overline{w}\}$.

Let us denote by $f_1 \cdot f_2$ the composition of the mappings f_1 and f_2, where $(f_1 \cdot f_2)(x) = f_1(f_2(x))$.

IV. A Representation of Certain Answer Sets under OWA

Firstly, we point out a proposition about two databases that are in a particular relation.

Proposition 1: Let D' and D be two databases over the schema $\mathcal{S}$ such that $D' \subseteq D$ and for each atom $R(\overline{w}) \in D - D'$, there exists a component v belonging to $\overline{w}$ such that $v \notin val(D')$. Then for each query Q having $Rel(Q) \subseteq Rel(\mathcal{V})$, we have $Q(D') \subseteq Q(D)$.

Proof: Let $\overline{u}$ be from $Q(D')$. This implies there exists a substitution θ from the variables of Q into Dom such that:

$$\theta pos(Q) \subseteq D', \theta neg(Q) \cap D' = \emptyset \text{ and } \theta\overline{z} = \overline{u}, \quad (5)$$

where the head of the query Q is $q(\overline{z})$. The hypothesis, the second statement from (5), and the safeness property of negation imply $\theta neg(Q) \cap D = \emptyset$, hence $\overline{u} \in Q(D)$. ∎
The following theorem points out some properties concerning the C-partitions from $Part(C \cup Y)$ and some sous-databases

of a database D.

Theorem 1: Let $\mathcal{V}$ be a set of view definitions on the schema $\mathcal{S}$, I an extension of $\mathcal{V}$, and D a database on Dom such that $I \subseteq \mathcal{V}(D)$, and $Rel(D) \subseteq Rel(\mathcal{V})$. Let $\phi(f_V)$ be the formula constructed for I and $\mathcal{V}$. There exist a C-partition π from $Part(C \cup Y)$ such that $\pi(\phi(f_V)) = TRUE$, a database D' such that $D' \subseteq D$, and a bijective mapping ψ_π from $Class_\pi$ into $val(D')$ having the following properties:
(i) For each atom $R(\overline{w})$ from $D - D'$, the vector $\overline{w}$ has at least a component t such that $t \notin val(D')$.
(ii) For any query Q such that $Rel(Q) \subseteq Rel(\mathcal{V})$, we have: $Q(D') \subseteq Q(D)$ and $\psi_\pi^{-1} Q(D') = Q(\psi_\pi^{-1}(D'))$.
(iii) Let $T = \psi_\pi^{-1}(D')$. We have $T \in \mathcal{M}_\pi$.
(iv) $I \subseteq \mathcal{V}(D')$.

Proof: Let $\mathcal{V}, I, D$ as in the hypothesis of the Theorem such that $I \subseteq \mathcal{V}(D)$. This inequality is equivalent to the statement: $V(\overline{w}_i) \in \mathcal{V}(D)$ for each $i, 1 \leq i \leq m$. This means:

$$(\exists \tau_i)(\tau_i : C \cup \overline{y}_i -> Dom)[D \models \tau_i f(\overline{w}_i, \overline{y}_i)], 1 \leq i \leq m \tag{6}$$

Moreover, we assumed that $\tau_i(c) = c$ for each element c from C. Let us emphasize the atoms from f_i:

$$f(\overline{w}_i, \overline{y}_i) = A_1 \wedge \ldots \wedge A_h \wedge \neg A_{h+1} \wedge \ldots \neg A_{h+p} \tag{7}$$

The relation $D \models \tau_i f(\overline{w}_i, \overline{y}_i)$ is equivalent to:

$$\tau_i A_j \in D, 1 \leq j \leq h \text{ and } \tau_i A_{h+l} \notin D, 1 \leq l \leq p \tag{8}$$

Since for $i \neq j$ we have $\overline{y}_i \cap \overline{y}_j = \emptyset$, there exists a mapping τ from $C \cup Y$ into Dom such that $\tau(c) = c$ for each c from C, and $\tau(y_\alpha) = \tau_i(y_\alpha)$, where $y_\alpha \in \overline{y}_i$. Associated to the mapping τ, we define a partition denoted π, and defined as follows: $t \equiv_\pi t'$ if $\tau(t) = \tau(t')$,where $t, t' \in C \cup Y$. Since the statements from (6) are true, it follows that $\pi(\phi(f_V)) = TRUE$. Let V' be the set of all values from $\tau(C \cup Y)$. Let D' be the database defined as follows:

$$D' = \{R(\overline{w}) | R(\overline{w}) \in D, R \in Rel(V) \text{ and } \overline{w} \text{ contains only values from } V'\} \tag{9}$$

It is clear that $val(D') = V'$, and the databases D, D' satisfy the statement (i) from the Theorem. Using this statement and Proposition 1, we obtain $Q(D') \subseteq Q(D)$ for each query Q.

Now, let us define a bijective mapping denoted ψ_π from $Class_\pi$ into $val(D')$, as follows; $\psi_\pi([t]_\pi) = \tau(t)$. Let us denote by ψ_π^{-1} the inverse mapping of ψ_π. Let us show the second part of the condition (ii). Let Q be a query defined on $\mathcal{S}$ and, having the form:

$$Q : q(\overline{z}) : -S_1(\widetilde{w}_1), \ldots, S_l(\widetilde{w}_l), \neg S_{l+1}(\widetilde{w}_{l+1}), \ldots, \neg S_{l+r}(\widetilde{w}_{l+r}) \tag{10}$$

Let $q(\overline{w})$ be from $Q(D')$. There exists a substitution θ from the set of all variables from Q into V' such that the following statements yield:

$$\theta S_i(\widetilde{w}_i) \in D', 1 \leq i \leq l, \theta S_{l+i}(\widetilde{w}_{l+i}) \notin D', 1 \leq i \leq r,$$

$$\theta(\overline{z}) = \overline{w} \tag{11}$$

Since the mapping ψ_π^{-1} is injective, from the relation (11), we get:

$$\psi_\pi^{-1}(\theta S_i(\widetilde{w}_i)) \in \psi_\pi^{-1}(D'), 1 \leq i \leq l, \psi_\pi^{-1}(\theta S_{l+i}(\widetilde{w}_{l+i})) \notin \psi_\pi^{-1}(D'), 1 \leq i \leq r, \psi_\pi^{-1}(\theta(\overline{z})) = \psi_\pi^{-1}(\overline{w}) \tag{12}$$

From the relations (12), we infer the substitution $\theta' = \theta \cdot \psi_\pi^{-1}$ satisfies the relations (11) with θ' instead of θ and $\psi_\pi^{-1}(D')$ instead of D'. This means the following statement is true:

$$\psi_\pi^{-1}(q(\overline{w})) \in Q(\varphi_\pi^{-1}(D')) \tag{13}$$

$$\text{The relation(13)implies } \psi_\pi^{-1}(Q(D')) \subseteq Q(\psi_\pi^{-1}(D')) \tag{14}$$

The inclusion $Q(\psi_\pi^{-1}(D')) \subseteq \psi_\pi^{-1}(Q(D'))$ follows in a similar manner, because ψ_π^{-1} is bijective. Now, let us consider the statement (iii). Let $T = \psi_\pi^{-1}(D')$. Since the mapping τ satisfies the relation $\tau(\cup_{i=1}^{i=m} pos(f(\overline{w}_i, \overline{y}_i))) \subseteq D'$, we obtain the following inclusion:

$$(\tau \cdot \psi_\pi^{-1})(\cup_{i=1}^{i=m} pos(f(\overline{w}_i, \overline{y}_i))) \subseteq \psi_\pi^{-1}(D') \tag{15}$$

On the other hand, we get $\tau \cdot \psi_\pi^{-1} = \varphi_\pi$.

$$\text{Therefore, from (15), we obtain: } T_\pi^{min} \subseteq T \tag{16}$$

Since the relation $\tau(\cup_{i=1}^{i=m} neg(f(\overline{w}_i, \overline{y}_i))) \cap D' = \emptyset$ holds, we obtain:

$$\varphi_\pi(\cup_{i=1}^{i=m} neg(f(\overline{w}_i, \overline{y}_i))) \cap \psi_\pi^{-1}(D') = \emptyset \tag{17}$$

Moreover, we have:

$$\psi_\pi^{-1}(D') \subseteq \{\varphi_\pi R(\overline{w}) | R \in Rel(V), \overline{w} \text{ is on } C \cup Y\} \tag{18}$$

$$\text{The relations (17) and (18) imply } \psi_\pi^{-1}(D') \subseteq T_\pi^{max} \tag{19}$$

From the statements (16) and (19), we obtain $T \in \mathcal{M}_\pi$. The statement (iv) results because the relation (6) is satisfied for the database D'. ■

The following theorem specifies the properties of C-partitions.

Theorem 2: Let π be a C-partition from $Part(C \cup Y)$ such that $\pi(\phi(f_V)) = TRUE$. Let T be an element from $\mathcal{M}_\pi$. For each injective $C - mapping$ ψ from $Class_\pi$ into Dom, we have:
(i) $I \subseteq \mathcal{V}(D')$, where $D' = \psi(T)$.

Proof: We consider the substitution τ from $C \cup Y$ into Dom, defined as: $\tau = \varphi_\pi \cdot \psi$. Let τ_i be the substitution obtained from τ by projection on $C \cup \overline{y}_i$, $1 \leq i \leq m$. We must show that:

$$D' \models \tau_i f(\overline{w}_i, \overline{y}_i), \text{ for each } i, 1 \leq i \leq m \tag{20}$$

Since T is a database from $\mathcal{M}_\pi$, we get for each $i, 1 \leq i \leq m$:

$$\varphi_\pi pos(f(\overline{w}_i, \overline{y}_i)) \subseteq T \text{ and } \varphi_\pi neg(f(\overline{w}_i, \overline{y}_i)) \cap T = \emptyset \tag{21}$$

Applying the mapping ψ to the first relation from (21), we obtain:

$$\tau pos(f(\overline{w}_i, \overline{y}_i)) \subseteq D', \text{ hence } \tau_i pos(f(\overline{w}_i, \overline{y}_i)) \subseteq D' \tag{22}$$

 ISBN:978-1-61208-115-1

Since the mapping ψ is injective, from the second relation from (21), we get:

$$\tau neg(f(\overline{w}_i, \overline{y}_i)) \cap D' = \emptyset, \text{ hence we have:}$$

$$\tau_i neg(f(\overline{w}_i, \overline{y}_i)) \cap D' = \emptyset \quad (23)$$

The statements (22) and (23) imply (20), therefore we have $I \subseteq \mathcal{V}(D')$. ■

In the following we emphasize other property of C-partitions and injective mappings.

Proposition 2: Let π be a C-partition on $C \cup Y$ such that $\pi(\phi(f_V)) = TRUE$. Let D_1 be a database defined on $Class_\pi$, and ψ an injective mapping from $Class_\pi$ into Dom. We have $\psi(Q(D_1)) = Q(\psi(D_1))$, for each query Q expressed as a union of conjunctive form, and having $Rel(Q) \subseteq Rel(V)$.

Proof: Let Q be a query having the form like as in (10), and ψ an injective mapping from $Class_\pi$ into Dom. The answer of Q for T is as follows:

$$Q(T) = \{\theta q(\overline{z}) | \theta pos(Q) \subseteq T \text{ and } \theta neg(Q) \cap T = \emptyset\} \quad (24)$$

From this relation, we get:

$$\psi(Q(T)) = \{(\theta \cdot \psi) q(\overline{z}) | \theta pos(Q) \subseteq T \text{ and}$$
$$\theta neg(Q) \cap T = \emptyset\} \quad (25)$$

Let $\widetilde{u}$ be from $\psi(Q(T))$. There exists θ a mapping from the variables of Q into $Class_\pi$ such that $\widetilde{u} = (\theta \cdot \psi) q(\overline{z})$, and θ satisfies the statements from (25). Since ψ is injective, from these relations, we obtain $(\theta \cdot \psi) pos(Q) \subseteq \psi(T)$ and $(\theta \cdot \psi) neg(Q) \cap \psi(T) = \emptyset$. These imply: $\widetilde{u} = (\theta \cdot \psi) q(\overline{z}) \in Q(\psi(T))$. Therefore, we have obtained $\psi(Q(T)) \subseteq Q(\psi(T))$. The inverse inclusion is inferred similarly. ■

Before we give the theorem about a representation of certain answers, we need to give some further notations. Let $P = \{\pi_1, \ldots, \pi_p\}$ be the set of all C-partitions from $Part(C \cup Y)$ such that $\pi_i(\phi(f_V)) = TRUE$. For a partition π_i, we denote by S_{π_i} the intersection of all answers of the query Q for databases T from $\mathcal{M}_{\pi_i}$, that means: $S_{\pi_i} = \cap\{Q(T) | T \in \mathcal{M}_{\pi_i}\}$. Let $\mathcal{A} = (S_{\pi_1}, \ldots, S_{\pi_p})$. Let ψ_i be an injective mapping from $Class_{\pi_i}$ into Dom, $1 \leq i \leq p$. Let $\mathcal{B}$ be the vector $(\psi_1, \ldots, \psi_p)$ and $Ans(\mathcal{B}) = \cap_{i=1}^{i=p} \psi_i(S_{\pi_i})$. Let $VMapp$ be the set of all vectors having the form $\mathcal{B}$, and $RCertAnsO$ the intersection of all $Ans(\mathcal{B})$ for all $\mathcal{B}$ from $VMapp$, i.e., $RCertAnsO = \cap\{Ans(\mathcal{B}) | \mathcal{B} \in VMapp\}$. Let us denote by $CertAnsO(\mathcal{V}, I, Q)$, the set of all certain answers for $\mathcal{V}$, I, Q. In the following theorem, we give a characterization of this certain answer set.

Theorem 3: Let $\mathcal{V}$, I, Q be a set of view definitions, an instance of $\mathcal{V}$ and a query, respectively. We have $CertAnsO(\mathcal{V}, I, Q)$=$RCertAnsO$.

Proof: Firstly, let $\overline{w}$ be a vector from $CertAnsO(\mathcal{V}, I, Q)$. To show that $\overline{w} \in RCertAnsO$. Let $\mathcal{B}$ be a vector of injective mappings, $\mathcal{B}$= $(\psi_1, \ldots, \psi_p)$, where ψ_i is a mapping from $Class_{\pi_i}$ into Dom. Let T be an element from $\mathcal{M}_{\pi_i}$. Let us denote the database $\psi_i(T)$ by $D_i^{'}$. By Theorem 2, we have $I \subseteq \mathcal{V}(D_i^{'})$. Using the hypothesis, we obtain $\overline{w} \in Q(D_i^{'}) = Q(\psi_i(T)) = \psi_i(Q(T))$. We get $\overline{w} \in \psi_i(S_{\pi_i})$, for each π_i from P and mapping vector $\mathcal{B}$, therefore $\overline{w} \in RCertAnsO$.

Inversely, assume that $\overline{w} \in RCertAnsO$. To show that $\overline{w} \in CertAnsO(\mathcal{V}, I, Q)$. Let D be a database on Dom such that $I \subseteq \mathcal{V}(D)$. We must show that $\overline{w} \in Q(D)$. By the hypothesis, we have $\overline{w} \in Ans(\mathcal{B})$, for each $\mathcal{B}$ from $VMapp$, hence we obtain:

$$\overline{w} \in \psi_i(S_{\pi_i}), \text{ for each } i, 1 \leq i \leq p, \text{ and for each } \mathcal{B}. \quad (26)$$

Using Proposition 2, we get: $\overline{w} \in \psi_i(Q(T))$, for each $T \in \mathcal{M}_{\pi_i}$. Using Theorem 1, we have: there exist a partition π_i, a mapping ψ_{π_i}, and a database D' such that $D' \subseteq D$, $I \subseteq \mathcal{V}(D')$, where ψ_{π_i} is a mapping from $Class_{\pi_i}$ into $val(D')$. In the relation (26) we take ψ_{π_i}, instead of ψ_i. Thus, using Proposition 2, we have: $\overline{w} \in \psi_{\pi_i}(Q(T))$=$Q(\psi_{\pi_i}(T))$=$Q(\psi_{\pi_i}(\psi_{\pi_i}^{-1}(D')))$=$Q(D')$. Since $Q(D') \subseteq Q(D)$, we obtain $\overline{w} \in Q(D)$. ■

V. Certain Answers under OWA

Based on the results of the precedence section, we give in this section a method to construct the set of all certain answers for $\mathcal{V}$, I, Q. Since we considered each constant from C belongs to the domain Dom, we have $C \subseteq Dom$. Moreover, for each π a C-partition on $C \cup Y$, we have $|C| \leq |Class_\pi|$, where $|C|$ denote the cardinality of C. Regarding to the vectors having components from $Class_\pi$, we need to introduce a condition denoted $Cond$ and defined in the following.

Definition 3: Let $\widetilde{w} = (\overline{w}_1, \ldots, \overline{w}_p)$, where $\overline{w}_j \in S_{\pi_j}$, and $\overline{w}_j = (t_{j1}, \ldots, t_{jr})$, $1 \leq j \leq p$. We say that the vector $\widetilde{w}$ satisfies the condition $Cond$ if the following statements yield:
(i) The class t_{ji} contains a constant denoted $c_{\alpha_{ji}}$ for all $j, 1 \leq j \leq p$, and $i, 1 \leq i \leq r$,
(ii) Let $t_{jl} = [c_{\alpha_{jl}}]_{\pi_l}$, $1 \leq l \leq r$, $1 \leq j \leq p$. Then we have: $c_{\alpha_{1l}} = \ldots = c_{\alpha_{pl}}$, for each $l, 1 \leq l \leq r$.

In the following, we point out some properties of the vector $\widetilde{w}$ that satisfy the condition $Cond$ from Definition 3.

Proposition 3: Let $\widetilde{w} = (\overline{w}_1, \ldots, \overline{w}_p)$, where $\overline{w}_j \in S_{\pi_j}$, $1 \leq j \leq p$. We have: the vector $\widetilde{w}$ satisfies the condition $Cond$ if and only if there exists a unique injective $C-mapping$ from $Class_{\pi_j}$ into Dom, denoted ψ_j, for each $j, 1 \leq j \leq p$ such that $\psi_1(\overline{w}_1) = \ldots = \psi_p(\overline{w}_p)$.

Remark 1: Let $\widetilde{w} = (\overline{w}_1, \ldots, \overline{w}_p)$, where $\overline{w}_j \in S_{\pi_j}$, $1 \leq j \leq p$. The vector $\widetilde{w}$ produces a certain answer, denoted $PROD(\widetilde{w})$, under OWA if and only if $\widetilde{w}$ satisfies the condition $Cond$. In this case, we have from Proposition 3, $PROD(\widetilde{w})$=$\psi_1(\overline{w}_1)$.

We can easy construct a procedure that computes the set of all certain answers for $\mathcal{V}$, I, Q.

Example 3 Let us consider I, V and Q as in Example 1. Using the results of Sections IV and V, we obtain that $w = (1)$ and $w = (2)$ are certain answers under OWA.

 ISBN:978-1-61208-115-1

VI. Certain Answers under CWA

In this section, we point out some theorems necessary to represent sets of certain answers under CWA. The proofs of these theorems are similar to that of Theorems 1, 2, 3, therefore they are omitted. Firstly, we need to consider a new notion regarding to a database T defined on $Class_\pi$, where π is a C-partition.

Definition 4: Let $I = \{\overline{w}_1, \ldots, \overline{w}_m\}$, π a C-partition such that $\pi(\phi(f_V)) = TRUE$ and T an element from $\mathcal{M}_\pi$. We say that T is closed with respect to I, if for each substitution θ from the variable set of Q into $Class_\pi$ such that $T \models \theta f_V(\overline{x}, \overline{z})$, there exists a tuple $\overline{w}_j$ from I such that $\eta(\theta(\overline{x})) = \overline{w}_j$, where η is the mapping from $Class_\pi$ into $C \cup Y$ defined by: $\eta([t]_\pi) = c$ if $c \in [t]_\pi$ and $\eta([t]_\pi) = y$ otherwise, where y is a variable from the class $[t]_\pi$.

Remark 2: Let I, π and T as in Definition 4. We have:
(i) For each $i, 1 \le i \le m$, there exists a substitution θ_i from the variable set of Q into $Class_\pi$ such that $T \models \theta_i f_V(\overline{x}, \overline{z})$ and $\eta(\theta_i(\overline{x})) = \overline{w}_i$, where the mapping θ_i is specified in Definition 4.

Proof: We specify the substitution θ_i. If $\overline{x} = x_1 \ldots x_h$, $\overline{z} = z_1 \ldots z_p$, $\overline{w}_i = t_1 \ldots t_h$, where $t_j \in Dom$, $\overline{y}_i = y_{\gamma_1} \ldots y_{\gamma_p}$ (the vector $\overline{y}_i$ consists of the y-variables from the expression $f_V(\overline{w}_i, \overline{y}_i)$). The mapping θ_i is defined as follows: $\theta_i(x_j) = [t_j]_\pi$, $1 \le j \le h$ and $\theta_i(z_j) = [y_{\gamma_j}]_\pi$, $1 \le j \le p$. ■
Now, we give the results regarding the representation of certain answer sets under CWA.

Theorem 4: Let $\mathcal{V}$ be a set of view definitions, I an extension of $\mathcal{V}$, and D a database on Dom such that $I = \mathcal{V}(D)$. Let $\phi(f_V)$ be the formula constructed for I and $\mathcal{V}$. There exist a C-partition π from $Part(C \cup Y)$ such that $\pi(\phi(f_V)) = TRUE$, a database $D' \subseteq D$, and an injective mapping ψ_π from $Class_\pi$ into $val(D')$ having the following properties:
(i) and (ii) as in Theorem 1,
$(iii)'$ Let $T = \psi_\pi^{-1}(D')$. We have $T \in \mathcal{M}_\pi$, and T is closed with respect to I (Definition 4).
$(iv)'$ $I = \mathcal{V}(D')$.

Theorem 5: Let π be a C-partition from $Part(C \cup Y)$ such that $\pi(\phi(f_V)) = TRUE$. Let T be a database from $\mathcal{M}_\pi$ that is closed with respect to I. Then for each injective C-mapping ψ from $Class_\pi$ into Dom, we have:
(i) $I = \mathcal{V}(D')$, where $D' = \psi(T)$.
Now, we use the notations specified for the case OWA, with except the following: $\overline{S}_{\pi_i}$ instead of S_{π_i}, $CertAnsC(\mathcal{V}, I, Q)$ instead of $CertAnsO(\mathcal{V}, I, Q)$, and $RCertAnsC$ instead of $RCertAnsO$, where
$\overline{S}_{\pi_i} = \cap\{Q(T) | T \in \mathcal{M}_{\pi_i}$ and T is closed w.r.t. $I\}$.

Theorem 6: Using the notations specified above, we have $CertAnsC(\mathcal{V}, I, Q)$=$RCertAnsC$.

VII. Time Complexity to Compute Certain Answers

It is known that the total number of partitions of an $n-$element set is the Bell number B_n, such that the following recursion equation yields ([13]): $B_{n+1} = \Sigma_{k=0}^{k=n} B_k$. Using the induction, we get the following inequalities: $B_n > 2^n$ for each $n \ge 5$ and $B_n < n^n$ for each $n > 1$. The second inequality implies $B_n < 2^{n^2}$, for each $n > 1$. On the other hand, the number of C-partitions defined on $C \cup Y$ is greater than the number of the partitions on Y. It results that the number of C-partitions defined on $C \cup Y$ is of the type $O(2^{p(|Y|)})$, where p is a polynomial. Let us discuss the number of all elements from $\mathcal{M}_\pi$, where π is a fixed C-partition on $C \cup Y$. Let $y_\pi = |Class_\pi|$, and r the maximum of the arities of the relational symbols from V. Then the cardinality of the set T_π^{max}(in relation (4))has the form: $O(y_\pi^r)$. Therefore, the number of the elements from $\mathcal{M}_\pi$ has the form $O(2^{y_\pi^r})$. It is clear that $y_\pi \le |C \cup Y|$. For a query Q having l variables, and T an element from $\mathcal{M}_\pi$, the number of the substitutions from $Var(Q)$ into $Class_\pi$ is y_π^l. Using these relations, and the constructions for certain answers under OWA and CWA, specified in Sections V and VI, we obtain that the time complexity to compute these certain answers is $EXPTIME$.

VIII. Conclusion

We have presented a representation of certain answers corresponding to a set of view definitions, a set of extensions of the view definitions, and a query. Two situations were considered: open-world assumption and close-world assumption. Using this representation, a method to compute the certain answers under the two assumptions was given.

References

[1] A. Y. Halevy, *Answering Queries Using Views: A survey*, VLDB Journal, vol. 10, nr. 4, 2001, pp. 270-294.
[2] S. Abiteboul and O. M. Duschka, *Complexity of answering queries using materialized views*, in PODS, 1998, pp.254-263.
[3] S. Flesca and S. Greco, *Rewriting queries using views*, IEEE Trans. Knowledge Data Engineering 13(6), 2001, pp.980-995.
[4] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi, *View-based query processing:On the relationship between rewriting, answering and losslessness*, Theoretical Computer Science, 371, 2007, pp. 169-182.
[5] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi, *Answering Regular Path Queries Views*, Proc. of the 16th IEEE Int. Conf. on Data Engineering, ICDE, 2000, pp. 389-398.
[6] G. Grahne, and A. O. Mendelson, *Tableau technique for querying information sources through global schemas*, Proc. of the 7th Int. Conf. on Database Theory, ICDT'99, in LNCS, vol. 1540, Springer 1999, pp. 332-347.
[7] A. Call, D. Lembro, and R. Rosati, *Query rewriting and answering under constraints in data integration systems*, IJCAI-03, 2003, pp.16-21.
[8] F. Afrati, C. Li, and P. Mitra, *Rewriting queries using views in the presence of arithmetic comparisons*, Theoretical Computer Science, 368, 2006, pp.88-123.
[9] O. M. Duschka, M. R. Genesereth, and A. Levy, *Recursive Query Plans for Data Integration*, J. of Logic Programming 2000, 43(1), pp. 49-73.
[10] T. Millstein, A. Halevy, and M.Friedman, *Query containment for data integration systems*, JCSS vol.66, 2003, pp. 20-39.
[11] Wei F. and Lausen G.: *Containment of Conjunctive Queries with Safe Negation*. In ICDT, (2003), LNCS, vol.2572, 346-360.
[12] J. Hammer, H. Garcia-Molina, S. Nestorov, R. Yerneni, M. M. Breunig, and V. Vassalos, *Template-based wrappers in the TSIMMIS system*, Proc. of ACM SIGMOD, 1997, pp. 532-535.
[13] R.A.Brualdi, *Introductory Combinatorics (4th edition ed.)*, Pearson Prentice Hall, ISBN 0131001191, 2004.