View-Based Query Containment

Diego Calvanese
Giuseppe De Giacomo
Maurizio Lenzerini
Dipartimento di Informatica e Sistemistica
Univ. di Roma "La Sapienza"
Via Salaria 113, I-00198 Roma, Italy
lastname@dis.uniroma1.it

Moshe Y. Vardi
Department of Computer Science
Rice University, P.O. Box 1892
Houston, TX 77251-1892, U.S.A.
vardi@cs.rice.edu

ABSTRACT

Query containment is the problem of checking whether for all databases the answer to a query is a subset of the answer to a second query. In several data management tasks, such as data integration, mobile computing, etc., the data of interest are only accessible through a given set of views. In this case, containment of queries should be determined relative to the set of views, as already noted in the literature. Such a form of containment, which we call *view-based query containment*, is the subject of this paper. The problem comes in various forms, depending on whether each of the two queries is expressed over the base alphabet or the alphabet of the view names. We present a thorough analysis of view-based query containment, by discussing all possible combinations from a semantic point of view, and by showing their mutual relationships. In particular, for the two settings of conjunctive queries and two-way regular path queries, we provide both techniques and complexity bounds for the different variants of the problem. Finally, we study the relationship between view-based query containment and view-based query rewriting.

1. Introduction

Querying is the fundamental mechanism for extracting information from a database. Besides the basic task of query answering, i.e., evaluating a query over a database, data and knowledge representation systems should support other reasoning services related to querying. One of the most important is query containment, i.e., verifying whether for all databases the answer to a query is a subset of the answer to a second query. Checking containment of queries is crucial in several contexts, such as query optimization, query reformulation, knowledge-base verification, information integration, integrity checking, and cooperative answering [22, 28, 16, 5, 9, 33, 27, 11, 18, 20, 32]. Obviously, query containment is also useful for checking equivalence of queries, i.e., verifying whether for all databases the answer to a query is the same as the answer to another query. The main results on query containment are summarized in [15].

PODS 2003, June 9-12, 2003, San Diego, CA.

In several data management tasks, the data of interest are only accessible through a given set of views. In other words, we would like to access a database, but we have only information about the data satisfying the views. For example, in data integration [25], data are stored in local sources modeled as views over a virtual database. It follows that answering a query expressed over the virtual database schema, amounts to answering the query based on the data satisfying the views, rather than the data in the database. Similarly, a data warehouse can be seen as a set of materialized views computed over a collection of data sources. Only these views, and not the raw data at the sources, are accessible when answering queries posed to the data warehouse. Also, in mobile computing, the data stored in the local device are those satisfying a set of views over a global database, and it is often desirable to avoid accessing the global database, in order to save network bandwidth. In all the above mentioned settings, query processing comes in a different form with respect to the traditional framework. Instead of considering a query over of a given database, one should take into account that the query is now processed by relying only on the information about the data satisfying the views. We call this kind of processing *view-based query processing* [12].

Again, the basic reasoning services related to view-based query processing is the one aiming at computing the answer to a query, based on the available views. The problem, called *view-based query answering* has been the subject of an extensive investigation in the last years [23]. Analogously to the traditional setting, also in the context of view-based query processing, the need arises of checking containment between queries. In this context, containment of queries should be determined relative to the set of views, as already noted in the literature [31, 30]. Such form of containment, called *view-based query containment*, is the subject of this paper.

To the best of our knowledge, the first paper dealing with view-based query containment is [31], where the problem, called "relative containment", is studied for variants of conjunctive queries and views. In particular, it is shown that relative containment is Π_2^p-complete in the case of conjunctive queries and views. In [30], the results are extended to the case where views have limited access patterns.

Although previous work on view-based query containment considered only the case of queries expressed in the alphabet of the database (base alphabet), we show here that the problem comes in various forms, depending on whether each of the two queries is expressed over the base alphabet, or the alphabet of the view names (view alphabet). The first contribution of this paper is a thorough analysis of view-based query containment. In particular, we discuss several semantics of containment for all the possible forms, and we study their mutual relationships.

The second contribution is a study of view-based query containment in two specific contexts: conjunctive queries in relational databases, and regular path queries in semistructured databases. The standard model for semistructured databases are edge-labeled graphs, where nodes represent objects, and labeled edges between two nodes represent links between objects [8, 19]. This model captures also data expressed using XML-like languages [7, 10]. *Regular path queries* (RPQ), which ask for all pairs of objects that are connected by a path conforming to a regular expression, are the basic querying mechanism in this framework [8, 2, 4]. Regular path queries can express complex navigations in a graph. In particular, union and transitive closure are crucial when we do not have a complete knowledge of the structure of the database. In our regular path queries, we include also the *inverse* operator, which enables us to navigate edges backward [8, 9], for example, from a child to its parent. We call these queries *two-way regular path queries* (2RPQs). For both conjunctive and two-way regular path queries and views, we provide techniques and complexity bounds for the different variants of view-based query containment.

In establishing the results for 2RPQs, we make use of and extend the known connection between view-based query answering and constraint satisfaction [13]. In particular, we show that such a connection implies a strong relationship between constraint satisfaction and view-based query containment, in the cases where at least one of the queries is expressed in the base alphabet. This relationship complements the one between constraint satisfaction and conjunctive query containment [24].

A third contribution is a study of the relationship between view-based query containment and view-based query rewriting. In particular, we show that view-based query containment allows one to verify desirable properties of rewritings.

The rest of the paper is organized as follows. Section 2 introduces the technical preliminaries on databases and queries. Section 3 extends the connection between view-based query answering and constraint satisfaction of [13] to 2RPQs. Sections 4, 5, 6, and 7 provide the definitions and the results for view-based query containment, in the various cases. Section 8 discusses the connection with view-based query rewriting. Section 9 concludes the paper.

2. Preliminaries

In this paper, we consider databases and view extensions as finite relational structures. Let Θ be an alphabet of relation symbols, each with an associated arity. A *finite relational structure* (or simply *structure*) $\mathcal{R}$ over Θ is a pair $(\Delta^{\mathcal{R}}, \cdot^{\mathcal{R}})$, where $\Delta^{\mathcal{R}}$ is a finite domain and $\cdot^{\mathcal{R}}$ is a function that assigns to each relation symbol in $r \in \Theta$ a relation $r^{\mathcal{R}}$, also denoted by $r(\mathcal{R})$, of the appropriate arity over $\Delta^{\mathcal{R}}$. Given a *query* Q^Θ over Θ, we denote by $Q^\Theta(\mathcal{R})$ the result of evaluating Q^Θ over $\mathcal{R}$. Note that we explicitly annotate queries with the alphabet over which they are defined. Given two structures $\mathcal{R}_1$ and $\mathcal{R}_2$ over Θ, we use $\mathcal{R}_1 \subseteq \mathcal{R}_2$ to denote that $r(\mathcal{R}_1) \subseteq r(\mathcal{R}_2)$, for each $r \in \Theta$. A query Q^Θ is *monotone* if $Q^\Theta(\mathcal{R}_1) \subseteq Q^\Theta(\mathcal{R}_2)$ whenever $\mathcal{R}_1 \subseteq \mathcal{R}_2$.

Let Σ be a finite alphabet of relation symbols, fixed once and for all, which we call *base alphabet*. A *database* is a structure over Σ. Consider a database that is accessible only through a collection $\mathcal{V}$ of views, and suppose we want to answer a query over the database only on the basis of our knowledge on the views. Specifically, the collection of views is represented by a finite set $\mathcal{V}$ of *view symbols*, each denoting a relation. Each view symbol $V \in \mathcal{V}$ has an associated *view definition* V^Σ, which is a query over Σ. A $\mathcal{V}$*-extension* $\mathcal{E}$ is a structure over $\mathcal{V}$.

We consider views to be *sound* [3, 21], i.e., we model a situation where the extension of the views provides a subset of the results of applying the view definitions to the database[1]. Formally, given a set of views $\mathcal{V}$ and a database $\mathcal{B}$, we use $\mathcal{V}^\Sigma(\mathcal{B})$ to denote the $\mathcal{V}$-extension $\mathcal{E}$ such that $V(\mathcal{E}) = V^\Sigma(\mathcal{B})$, for each $V \in \mathcal{V}$. We say that a $\mathcal{V}$-extension $\mathcal{E}$ is *sound wrt a database* $\mathcal{B}$ if $\mathcal{E} \subseteq \mathcal{V}^\Sigma(\mathcal{B})$. In other words, for a view $V \in \mathcal{V}$, all the tuples in $V(\mathcal{E})$ must appear in $V^\Sigma(\mathcal{B})$, but $V^\Sigma(\mathcal{B})$ may contain tuples not in $V(\mathcal{E})$. A set $\mathcal{V}$ of views is *non-constraining* if each $\mathcal{V}$-extension is sound wrt some database, i.e., for each $\mathcal{V}$-extension $\mathcal{E}$ there exists a database $\mathcal{B}$ such that $\mathcal{E} \subseteq \mathcal{V}^\Sigma(\mathcal{B})$. Intuitively, all extensions of non-constraining views are admissible, since they are sound wrt some database.

Given a set $\mathcal{V}$ of views, a $\mathcal{V}$-extension $\mathcal{E}$, and a query Q^Σ, the set of *certain answers under sound views* to Q^Σ with respect to $\mathcal{V}$ and $\mathcal{E}$ is the set of tuples t of objects such that $t \in Q^\Sigma(\mathcal{B})$ for every database $\mathcal{B}$ wrt which $\mathcal{E}$ is sound, i.e., $\mathcal{E} \subseteq \mathcal{V}^\Sigma(\mathcal{B})$. *View-based query answering under sound views* consists in deciding whether a given tuple of objects is a certain answer under sound views to Q with respect to $\mathcal{V}$ and $\mathcal{E}$. Given a set $\mathcal{V}$ of views and a query Q^Σ, we denote by $cert_{Q^\Sigma,\mathcal{V}}$ the query that, for every $\mathcal{V}$-extension $\mathcal{E}$, returns the set of certain answers under sound views to Q^Σ with respect to $\mathcal{V}$ and $\mathcal{E}$. Since in this paper we consider sound views only, we simply refer to *certain answers* and *view-based query answering*, thus dropping the qualification "under sound views".

We consider the case of conjunctive queries over relational databases. In particular, we deal with standard conjunctive queries without equalities and without constants. Observe that such queries are monotone and that every set of views whose definitions are conjunctive queries is non-constraining. We also consider (in fact mainly focus on) the case of regular path queries over semistructured databases. A *semistructured database* is a finite graph whose nodes represent objects and whose edges are labeled by elements from

[1]This corresponds to adopting the so-called "open world assumption" for the views.

Σ [8, 1]. An edge (x, r, y) from node x to node y labeled by r represents the fact that relation r holds between the object x and the object y. Note that a semistructured database can be viewed as a structure $\mathcal{B}$ over the set Σ of binary relational symbols.

A *regular-path query* (RPQ) over an alphabet Θ of binary relation symbols is expressed as a regular expression or a finite-state automaton over Θ. When evaluated on a structure $\mathcal{R}$ over Θ, an RPQ computes the set of pairs of objects connected in $\mathcal{R}$ by a path in the regular language defined by the RPQ. We consider *two-way regular-path queries* (2RPQs) [12], which extend RPQs with the *inverse* operator. Formally, let $\Theta^{\pm} = \Theta \cup \{r^- \mid r \in \Theta\}$ be the alphabet including a new symbol r^- for each r in Θ. Intuitively, r^- denotes the inverse of the binary relation r. If $q \in \Theta^{\pm}$, then we use q^- to mean the *inverse* of q, i.e., if q is r, then q^- is r^-, and if q is r^-, then q^- is r. 2RPQs are expressed by means of regular expressions or finite-state automata over $\Theta^{\pm}$, whose language is different from the language consisting only of the empty word ε. When evaluated on a structure $\mathcal{R}$ over Θ, a 2RPQ Q^{Θ} computes the set $Q^{\Theta}(\mathcal{R})$ of pairs of objects connected in $\mathcal{R}$ by a semipath that conforms to the regular language $L(Q^{\Theta})$. A *semipath* in $\mathcal{R}$ from x to y (labeled with $q_1 \cdots q_n$) is a sequence of the form $(y_0, q_1, y_1, \ldots, y_{n-1}, q_n, y_n)$, where $n \geq 0$, $y_0 = x$, $y_n = y$, and for each y_{i-1}, q_i, y_i, we have that $q_i \in \Theta^{\pm}$, and, if $q_i = r$ then $(y_{i-1}, y_i) \in r(\mathcal{R})$, and if $q_i = r^-$ then $(y_i, y_{i-1}) \in r(\mathcal{R})$. We say that a semipath $(y_0, q_1, \ldots, q_n, y_n)$ *conforms to* Q^{Θ} if $q_1 \cdots q_n \in L(Q^{\Theta})$. A semipath is said to be *simple* if no object appears more than once in the corresponding sequence. Observe that 2RPQs (resp., RPQs) are monotone and that every set of 2RPQ (resp., RPQ) views is non-constraining.

Given two queries Q_1^{Θ} and Q_2^{Θ} over the same alphabet Θ, we say that Q_1^{Θ} is *contained in* Q_2^{Θ}, denoted $Q_1^{\Theta} \subseteq Q_2^{\Theta}$, if for every structure $\mathcal{R}$ over Θ, we have that $Q_1^{\Theta}(\mathcal{R}) \subseteq Q_2^{\Theta}(\mathcal{R})$. In this paper we consider query containment relative to a set of views. In particular, we study the notion of *view-based query containment of a query* $Q_1^{\Theta_1}$ *in a query* $Q_2^{\Theta_2}$ *wrt a set* $\mathcal{V}$ *of views*, denoted by $Q_1^{\Theta_1} \subseteq_{\mathcal{V}} Q_2^{\Theta_2}$, where each Θ_i is either Σ or $\mathcal{V}$. The semantics of this notion depends on Θ_1 and Θ_2, i.e., the alphabets over which the two queries are expressed, and will be discussed for the various cases in the next sections. The complexity results we provide refer to two specific settings: conjunctive queries, where both queries and views are conjunctive, and 2RPQs, where both queries and views are 2RPQs.

3. Constraint-satisfaction and Query Answering

A *constraint-satisfaction problem (CSP)* is traditionally defined in terms of a set of variables, a set of values, and a set of constraints, and asks whether there is an assignment of the variables with the values that satisfies the constraints. A characterization of CSP can be given in terms of homomorphisms between structures [17].

A *homomorphism* $h : A \rightarrow B$ between two structures A and B over the same alphabet is a mapping $h : \Delta^A \rightarrow \Delta^B$ such that, if $(c_1, \ldots, c_n) \in R^A$, then $(h(c_1), \ldots, h(c_n)) \in R^B$, for every relation symbol R in the alphabet. Let $\mathcal{A}$ and $\mathcal{B}$ be two classes of structures. The *(uniform) constraint-satisfaction problem* $CSP(\mathcal{A}, \mathcal{B})$ is the following decision problem: given a structure $A \in \mathcal{A}$ and a structure $B \in \mathcal{B}$ over the same alphabet, is there a homomorphism $h : A \rightarrow B$? When $\mathcal{B}$ consists of a single structure B and $\mathcal{A}$ is the set of all structures over the alphabet of B, we get the so-called *non-uniform* constraint-satisfaction problem, denoted by $CSP(B)$, where B is fixed and the input is just a structure $A \in \mathcal{A}$. As usual, we consider $CSP(B)$ as the set of structures A such that there is a homomorphism from A to B.

Interestingly, containment between two CSPs can be characterized in terms of homomorphism. Indeed, the following general result holds.

PROPOSITION 1. *Let B_1 and B_2 be two structures over the same alphabet. Then $CSP(B_1)$ is contained in $CSP(B_2)$ iff there exists a homomorphism from B_1 to B_2.*

PROOF. "$\Leftarrow$" Suppose there exists a homomorphism h from B_1 to B_2. Given an instance A of $CSP(B_1)$, by definition, there is a homomorphism g from A to B_1. Then $g \circ h$ is an homomorphism from A to B_2.

"$\Rightarrow$" B_1 is obviously in $CSP(B_1)$, since the identity function is an homomorphism. Since by hypothesis $CSP(B_1)$ is contained in $CSP(B_2)$, we have that B_1 is in $CSP(B_2)$, i.e, there is an homomorphism from B_1 to B_2. $\square$

We will exploit such a characterization for the results in the following sections.

A tight relationship between non-uniform CSP and view-based query answering for RPQs is illustrated in [13], which provides a polynomial reduction from view-based query answering for RPQs to non-uniform CSP. In this paper we extend this relationship to 2RPQs and exploit it in different ways, by making use of the notions of constraint template and constraints instance (both for query answering and for view-based query answering), defined as follows.

Given a 2RPQ Q^{Θ} over an alphabet Θ, the *constraint template* $CT_{Q^{\Theta},\Theta}$ of Q^{Θ} wrt Θ is the structure C defined as follows.

- The alphabet of C is $\Theta^{\pm} \cup \{U_i, U_f\}$, where U_i and U_f denote unary relation symbols.
- Let $A_{Q^{\Theta}} = (\Theta^{\pm}, S, S_0, \rho, F)$ be a (nondeterministic) automaton for Q^{Θ}. The structure $C = (\Delta^C, \cdot^C)$ is given by:
  - $\Delta^C = 2^S$;
  - $\sigma \in U_i^C$ iff $S_0 \subseteq \sigma$;
  - $\sigma \in U_f^C$ iff $\sigma \cap F = \emptyset$;
  - $(\sigma_1, \sigma_2) \in r^C$ iff $\rho(\sigma_1, r) \subseteq \sigma_2$ and $\rho(\sigma_2, r^-) \subseteq \sigma_1$ — we consider here ρ as extended to sets of states in the usual way.

Given a structure $\mathcal{R}$ over Θ and a pair of objects c and d, the *constraint instance* $\mathcal{R}^{c,d}$ of $CSP(CT_{Q^{\Theta},\Theta})$ is the structure $I = (\Delta^I, \cdot^I)$ defined as follows:

- $\Delta^I = \Delta^{\mathcal{R}} \cup \{c, d\}$;
- $r^I = r^{\mathcal{R}}$, for each $r \in \Theta$;
- $U_i^I = \{c\}$, and $U_f^I = \{d\}$.

THEOREM 2. *Let Q^Θ be a 2RPQ over an alphabet Θ, $\mathcal{R}$ a structure over Θ, and c, d a pair of objects. Then, $(c,d) \notin Q^\Theta(\mathcal{R})$ if and only if there is a homomorphism from $\mathcal{R}^{c,d}$ to $CT_{Q^\Theta,\Theta}$.*

PROOF. "$\Leftarrow$" By contradiction. Let h be a homomorphism from $\mathcal{R}^{c,d}$ to $CT_{Q^\Theta,\Theta}$, and assume there is a semipath $p = (x_0, q_1, x_1, \ldots, x_{m-1}, q_m, x_m)$ from $c = x_0$ to $d = x_m$ in $\mathcal{R}$ with $q_1 \cdots q_m \in L(Q^\Theta)$. Hence there is a sequence $\delta = (s_0, \ldots, s_m)$ of states of A_{Q^Θ} such that $s_0 \in S_0$, $s_m \in F$, and $s_{i+1} \in \rho(s_i, q_{i+1})$ for each $i \in \{0, \ldots, m-1\}$. Since h is a homomorphism, we have that $s_0 \in h(c)$, and hence, by construction of the relations r_i in the constraint template, we have that $s_i \in h(x_i)$ for each $i \in \{0, \ldots, m-1\}$. Moreover, since $h(d) \in U_f$, we have that $s_m \notin F$. Contradiction.

"$\Rightarrow$" Given a structure $\mathcal{R}$ and two objects c, d such that $(c,d) \notin Q^\Theta(\mathcal{R})$, we build a mapping $h : \Delta^{\mathcal{R}} \to 2^S$ by putting each state in S_0 in $h(c)$ and repeating the following until h does not change any more: if $(x,y) \in r(\mathcal{R})$ and $s \in h(x)$ then add $\rho(s,r)$ to $h(y)$, and if $(x,y) \in r(\mathcal{R})$ and $s \in h(y)$ then add $\rho(s, r^-)$ to $h(x)$. Note that, since $(c,d) \notin Q^\Theta(\mathcal{R})$, we have that $h(d) \cup F = \emptyset$. Hence, h is indeed an homomorphism from $\mathcal{R}^{c,d}$ to $CT_{Q^\Theta,\Theta}$. $\square$

We note that it follows from Theorem 2 and [17] that the complement of a 2RPQ can be expressed in monadic NP (i.e., in monadic existential second-order logic). In contrast, it follows from [6] that even a simple query such as a^* cannot be expressed in monadic NP.

Next we define constraint templates for the certain answers of 2RPQs wrt a set of 2RPQ views, extending the analogous notion for RPQs in [13]. Given a 2RPQ Q^Σ and a set $\mathcal{V}$ of 2RPQ views, the *constraint template* $CT_{Q^\Sigma,\mathcal{V}}$ of Q^Σ wrt $\mathcal{V}$ is the structure C defined as follows.

- The alphabet of C is $\mathcal{V} \cup \{U_i, U_f\}$.
- Let $A_{Q^\Sigma} = (\Sigma^\pm, S, S_0, \rho, F)$ be a (nondeterministic) automaton for Q^Σ. The structure $C = (\Delta^C, \cdot^C)$ is given by:
  - $\Delta^C = 2^S$;
  - $\sigma \in U_i^C$ iff $S_0 \subseteq \sigma$;
  - $\sigma \in U_f^C$ iff $\sigma \cap F = \emptyset$;
  - $(\sigma_1, \sigma_2) \in V^C$ iff there exists a word $q_1 \cdots q_k \in L(V)$ and a sequence $T_0, \ldots, T_k$ of subsets of S such that the following hold:
    1. $T_0 = \sigma_1$ and $T_k = \sigma_2$,
    2. if $s \in T_i$ and $t \in \rho(s, q_i)$ then $t \in T_{i+1}$, for $0 \le i < k$, and
    3. if $s \in T_i$ and $t \in \rho(s, q_i^-)$ then $t \in T_{i-1}$, for $0 < i \le k$.

To check the existence of a word $q_1 \cdots q_k \in L(V)$ and of a sequence $T_0, \ldots, T_k$ of subsets of S such that conditions 1–3 above are satisfied, we check nonemptiness of the intersection of V with an automaton A defined as follows [35]. $A = (\Sigma^\pm, S^A, I^A, \delta^A, F^A)$ with $S^A = 2^S \times 2^S$, $I^A = \{(\sigma_1, \sigma_1)\}$, $F^A = \{(T, \sigma_2) \mid T \in 2^S\}$, and with $(U, V) \in \delta^A((T, U), q)$ if the following holds:

- if $s \in U$ and $t \in \rho(s, q^-)$, then $t \in T$;
- if $s \in U$ and $t \in \rho(s, q)$, then $t \in V$.

It is easy to see that the automaton A accepts a word $q_1 \cdots q_k$ if there exists a sequence $T_0, \ldots, T_k$ of subsets of S such that conditions 1–3 above are satisfied. Hence, by constructing A on the fly while checking for nonemptiness, one can verify whether $(\sigma_1, \sigma_2) \in V^C$ in NLOGSPACE in the size of V and in PSPACE in the size of Q^Σ.

Given a $\mathcal{V}$-extension $\mathcal{E}$ and a pair of objects c and d, the *constraint instance* $\mathcal{E}^{c,d}$ of $CSP(CT_{Q^\Sigma,\mathcal{V}})$ is the structure $I = (\Delta^I, \cdot^I)$ over the alphabet $\mathcal{V} \cup \{U_i, U_f\}$ defined as follows:

- $\Delta^I = \Delta^{\mathcal{E}} \cup \{c, d\}$;
- $V^I = V^{\mathcal{E}}$, for each $V \in \mathcal{V}$;
- $U_i^I = \{c\}$, and $U_f^I = \{d\}$.

THEOREM 3. *Let Q^Σ be a 2RPQ, $\mathcal{V}$ a set of 2RPQ views, $\mathcal{E}$ a $\mathcal{V}$-extension, and c, d a pair of objects. Then, $(c,d) \notin cert_{Q^\Sigma,\mathcal{V}}(\mathcal{E})$ if and only if there is a homomorphism from $\mathcal{E}^{c,d}$ to $CT_{Q^\Sigma,\mathcal{V}}$.*

PROOF. "$\Leftarrow$" By contradiction. Given a homomorphism h from $\mathcal{E}^{c,d}$ to $CT_{Q^\Sigma,\mathcal{V}}$, we construct a database $\mathcal{B}$ such that $\mathcal{E} \subseteq \mathcal{V}(\mathcal{B})$ as follows: for every view V and every pair $(a, b) \in V(\mathcal{E})$ we choose a word $w = q_1 \cdots q_k \in L(V^\Sigma)$ such that there is a sequence $T_0, \ldots, T_k$ of subsets of S satisfying the condition used to build the constraint template, and introduce in $\mathcal{B}$ a simple semipath $(y_0, q_1, y_1, \ldots, y_{k-1}, q_k, y_k)$ where $y_0 = a$, $y_k = b$ and $y_1, \ldots, y_{k-1}$ are new objects. Suppose that $(c,d) \in Q^\Sigma(\mathcal{B})$, i.e., there is a semipath $p = (x_0, q_1, x_1, \ldots, x_{m-1}, q_m, x_m)$ from $c = x_0$ to $d = x_m$ in $\mathcal{B}$ with $q_1 \cdots q_m \in L(Q^\Sigma)$. Hence there is a sequence $\delta = (s_0, \ldots, s_m)$ of states of A_{Q^Σ} such that $s_0 \in S_0$, $s_m \in F$, and $s_{i+1} \in \rho(s_i, q_{i+1})$ for each $i \in \{0, \ldots, m-1\}$. Let x_i and $x_i + h$ be two objects in the extension $\mathcal{E}$ such that, for each $i < j < i{+}h$, x_j is one of the new objects introduced by the construction of $\mathcal{B}$. By construction of the relations V in the constraint template, we have that, if $s_i \in h(x_i)$, then $s_{i+h} \in h(x_{i+h})$. Now we have that $s_0 \in h(c)$, and hence, by induction on the number of objects among $x_1, \ldots, x_m$ that appear in $\mathcal{E}$, we have that $s_m \in h(d)$. But since $h(d) \in U_f$, we have that $s_m \notin F$. Contradiction.

"$\Rightarrow$" Given a database $\mathcal{B}$ such that $\mathcal{E} \subseteq \mathcal{V}(\mathcal{B})$ and two objects c, d such that $(c,d) \notin Q^\Sigma(\mathcal{B})$, we build a mapping $h' : \Delta^{\mathcal{B}} \to 2^S$ by putting each state in S_0 in $h'(c)$ and repeating the following until h does not change any more: if $(x,y) \in r(\mathcal{B})$ and $s \in h'(x)$ then add $\rho(s,r)$ to $h'(y)$, and if $(x, y) \in r(\mathcal{B})$ and $s \in h'(y)$ then add $\rho(s, r^-)$ to $h'(x)$. Note that, since $(c, d) \notin Q^\Sigma(\mathcal{B})$, we have that $h'(d) \cup F = \emptyset$. Projecting h' on $\Delta^\mathcal{E}$ we obtain a homomorphism from $\mathcal{E}^{c,d}$ to $CT_{Q^\Sigma,\mathcal{V}}$. $\Box$

Again, by Theorem 2 and [17], it follows that the complement of computing certain tuples of a 2RPQ with respect to a fixed set of 2RPQ views is in monadic NP. The proof via CSP is much simpler than the proof in [12] that computing the certain answers is in coNP in data complexity.

4. View-based containment of Q_1^Σ in Q_2^Σ

The first problem we address is view-based containment between two queries over the base alphabet. This is the problem originally studied in [31].

Definition 1. *$Q_1^\Sigma \subseteq_\mathcal{V} Q_2^\Sigma$ if for every database $\mathcal{B}$, and for every $\mathcal{V}$-extension $\mathcal{E}$ with $\mathcal{E} \subseteq \mathcal{V}(\mathcal{B})$, we have $cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{E}) \subseteq cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{E})$.*

A simple example showing the difference between traditional query containment and view-based query containment is the following. Let the base alphabet be $\Sigma = \{person, worksfor\}$ and consider the two queries

$$\begin{aligned} Q_1^\Sigma &= \{x \mid person(x)\} \\ Q_2^\Sigma &= \{x \mid person(x), worksfor(x, y)\} \end{aligned}$$

Obviously, Q_1^Σ is not contained in Q_2^Σ. Now, suppose our database is only accessible through the views $\mathcal{V} = \{V\}$, where the definition associated to the only view V is $V^\Sigma = \{x \mid person(x), worksfor(x, y)\}$. It is easy to see that, for every extension E of V, the certain answers to both Q_1^Σ and Q_2^Σ wrt $\mathcal{V}$ and E coincide with E. It follows that view-based containment between Q_1^Σ and Q_2^Σ holds, i.e., $Q_1^\Sigma \subseteq_\mathcal{V} Q_2^\Sigma$.

Observe that in Definition 1 we consider only $\mathcal{V}$-extensions that are sound wrt some database. It turns out that, w.l.o.g. we can drop this requirement of $\mathcal{V}$-extensions. Intuitively, this happens because for each extension $\mathcal{E}$ that is not contained in $\mathcal{V}(\mathcal{B})$ for some database $\mathcal{B}$, both $cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{E})$ and $cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{E})$ become trivial and return all possible tuples of objects in $\mathcal{E}$. Formally, we have the following result.

Proposition 4. *$Q_1^\Sigma \subseteq_\mathcal{V} Q_2^\Sigma$ if and only if for every $\mathcal{V}$-extension $\mathcal{E}$, we have $cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{E}) \subseteq cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{E})$.*

Proof. "$\Leftarrow$" Obvious. "$\Rightarrow$" By contradiction. Suppose that for all databases $\mathcal{B}$, and for all $\mathcal{V}$-extensions $\mathcal{E}$ with $\mathcal{E} \subseteq \mathcal{V}(\mathcal{B})$, we have that $cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{E}) \subseteq cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{E})$, but there exists an $\bar{\mathcal{E}}$ such that $cert_{Q_1^\Sigma,\mathcal{V}}(\bar{\mathcal{E}}) \not\subseteq cert_{Q_2^\Sigma,\mathcal{V}}(\bar{\mathcal{E}})$. The latter implies that there is a tuple t of objects in $\bar{\mathcal{E}}$ such that $t \in cert_{Q_1^\Sigma,\mathcal{V}}(\bar{\mathcal{E}})$ and for a database $\bar{\mathcal{B}}$ with $\bar{\mathcal{E}} \subseteq \mathcal{V}(\bar{\mathcal{B}})$ we have $t \notin Q_2^\Sigma(\bar{\mathcal{B}})$. But this contradicts the hypothesis. $\Box$

Note that the condition in the proposition above is actually the definition of relative containment in [31].

One could also compare the two queries only wrt extensions that correspond exactly to the evaluation of the views over some database. Formally, this notion corresponds to checking whether, for all databases $\mathcal{B}$, we have $cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{V}(\mathcal{B}))$. It turns out that such a notion is equivalent to view-based query containment as defined in Definition 1.

Proposition 5. *$cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{V}(\mathcal{B}))$ for every database $\mathcal{B}$, if and only if $Q_1^\Sigma \subseteq_\mathcal{V} Q_2^\Sigma$.*

Proof. By Proposition 4, it suffices to show that the conditions:

1. for every $\mathcal{V}$-extension $\mathcal{E}$ we have that $cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{E}) \subseteq cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{E})$
2. for every database $\mathcal{B}$, we have that $cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{V}(\mathcal{B}))$

are equivalent.

(1) implies (2): straightforward, since for each database $\mathcal{B}$, $\mathcal{V}(\mathcal{B})$ is a $\mathcal{V}$-extension. (2) implies (1): we show that not (1) implies not (2). Assume that there exists a $\mathcal{V}$-extension $\bar{\mathcal{E}}$ and a tuple of objects t such that for all databases $\mathcal{B}'$ such that $\bar{\mathcal{E}} \subseteq \mathcal{V}(\mathcal{B}')$, we have that $t \in Q_1^\Sigma(\mathcal{B}')$, and there exists a database $\bar{\mathcal{B}}$ such that $\bar{\mathcal{E}} \subseteq \mathcal{V}(\bar{\mathcal{B}})$ and $t \notin Q_2^\Sigma(\bar{\mathcal{B}})$. Observe that, since $\bar{\mathcal{E}} \subseteq \mathcal{V}(\bar{\mathcal{B}})$, we have that $t \in Q_1^\Sigma(\bar{\mathcal{B}})$. Moreover, for all databases $\mathcal{B}''$ such that $\mathcal{V}(\bar{\mathcal{B}}) \subseteq \mathcal{V}(\mathcal{B}'')$, since $\bar{\mathcal{E}} \subseteq \mathcal{V}(\mathcal{B}'')$, we have that $t \in Q_1^\Sigma(\mathcal{B}'')$. Hence we get that for the database $\bar{\mathcal{B}}$, we have $t \in cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{V}(\bar{\mathcal{B}}))$ but $t \notin cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{V}(\bar{\mathcal{B}}))$. $\Box$

Techniques for view-based query containment of Q_1^Σ in Q_2^Σ for conjunctive queries are described in [31]. It is well known that for conjunctive views $\mathcal{V}$ and a conjunctive query Q^Σ, the certain answers $cert_{Q^\Sigma,\mathcal{V}}$ can be expressed as a union of conjunctive queries, called *maximally contained rewriting* of Q wrt $\mathcal{V}$ [26]. Hence, checking $Q_1^\Sigma \subseteq_\mathcal{V} Q_2^\Sigma$ reduces to checking (traditional) containment between the maximally contained rewriting of Q_1 wrt $\mathcal{V}$ and the maximally contained rewriting of Q_2 wrt $\mathcal{V}$. Considering that the size of the maximally contained rewriting of a conjunctive query Q wrt $\mathcal{V}$ is in general exponential in the size of Q, and that a union of conjunctive queries $\cup_i q_i$ is contained in a union of conjunctive queries $\cup_j r_j$, if and only if for every i there is some j such that q_i is contained in r_j, one gets the upper bound given in [31]. In the same paper, also a matching lower bound is proved, thus getting the following computational complexity characterization.

Theorem 6 ([31]). *Checking $Q_1^\Sigma \subseteq_\mathcal{V} Q_2^\Sigma$ is Π_2^P-complete for conjunctive queries.*

Next we study the problem for 2RPQs. By Proposition 4, we can concentrate on checking whether for all $\mathcal{V}$-extensions $\mathcal{E}$ we have that $cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{E}) \subseteq cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{E})$. By Theorem 3, we know for a 2RPQ Q^Σ that (c,d) is not in $cert_{Q^\Sigma,\mathcal{V}}(\mathcal{E})$ if and only if $\mathcal{E}^{c,d}$ is in $CSP(CT_{Q^\Sigma,\mathcal{V}})$. Therefore, $Q_1^\Sigma \subseteq_\mathcal{V} Q_2^\Sigma$ if and only if for all $\mathcal{V}$-extensions $\mathcal{E}$, and for all objects c,d, we have that $\mathcal{E}^{c,d}$ in $CSP(CT_{Q_2^\Sigma,\mathcal{V}})$ implies $\mathcal{E}^{c,d}$ in $CSP(CT_{Q_1^\Sigma,\mathcal{V}})$.

In order to perform such a check we resort to a suitable variant of containment between CSPs (see, Theorem 1). Given a constraint template $CT_{Q^\Sigma,\mathcal{V}}$, a *proper constraint template* $CT^{\alpha,\beta}_{Q^\Sigma,\mathcal{V}}$ is obtained by eliminating from $\mathcal{U}_i$ all but one element α and from $\mathcal{U}_f$ all but one element β. Observe that, in general, there are many proper constraint templates for each constraint template $CT_{Q^\Sigma,\mathcal{V}}$ (more precisely there are $|2^\mathcal{S} \times 2^\mathcal{S}|$, where $\mathcal{S}$ is the set of states of the automaton for Q^Σ). Observe that a proper constraint template is itself a constraint instance of the original constraint template.

THEOREM 7. *For 2RPQs, $Q_1^\Sigma \subseteq_\mathcal{V} Q_2^\Sigma$ if and only if, for every proper constraint template $CT^{\alpha,\beta}_{Q_2^\Sigma,\mathcal{V}}$ of $CT_{Q_2^\Sigma,\mathcal{V}}$, there is a homomorphism from $CT^{\alpha,\beta}_{Q_2^\Sigma,\mathcal{V}}$ to $CT_{Q_1^\Sigma,\mathcal{V}}$.*

PROOF. By Theorem 3, it suffices to verify that, if for all $\mathcal{V}$-extensions $\mathcal{E}$ and for all objects c,d, we have that $\mathcal{E}^{c,d}$ in $CSP(CT_{Q_2^\Sigma,\mathcal{V}})$ implies $\mathcal{E}^{c,d}$ in $CSP(CT_{Q_1^\Sigma,\mathcal{V}})$, then for every proper constraint template $CT^{\alpha,\beta}_{Q_2^\Sigma,\mathcal{V}}$ of $CT_{Q_2^\Sigma,\mathcal{V}}$ there exists a homomorphism from $CT^{\alpha,\beta}_{Q_2^\Sigma,\mathcal{V}}$ to $CT_{Q_1^\Sigma,\mathcal{V}}$, and vice-versa.

"$\Leftarrow$" Suppose for every proper constraint template $CT^{\alpha,\beta}_{Q_2^\Sigma,\mathcal{V}}$ of $CT_{Q_2^\Sigma,\mathcal{V}}$ there exists a homomorphism from $CT^{\alpha,\beta}_{Q_2^\Sigma,\mathcal{V}}$ to $CT_{Q_1^\Sigma,\mathcal{V}}$. Consider an extension $\mathcal{E}^{c,d}$ in $CSP(CT_{Q_2^\Sigma,\mathcal{V}})$, i.e, there is a homomorphism g from $\mathcal{E}^{c,d}$ to $CT_{Q_2^\Sigma,\mathcal{V}}$. Consider the proper constraint template $CT^{g(c),g(d)}_{Q_2^\Sigma,\mathcal{V}}$. By hypothesis there is an homomorphism h from $CT^{g(c),g(d)}_{Q_2^\Sigma,\mathcal{V}}$ to $CT_{Q_1^\Sigma,\mathcal{V}}$. Then $g \circ h$ is an homomorphism from $\mathcal{E}^{c,d}$ to $CT_{Q_1^\Sigma,\mathcal{V}}$.

"$\Rightarrow$" Every proper constraint template $CT^{\alpha,\beta}_{Q_2^\Sigma,\mathcal{V}}$ of $CT_{Q_2^\Sigma,\mathcal{V}}$ is obviously in $CSP(CT_{Q_2^\Sigma,\mathcal{V}})$. Since by hypothesis every constraint instance in $CSP(CT_{Q_2^\Sigma,\mathcal{V}})$ is also in $CSP(CT_{Q_1^\Sigma,\mathcal{V}})$, we have that $CT^{\alpha,\beta}_{Q_2^\Sigma,\mathcal{V}}$ is in $CSP(CT_{Q_1^\Sigma,\mathcal{V}})$, i.e, there is an homomorphism from $CT^{\alpha,\beta}_{Q_2^\Sigma,\mathcal{V}}$ to $CT_{Q_1^\Sigma,\mathcal{V}}$. $\square$

For the following lemma, we exploit the characterization of $Q_1^\Sigma \subseteq_\mathcal{V} Q_2^\Sigma$, in Theorem 7.

LEMMA 8. *Checking $Q_1^\Sigma \subseteq_\mathcal{V} Q_2^\Sigma$ is in NEXPTIME for 2RPQs.*

PROOF. We have to check whether every proper constraint template $CT^{\alpha,\beta}_{Q_2^\Sigma,\mathcal{V}}$ of $CT_{Q_2^\Sigma,\mathcal{V}}$ is in $CSP(CT_{Q_1^\Sigma,\mathcal{V}})$. Both constructing the constraint template $CT_{Q_1^\Sigma,\mathcal{V}}$ and constructing each proper constraint template $CT^{\alpha,\beta}_{Q_2^\Sigma,\mathcal{V}}$ is exponential in the size of Q_1^Σ and Q_2^Σ respectively, and polynomial in the size of the definitions of the views in $\mathcal{V}$. Checking the existence of each homomorphism is NP in the size of the $CT_{Q_1^\Sigma,\mathcal{V}}$. Moreover, the number of proper constraint templates is $|2^\mathcal{S} \times 2^\mathcal{S}|$, where $\mathcal{S}$ is the set of states of the automaton for Q_2^Σ, and hence it is exponential in the size of Q_2^Σ. These bounds give us the claim. $\square$

The next lemma gives us a matching lower-bound.

LEMMA 9. *Checking $Q_1^\Sigma \subseteq_\mathcal{V} Q_2^\Sigma$ is NEXPTIME-hard for RPQs.*

PROOF. In Lemma 20, we will prove NEXPTIME-hardness of $Q_1^\Sigma \subseteq_\mathcal{V} Q_2^\mathcal{V}$ for RPQs. Consider the RPQ $Q_2^\mathcal{V}$ in that proof, and let Q_2^Σ be the RPQ obtained by replacing in $Q_2^\mathcal{V}$ the view symbols by their definitions. It is easy to see that $Q_2^\mathcal{V}(\mathcal{E}) = cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{E})$. Thus, the reduction in Lemma 20 also proves NEXPTIME-hardness for the containment of $cert_{Q_1^\Sigma,\mathcal{V}}$ in $cert_{Q_2^\Sigma,\mathcal{V}}$. $\square$

From Lemmas 8 and 9, we get the following computational complexity characterization.

THEOREM 10. *Checking $Q_1^\Sigma \subseteq_\mathcal{V} Q_2^\Sigma$ is NEXPTIME-complete both for 2RPQs and for RPQs.*

5. View-based containment of $Q_1^\mathcal{V}$ in Q_2^Σ

In this section we study view-based containment of a query $Q_1^\mathcal{V}$ expressed on the view alphabet in a query Q_2^Σ expressed on the base alphabet. Intuitively, we want to check whether, for all view extensions $\mathcal{E}$ that are sound with respect to some database, the evaluation of $Q_1^\mathcal{V}$ over $\mathcal{E}$ yields a subset of the certain answers of Q_1^Σ with respect to $\mathcal{V}$ and $\mathcal{E}$.

DEFINITION 2. *$Q_1^\mathcal{V} \subseteq_\mathcal{V} Q_2^\Sigma$ if for every database $\mathcal{B}$, and for every $\mathcal{V}$-extension $\mathcal{E}$ with $\mathcal{E} \subseteq \mathcal{V}(\mathcal{B})$, we have $Q_1^\mathcal{V}(\mathcal{E}) \subseteq cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{E})$.*

Again, the above definition considers only $\mathcal{V}$-extensions that are sound with respect to some database. However, as for the case of previous section, it turns out that we can drop this requirement w.l.o.g.

PROPOSITION 11. *$Q_1^\mathcal{V} \subseteq_\mathcal{V} Q_2^\Sigma$ if and only if for every $\mathcal{V}$-extension $\mathcal{E}$ we have $Q_1^\mathcal{V}(\mathcal{E}) \subseteq cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{E})$.*

PROOF. We show that the following two conditions are equivalent:

1. for all databases $\mathcal{B}$, for all $\mathcal{V}$-extensions $\mathcal{E} \subseteq \mathcal{V}(\mathcal{B})$, $Q_1^\mathcal{V}(\mathcal{E}) \subseteq cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{E})$
2. for all $\mathcal{V}$-extensions $\mathcal{E}$, $Q_1^\mathcal{V}(\mathcal{E}) \subseteq cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{E})$.

(2) implies (1): obvious.

(1) implies (2): by contradiction. Suppose that for every database $\mathcal{B}$, and for every $\mathcal{V}$-extension $\mathcal{E}$ with $\mathcal{E} \subseteq \mathcal{V}(\mathcal{B})$, we have that $Q_1^{\mathcal{V}}(\mathcal{E}) \subseteq cert_{Q_2^{\Sigma},\mathcal{V}}(\mathcal{E})$, but there exists an $\bar{\mathcal{E}}$ such that $Q_1^{\mathcal{V}}(\bar{\mathcal{E}}) \not\subseteq cert_{Q_2^{\Sigma},\mathcal{V}}(\bar{\mathcal{E}})$. The latter implies that there is a tuple t of objects in $\bar{\mathcal{E}}$ such that $t \in Q_1^{\mathcal{V}}(\bar{\mathcal{E}})$ and for a database $\bar{\mathcal{B}}$ with $\bar{\mathcal{E}} \subseteq \mathcal{V}(\bar{\mathcal{B}})$ we have $t \notin Q_2^{\Sigma}(\bar{\mathcal{B}})$. But this contradicts the hypothesis. □

As for the case of Section 4, one could also conceive a different notion of containment of $Q_1^{\mathcal{V}}$ in Q_2^{Σ} than the one given in Definition 2, namely: for every database $\mathcal{B}$, we have that $Q_1^{\mathcal{V}}(\mathcal{V}(\mathcal{B}))$ is a subset of $cert_{Q_2^{\Sigma},\mathcal{V}}(\mathcal{V}(\mathcal{B}))$. In general this notion differs from the one given in Definition 2, as shown by the following example.

Let the base alphabet be $\Sigma = \{R\}$ and the views be $\mathcal{V} = \{V_1, V_2\}$ with definitions $V_1^{\Sigma} = V_2^{\Sigma} = R$. Consider the two unary queries

$$\begin{aligned} Q_1^{\mathcal{V}} &= \{x \mid V_1(x), \neg V_2(x)\} \\ Q_2^{\Sigma} &= \emptyset \end{aligned}$$

It is easy to see that, for every database $\mathcal{B}$, we have $Q_1^{\mathcal{V}}(\mathcal{V}(\mathcal{B})) = \emptyset$, and hence $Q_1^{\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq cert_{Q_2^{\Sigma},\mathcal{V}}(\mathcal{V}(\mathcal{B}))$ (in fact, also $cert_{Q_2^{\Sigma},\mathcal{V}}(\mathcal{V}(\mathcal{B})) = \emptyset$). However, consider the database $\bar{\mathcal{B}}$ with $R^{\bar{\mathcal{B}}} = \{(a,b)\}$ and the $\mathcal{V}$-extension $\bar{\mathcal{E}}$ with $V_1^{\bar{\mathcal{E}}} = \{(a,b)\}$ and $V_2^{\bar{\mathcal{E}}} = \emptyset$, which is such that $\bar{\mathcal{E}} \subseteq \mathcal{V}^{\Sigma}(\bar{\mathcal{B}})$. For such a database and view extension we have $Q_1^{\mathcal{V}}(\bar{\mathcal{E}}) = \{(a,b)\}$, and hence $Q_1^{\mathcal{V}}(\bar{\mathcal{E}}) \not\subseteq cert_{Q_2^{\Sigma},\mathcal{V}}(\bar{\mathcal{E}})$.

However it turns out that, if $Q_1^{\mathcal{V}}$ is monotone, the two notions are equivalent.

PROPOSITION 12. *Let $Q_1^{\mathcal{V}}$ be monotone. Then, $Q_1^{\mathcal{V}} \subseteq_{\mathcal{V}} Q_2^{\Sigma}$ if and only if for every database $\mathcal{B}$, we have $Q_1^{\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq cert_{Q_2^{\Sigma},\mathcal{V}}(\mathcal{V}(\mathcal{B}))$.*

PROOF. By Proposition 11, it suffices to prove that the following two conditions are equivalent:

1. for all $\mathcal{V}$-extensions $\mathcal{E}$ we have that $Q_1^{\mathcal{V}}(\mathcal{E}) \subseteq cert_{Q_2^{\Sigma},\mathcal{V}}(\mathcal{E})$
2. for all databases $\mathcal{B}$, we have that $Q_1^{\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq cert_{Q_2^{\Sigma},\mathcal{V}}(\mathcal{V}(\mathcal{B}))$

(1) implies (2): straightforward, since for each database $\mathcal{B}$, $\mathcal{V}(\mathcal{B})$ is a $\mathcal{V}$-extension.

(2) implies (1): we show that not (1) implies not (2). Assume that there exists a $\mathcal{V}$-extension $\bar{\mathcal{E}}$ and a tuple of objects t such that $t \in Q_1^{\mathcal{V}}(\mathcal{E})$, and there exists a database $\bar{\mathcal{B}}'$ such that $\bar{\mathcal{E}} \subseteq \mathcal{V}(\bar{\mathcal{B}}')$ and $t \notin Q_2^{\Sigma}(\bar{\mathcal{B}}')$. Observe that, *by monotonicity of $Q_1^{\mathcal{V}}$*, we have that $t \in Q_1^{\mathcal{V}}(\mathcal{V}(\bar{\mathcal{B}}'))$. Hence the database $\bar{\mathcal{B}}'$ is such that $Q_1^{\mathcal{V}}(\mathcal{V}(\bar{\mathcal{B}}')) \not\subseteq cert_{Q_2^{\Sigma},\mathcal{V}}(\mathcal{V}(\bar{\mathcal{B}}'))$. □

Interestingly, it turns out that, if both queries $Q_1^{\mathcal{V}}$ and Q_2^{Σ} are monotone, the condition in the proposition above is equivalent to a simpler condition, namely: for every database $\mathcal{B}$ we have that $Q_1^{\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq Q_2^{\Sigma}(\mathcal{B})$. This is stated in the following proposition.

PROPOSITION 13. *Let $Q_1^{\mathcal{V}}$ and Q_2^{Σ} be monotone. Then, $Q_1^{\mathcal{V}} \subseteq_{\mathcal{V}} Q_2^{\Sigma}$ if and only if for every database $\mathcal{B}$, we have $Q_1^{\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq Q_2^{\Sigma}(\mathcal{B})$.*

PROOF. Since $Q_1^{\mathcal{V}}$ *is monotone*, by Propositions 11 and 12, it suffices to prove that the following two conditions are equivalent:

1. for all databases $\mathcal{B}$, we have that $Q_1^{\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq cert_{Q_2^{\Sigma},\mathcal{V}}(\mathcal{V}(\mathcal{B}))$
2. for all databases $\mathcal{B}$, we have that $Q_1^{\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq Q_2^{\Sigma}(\mathcal{B})$

(1) implies (2): straightforward, since by definition of certain answers, for all databases $\mathcal{B}$ we have that $cert_{Q_2^{\Sigma},\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq Q_2^{\Sigma}(\mathcal{B})$.

(2) implies (1): we show that not (1) implies not (2). Assume that there exists a database $\bar{\mathcal{B}}$ and a tuple of objects t such that $t \in Q_1^{\mathcal{V}}(\mathcal{V}(\bar{\mathcal{B}}))$, and there exists a database $\bar{\mathcal{B}}'$ such that $\mathcal{V}(\bar{\mathcal{B}}) \subseteq \mathcal{V}(\bar{\mathcal{B}}')$ and $t \notin Q_2^{\Sigma}(\bar{\mathcal{B}}')$. Observe that, *by monotonicity of Q_2^{Σ}*, we have that $t \notin Q_2^{\Sigma}(\mathcal{V}(\bar{\mathcal{B}}))$. Hence for the database $\bar{\mathcal{B}}$ we have that $t \in Q_1^{\mathcal{V}}(\mathcal{V}(\bar{\mathcal{B}}))$ and $t \notin Q_2^{\Sigma}(\mathcal{V}(\bar{\mathcal{B}}))$, thus getting the thesis. □

Since conjunctive queries are monotone, the above theorem implies that, for conjunctive queries, checking $Q_1^{\mathcal{V}} \subseteq_{\mathcal{V}} Q_2^{\Sigma}$ simply amounts to checking whether for every $\mathcal{B}$, $Q_1^{\mathcal{V}}(\mathcal{V}(B)) \subseteq Q_2^{\Sigma}(\mathcal{B})$. This is equivalent to substituting the view symbols appearing in $Q_1^{\mathcal{V}}$ with the corresponding view definitions, thus getting a new conjunctive query Q_1^{Σ} over the base alphabet Σ, and then checking $Q_1^{\Sigma} \subseteq Q_2^{\Sigma}$.

THEOREM 14. *Checking $Q_1^{\mathcal{V}} \subseteq_{\mathcal{V}} Q_2^{\Sigma}$ is NP-complete for conjunctive queries.*

Now we turn to the problem of checking whether $Q_1^{\mathcal{V}} \subseteq_{\mathcal{V}} Q_2^{\Sigma}$ in the setting of 2RPQs. As for the case of conjunctive queries, since 2RPQs are monotone, checking view-based containment of $Q_1^{\mathcal{V}}$ in Q_2^{Σ} reduces to checking whether for every $\mathcal{B}$, $Q_1^{\mathcal{V}}(\mathcal{V}(B)) \subseteq Q_2^{\Sigma}(\mathcal{B})$. Again, this is equivalent to substituting views with their definitions, thus getting a new 2RPQ Q_1^{Σ}, and then checking $Q_1^{\Sigma} \subseteq Q_2^{\Sigma}$.

THEOREM 15. *Checking $Q_1^{\mathcal{V}} \subseteq_{\mathcal{V}} Q_2^{\Sigma}$ is PSPACE-complete both for 2RPQs and for RPQs.*

PROOF. Since 2RPQs are monotone, by Proposition 13, checking $Q_1^{\mathcal{V}} \subseteq_{\mathcal{V}} Q_2^{\Sigma}$ is equivalent to checking whether $Q_1^{\Sigma} \subseteq Q_2^{\Sigma}$, where Q_1^{Σ} is the 2RPQ over Σ obtained from $Q_1^{\mathcal{V}}$ by substituting each view symbol V with its definition

V^{Σ}. The upper bound follows from the fact that containment between two 2RPQs can be checked in PSPACE [15]. The lower bound follows from PSPACE-hardness of containment of regular expressions, considering a set of views that coincide with the base symbols. $\square$

6. View-based containment of Q_1^{Σ} in $Q_2^{\mathcal{V}}$

We address the problem of view-based containment between a query over the base alphabet and a query over the view alphabet.

DEFINITION 3. *$Q_1^{\Sigma} \subseteq_{\mathcal{V}} Q_2^{\mathcal{V}}$ if for every database $\mathcal{B}$, and for every $\mathcal{V}$-extension $\mathcal{E} \subseteq \mathcal{V}(\mathcal{B})$, we have $cert_{Q_1^{\Sigma},\mathcal{V}}(\mathcal{E}) \subseteq Q_2^{\mathcal{V}}(\mathcal{E})$.*

Again in this definition we consider only $\mathcal{V}$-extensions that are sound wrt some database. However, we can drop this requirement w.l.o.g. for non-constraining views.

PROPOSITION 16. *Let $\mathcal{V}$ be a set of non-constraining views. Then, $Q_1^{\Sigma} \subseteq_{\mathcal{V}} Q_2^{\mathcal{V}}$ if and only if for every $\mathcal{V}$-extension $\mathcal{E}$, we have $cert_{Q_1^{\Sigma},\mathcal{V}}(\mathcal{E}) \subseteq Q_2^{\mathcal{V}}(\mathcal{E})$*

PROOF. "$\Leftarrow$" Obvious. "$\Rightarrow$" By contradiction. Suppose that for each database $\mathcal{B}$ and for each $\mathcal{V}$-extensions $\mathcal{E}$ with $\mathcal{E} \subseteq \mathcal{V}(\mathcal{B})$, we have that $cert_{Q_1^{\Sigma},\mathcal{V}}(\mathcal{E}) \subseteq Q_2^{\mathcal{V}}(\mathcal{E})$, but suppose there exists an $\bar{\mathcal{E}}$ such that $cert_{Q_1^{\Sigma},\mathcal{V}}(\bar{\mathcal{E}}) \not\subseteq Q_2^{\mathcal{V}}(\bar{\mathcal{E}})$. Now, since views are non-constraining, there exists a database $\bar{\mathcal{B}}$ such that $\bar{\mathcal{E}} \subseteq \mathcal{V}(\bar{\mathcal{B}})$. Hence $\bar{\mathcal{E}}$ contradicts the hypothesis. $\square$

For conjunctive queries, we know that $cert_{Q_1^{\Sigma},\mathcal{V}}$ coincides with the maximally contained rewriting of Q_1^{Σ} wrt $\mathcal{V}$, which is a possibly exponential union of conjunctive queries, each of linear size in Q_1^{Σ} [23]. To check non-containment $cert_{Q_1^{\Sigma},\mathcal{V}} \not\subseteq Q_2^{\mathcal{V}}$, we have to guess a conjunctive query in the maximal rewriting of Q_1^{Σ}, and verify that it is not contained in $Q_2^{\mathcal{V}}$. Hence we get the following upper bound.

THEOREM 17. *Checking $Q_1^{\Sigma} \subseteq_{\mathcal{V}} Q_2^{\mathcal{V}}$ is in Π_2^P for conjunctive queries.*

Whether this bound is tight is an open problem.

Now we turn to the problem of checking whether $Q_1^{\Sigma} \subseteq_{\mathcal{V}} Q_2^{\mathcal{V}}$, in the setting of 2RPQs. Since 2RPQ views are non-constraining, by Proposition 16, it is sufficient to check whether $cert_{Q_2^{\Sigma},\mathcal{V}}$ is contained in $Q_1^{\mathcal{V}}$.

By Theorem 3, we know for a 2RPQ Q^{Σ} that (c,d) is not in $cert_{Q^{\Sigma},\mathcal{V}}(\mathcal{E})$ iff $\mathcal{E}^{c,d}$ is in $CSP(CT_{Q^{\Sigma},\mathcal{V}})$. On the other hand, by Theorem 2, we know for a query $Q^{\mathcal{V}}$ that (c,d) is not in $Q^{\mathcal{V}}(\mathcal{E})$ iff $\mathcal{E}^{c,d}$ is in $CSP(CT_{Q^{\mathcal{V}},\mathcal{V}})$. Therefore, $Q_1^{\Sigma} \subseteq_{\mathcal{V}} Q_2^{\mathcal{V}}$ if and only if for every $\mathcal{V}$-extension $\mathcal{E}$, and for every pair objects c and d, we have that $\mathcal{E}^{c,d}$ in $CSP(CT_{Q_2^{\mathcal{V}},\mathcal{V}})$ implies $\mathcal{E}^{c,d}$ in $CSP(CT_{Q_1^{\Sigma},\mathcal{V}})$.

In order to perform such a check we again resort to a suitable variant of containment between CSPs (see Theorem 1). Again, we introduce proper constraint templates. Given a constraint template $CT_{Q^{\mathcal{V}},\mathcal{V}}$, a *proper constraint template* $CT^{\alpha,\beta}_{Q^{\mathcal{V}},\mathcal{V}}$ is obtained by eliminating from $\mathcal{U}_i$ all but one element α and from $\mathcal{U}_f$ all but one element β.

THEOREM 18. *For 2RPQs, $Q_1^{\Sigma} \subseteq_{\mathcal{V}} Q_2^{\mathcal{V}}$ if and only if for every proper constraint template $CT^{\alpha,\beta}_{Q_2^{\mathcal{V}},\mathcal{V}}$ of $CT_{Q_2^{\mathcal{V}},\mathcal{V}}$ there exists a homomorphism from $CT^{\alpha,\beta}_{Q_2^{\mathcal{V}},\mathcal{V}}$ to $CT_{Q_1^{\Sigma},\mathcal{V}}$.*

PROOF. Analogous to the proof of Theorem 7. $\square$

For the following lemma, we exploit the above characterization of $Q_1^{\Sigma} \subseteq_{\mathcal{V}} Q_2^{\mathcal{V}}$.

LEMMA 19. *Checking $Q_1^{\Sigma} \subseteq_{\mathcal{V}} Q_2^{\mathcal{V}}$ is in NEXPTIME for 2RPQs.*

PROOF. Analogous to the proof of Theorem 8. $\square$

The next lemma gives us a matching lower-bound.

LEMMA 20. *Checking $Q_1^{\Sigma} \subseteq_{\mathcal{V}} Q_2^{\mathcal{V}}$ is NEXPTIME-hard for RPQs.*

PROOF. We show a reduction from a bounded version of the *tiling problem* [36]. The problem is defined as follows. A *tile system* Ω consists of a finite tile set T, two adjacency relations $V \subseteq T \times T$ and $H \subseteq T \times T$, an initial tile τ_0, and a final tile τ_f. In a bounded tiling problem we are given such a tile system Ω and a bound $n > 0$ (in unary). We have to decide whether there is a *proper* tiling of the $2^n \times 2^n$-grid such that: (1) t_0 is in the bottom left corner and t_f is in the top left corner, (2) every pair of horizontal neighbors is in H, and (3) every pair of vertical neighbors is in V. Formally, we have to decide whether there is a tiling function $\alpha : 2^n \times 2^n \to T$ such that (1) $\alpha(0,0) = \tau_0$ and $t(0, 2^n - 1) = \tau_f$, (2) for every $0 \leq i < 2^n - 1$, and $0 \leq j < 2^n$, we have that $(\alpha(i,j), \alpha(i+1,j)) \in H$, and (3) for every $0 \leq i < 2^n$, and $0 \leq j < 2^n - 1$, we have that $(\alpha(i,j), \alpha(i,j+1)) \in V$ and also $\alpha(2^n-1,j), \alpha(0,j+1)) \in H$. This problem is NEXPTIME-complete [29, 34]. We reduce it to our containment problem. We construct a set $\mathcal{V}$ of RPQ views and RPQs Q_1^{Σ} and $Q_2^{\mathcal{V}}$ such that $cert_{Q_1^{\Sigma},\mathcal{V}}$ is contained in $Q_2^{\mathcal{V}}$ iff the tiling problem is satisfiable.

We encode grids by labeled directed graphs, using the set of views $\mathcal{V} = \{s, f, h, v, 0, 1, t\}$. The intuition is that s labels the starting edge of the grid, f labels the final edge of the grid, h label horizontal adjacency edges, v labels vertical adjacency edges, 0 and 1 label numerical edges (representing bits of the coordinates), and t labels tile edges. We intend the whole grid to be represented by a path of the form $s((0+1)^{2n}th)^{4^n-1}(0+1)^{2n}tf$, where each block of the form $(0+1)^{2n}t$ represents one point in the grid and the sequence of 4^n blocks of the form $(0+1)^{2n}$ represents a $2n$-bit counter.

Note that h-edges connect horizontally adjacent blocks, i.e., either $i' = i+1$ and $j' = j$ or $i = 2^n - 1$, $i' = 0$, and $j' = j+1$. We also intend that if there are v-edges between nodes on this path they should connect vertically adjacent blocks, i.e., $i' = i$ and $j' = j+1$. We call a path of this form a *good path*.

Good paths cannot be described by succinct regular expressions. We can, however, write a regular expression $Q_2^{\mathcal{V}}$ of size polynomial in n that describes *bad paths*, i.e.:

- paths that do not contain v-edges but are not of the form $s((0+1)^{2n}th)^*(0+1)^{2n}tf$,
- paths of the form $s((0+1)^{2n}th)^*(0+1)^{2n}tf$, where the first block is not 0^{2n} or the last block is not 1^{2n},
- paths of the form $s((0+1)^{2n}th)^*(0+1)^{2n}tf$, where an h-edge connects blocks that are not horizontally adjacent, or
- paths of the form $s((0+1)^{2n}th)^*(0+1)^{2n}tv((0+1)^{2n}th)^*(0+1)^{2n}tf$, where the v-edge connects blocks that are not vertically adjacent (we call these *bad v-edges*).

Let the base alphabet be $\Sigma = \{s, f, h, v, 0, 1\} \cup T$. The view definitions are as follows: (1) s, f, h, v, 0, and 1 are the identity views, e.g., $s^\Sigma = s$, (2) $t^\Sigma = \mathbf{t}$, where $\mathbf{t} = \cup_{\tau \in T}\tau$. Intuitively, the view t "forgets" the precise tiling. The query Q_1^Σ is an RPQ of polynomial size that accepts paths with bad tiling:

- paths of the form $s((0+1)^*\mathbf{t}h)^*(0+1)^*\mathbf{t}f$, where the first tile is not τ_0 or the last tile is not τ_f,
- paths of the form $s((0+1)^*\mathbf{t}h)^*(0+1)^*\mathbf{t}f$, where an h-edge connects pairs of tiled blocks violating H, or
- paths of the form $s((0+1)^*\mathbf{t}h)^*(0+1)^*\mathbf{t}v((0+1)^*\mathbf{t}h)^*(0+1)^*\mathbf{t}f$, where the v-edge connects pairs of tiled blocks that violate V.

Suppose first that the given tiling problem is not satisfiable. Consider a $\mathcal{V}$-extension $\mathcal{E}$ that contains a good path from a to b, augmented by all allowable v-edges, i.e., v-edges connecting every pair of vertically adjacent blocks. Since the path is good, $(a, b) \notin Q_2^{\mathcal{V}}(\mathcal{E})$. Let B be a database such that $\mathcal{E} = \mathcal{V}(\mathcal{B})$ (observe that, because of the particular form of the views definitions, such a $\mathcal{B}$ always exists). Thus, every t-edge in $\mathcal{E}$ appears in $\mathcal{B}$ as a τ-edge for some $\tau \in T$. Thus, the path from a to b represents a candidate tiling. Since the tiling problem is not satisfiable, we know that this tiling is not a proper tiling, that is, it either does not start with τ_0, does not end with τ_f, violates H, or violates V. In all these cases we have that $(a, b) \in Q_1^\Sigma(\mathcal{B})$. It follows that $(a, b) \in cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{E})$. Thus, $cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{E})$ is not contained in $Q_2^{\mathcal{V}}$.

Suppose now that the given tiling problem is satisfiable. Consider a $\mathcal{V}$-extension $\mathcal{E}$ and a pair $(a, b) \notin Q_2^{\mathcal{V}}(\mathcal{E})$. If a and b are not connected, then $(a, b) \notin cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{E})$. If a and b are connected, then there is a good path from a to b. Let $\mathcal{B}$ be a database such that $\mathcal{E} = \mathcal{V}(\mathcal{B})$ (again such a database exists, because of the form of the view definitions). Thus, every t-edge in $\mathcal{E}$ appears in $\mathcal{B}$ as a τ-edge for some $\tau \in T$. Thus, the path from a to b represents a candidate tiling. Since the tiling problem is satisfiable, we know that there is some $\mathcal{B}$ such that this tiling is a proper tiling, that is, it does start with τ_0, end with τ_f, respects H, and respect V. It follows that $(a, b) \notin Q_1^\Sigma(\mathcal{B})$. Thus, $(a, b) \notin cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{E})$. Consequently, $cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{E})$ is contained in $Q_2^{\mathcal{V}}$. $\square$

Note that Q_1^Σ depends only on H and V but not on n. By starting with a NEXPTIME-complete Turing machine, we can obtain a fixed tiling system Ω whose satisfiability problem is NEXPTIME-complete, where in addition to n we need to specify the first n tiles on the grid. From this we get that our containment problem is NEXPTIME-complete even for fixed $\mathcal{V}$ and Q_1^Σ.

From Lemmas 19 and 20, we get the following computational complexity characterization.

THEOREM 21. *Checking $Q_1^\Sigma \subseteq_{\mathcal{V}} Q_2^{\mathcal{V}}$ is NEXPTIME-complete both for 2RPQs and for RPQs.*

One could also compare the two queries only wrt extensions that correspond exactly to the evaluation of the views over some database. Formally, this corresponds to checking whether, for every database $\mathcal{B}$, we have that $cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{V}(\mathcal{B}))$ is a subset of $Q_2^{\mathcal{V}}(\mathcal{V}(\mathcal{B}))$. Such a notion *differs* from $Q_1^\Sigma \subseteq_{\mathcal{V}} Q_2^{\mathcal{V}}$, even for non-constraining views and monotone queries. Indeed, consider $\Sigma = \{R\}$, $\mathcal{V} = \{V_1, V_2\}$ with $V_1^\Sigma = V_2^\Sigma = R$, and queries $Q_1^\Sigma = R$ and $Q_2^{\mathcal{V}} = V_2$. It is easy to see that, for every database $\mathcal{B}$ we have $cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq Q_2^{\mathcal{V}}(\mathcal{V}(\mathcal{B}))$. However, the $\mathcal{V}$-extension $\mathcal{E}$ with $V_1(\mathcal{E}) = \{(c, d)\}$ and $V_2(\mathcal{E}) = \emptyset$ is such that $(c, d) \in cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{E})$ but $(c, d) \notin Q_2^{\mathcal{V}}(\mathcal{E})$, thus showing that $Q_1^\Sigma \not\subseteq_{\mathcal{V}} Q_2^{\mathcal{V}}$.

Since the two versions of containment differ, the question arises on whether we are able to check containment of Q_1^Σ in $Q_2^{\mathcal{V}}$ only for those $\mathcal{V}$-extensions that correspond to the evaluation of the view definitions over some database. To this purpose we can exploit the following proposition.

PROPOSITION 22. *Let $Q^{\mathcal{V}}$ be monotone, and let Q^Σ be the query over Σ obtained by substituting in $Q^{\mathcal{V}}$ the view symbols with the corresponding view definitions. Then, for every database $\mathcal{B}$, we have $Q^{\mathcal{V}}(\mathcal{V}(\mathcal{B})) = cert_{Q^\Sigma,\mathcal{V}}(\mathcal{V}(\mathcal{B}))$.*

PROOF. $\subseteq$. Suppose there exists a database $\bar{\mathcal{B}}$ and a tuple t such that $t \in Q^{\mathcal{V}}$ but for some database $\mathcal{B}'$ such that $\mathcal{V}(\bar{\mathcal{B}}) \subseteq \mathcal{V}(\mathcal{B}')$, $t \notin Q^\Sigma(\mathcal{B}')$. Now $Q^\Sigma(\mathcal{B}') = Q^{\mathcal{V}}(\mathcal{V}(\mathcal{B}'))$, hence since $Q^{\mathcal{V}}$ is monotone, we get a contradiction.

$\supseteq$. Suppose there exists a database $\bar{\mathcal{B}}$ and a tuple t such that for all databases $\mathcal{B}'$ such that $\mathcal{V}(\bar{\mathcal{B}}) \subseteq \mathcal{V}(\mathcal{B}')$ we have that $t \in Q^\Sigma(\mathcal{B}')$, but $t \notin Q^{\mathcal{V}}(\mathcal{V}(\bar{\mathcal{B}}))$. Now since $Q^\Sigma(\mathcal{B}') = Q^{\mathcal{V}}(\mathcal{V}(\mathcal{B}'))$, we get a contradiction. $\square$

Observe that, in general, it is not true that, for every $\mathcal{V}$-extension $\mathcal{E}$, we have $Q^{\mathcal{V}}(\mathcal{E})) = cert_{Q^\Sigma,\mathcal{V}}(\mathcal{E})$. For example, consider $\Sigma = \{R\}$, the set of views $\mathcal{V} = \{V_1, V_2\}$ with $V_1^\Sigma = V_2^\Sigma = R$, and the $\mathcal{V}$-extension $\mathcal{E}$ with $V_1(\mathcal{E}) = \{(a, b)\}$ and $V_2(\mathcal{E}) = \emptyset$. Then, for the query $Q^{\mathcal{V}} = V_2$, we have that $Q^{\mathcal{V}}(\mathcal{E}) = \emptyset$, while $cert_{Q^\Sigma,\mathcal{V}}(\mathcal{E}) = \{(a, b)\}$.

By Proposition 22, if $Q_2^{\mathcal{V}}$ is monotone, we can check whether for every database $\mathcal{B}$ we have that $cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq Q_2^{\mathcal{V}}(\mathcal{V}(\mathcal{B}))$, by checking whether for every database $\mathcal{B}$ we have that $cert_{Q_1^\Sigma,\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{V}(\mathcal{B}))$, where Q_2^Σ is the query obtained from $Q_2^{\mathcal{V}}$ by expanding view symbols with their definitions. By Proposition 5, this can be done by checking whether $Q_1^\Sigma \subseteq_{\mathcal{V}} Q_2^\Sigma$ (cf. Section 4), and therefore the problem is Π_2^P-complete for conjunctive queries, and NEXPTIME-complete for 2RPQs and RPQs.

7. View-based containment of $Q_1^{\mathcal{V}}$ in $Q_2^{\mathcal{V}}$

The last form of view-based query containment is when the two queries are over the view alphabet.

Definition 4. $Q_1^{\mathcal{V}} \subseteq_{\mathcal{V}} Q_2^{\mathcal{V}}$ *if for every database $\mathcal{B}$, and for every $\mathcal{V}$-extension $\mathcal{E} \subseteq \mathcal{V}(\mathcal{B})$, we have $Q_1^{\mathcal{V}}(\mathcal{E}) \subseteq Q_2^{\mathcal{V}}(\mathcal{E})$.*

Analogously to the previous case, for non-constraining views, we can drop w.l.o.g. the requirement to consider only $\mathcal{V}$-extensions that are sound wrt some database. Therefore, if views are non-constraining, checking view-based containment reduces to checking containment of the two queries over the view alphabet.

Theorem 23. *Checking $Q_1^{\mathcal{V}} \subseteq_{\mathcal{V}} Q_2^{\mathcal{V}}$ is NP-complete for conjunctive queries, and PSPACE-complete both for 2RPQs and for RPQs.*

It is interesting to observe that, even for non-constraining views and monotone queries, checking $Q_1^{\mathcal{V}} \subseteq_{\mathcal{V}} Q_2^{\mathcal{V}}$ is *not* equivalent to checking whether, for every database $\mathcal{B}$, we have that $Q_1^{\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq Q_2^{\mathcal{V}}(\mathcal{V}(\mathcal{B}))$. Indeed, consider $\Sigma = \{R\}$, $\mathcal{V} = \{V_1, V_2\}$, with $V_1{}^\Sigma = R$ and $V_2{}^\Sigma = R$, and queries $Q_1^{\mathcal{V}} = V_1$ and $Q_2^{\mathcal{V}} = V_2$. It is easy to see that for every database $\mathcal{B}$, we have $Q_1^{\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq Q_2^{\mathcal{V}}(\mathcal{V}(\mathcal{B}))$. However, the $\mathcal{V}$-extension $\mathcal{E}$ with $V_1(\mathcal{E}) = \{(a, b)\}$ and $V_2(\mathcal{E}) = \emptyset$ shows that $Q_1^{\mathcal{V}} \not\subseteq_{\mathcal{V}} Q_2^{\mathcal{V}}$.

Since the two versions of containment differ, the question arises on whether we are able to check the condition that for each database $\mathcal{B}$, $Q_1^{\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq Q_2^{\mathcal{V}}(\mathcal{V}(\mathcal{B}))$. It turns out that this is equivalent to substituting view symbols with their definitions in both queries, thus getting new queries Q_1^Σ and Q_2^Σ, and then checking $Q_1^\Sigma \subseteq Q_2^\Sigma$. It follows that checking whether, for each database $\mathcal{B}$, $Q_1^{\mathcal{V}}(\mathcal{V}(\mathcal{B})) \subseteq Q_2^{\mathcal{V}}(\mathcal{V}(\mathcal{B}))$, is NP-complete for conjunctive queries, and PSPACE-complete for 2RPQs and RPQs.

8. Relationship with Rewriting

Although in the previous sections we have referred to view-based query answering, another approach to view-based query processing has been advocated, namely the one based on rewriting. In view-based query rewriting, a query Q^Σ over the base alphabet is processed by first reformulating Q^Σ into an expression of a fixed language (e.g., union of conjunctive queries) over the view alphabet, and then evaluating such an expression over the view extensions. The relationship between view-based query answering and view-based query rewriting is investigated in [23, 13, 25].

Note that, in our setting, views are sound, and this property must be taken into account in the reformulation step of the rewriting process. However, most papers on rewriting queries using views are based, either implicitly or explicitly, on the exact view assumption, which states that the extension of the views provides exactly the results of applying the view definitions to the database. It follows that we need to provide an adequate definition of rewriting in a setting where views are sound.

Definition 5. $Q_1^{\mathcal{V}}$ *is a* rewriting *of Q_2^Σ* under sound views *$\mathcal{V}$, if for every database $\mathcal{B}$ and for every $\mathcal{V}$-extension $\mathcal{E}$ with $\mathcal{E} \subseteq \mathcal{V}(\mathcal{B})$, we have that $Q_1^{\mathcal{V}}(\mathcal{E}) \subseteq Q_2^\Sigma(\mathcal{B})$.*

Two interesting observations easily follow from this definition. First, it is immediate to verify that $cert_{Q^\Sigma,\mathcal{V}}$, when seen as a query over the view alphabet, conforms to the definition, and therefore *is* a rewriting of Q^Σ under sound views $\mathcal{V}$. Second, if $Q_1^{\mathcal{V}}$ is a rewriting of Q_2^Σ under sound views $\mathcal{V}$, then for each $\mathcal{V}$-extension $\mathcal{E}$ with $\mathcal{E} \subseteq \mathcal{V}(\mathcal{B})$, we have that $Q_1^{\mathcal{V}}(\mathcal{E}) \subseteq cert_{Q_2^\Sigma,\mathcal{V}}(\mathcal{E})$.

It is clear from the above observations that, if we do not have any constraint on the language used to express the rewriting, $cert_{Q^\Sigma,\mathcal{V}}$ is the best rewriting of Q^Σ one can obtain. This motivates why, in [13], $cert_{Q^\Sigma,\mathcal{V}}$ has been called the *perfect rewriting of Q^Σ under sound views $\mathcal{V}$*. On the other hand, when the rewriting is restricted to belong to a certain query class $\mathcal{C}$, one is typically interested in a so-called maximally contained rewriting, i.e., a rewriting that belongs to $\mathcal{C}$, and is maximal in $\mathcal{C}$. The maximally contained rewriting is obviously the best possible rewriting in the class $\mathcal{C}$.

Now, given a query $Q_1^{\mathcal{V}}$ over the view alphabet and a query Q_2^Σ over the base alphabet, three interesting questions related to view-based query rewriting arise:

1. Is $Q_1^{\mathcal{V}}$ a rewriting of Q_2^Σ under sound views $\mathcal{V}$?
2. Is $Q_1^{\mathcal{V}}$ the perfect rewriting of Q_2^Σ under sound views $\mathcal{V}$?
3. For a class $\mathcal{C}$ of rewritings, is the maximally contained rewriting of Q_2^Σ perfect?

We show in the following that view-based query containment is the right conceptual tool to provide the answers to the above questions.

Starting from the 1st question, the following proposition shows that view-based query containment provides a method for checking whether $Q_1^{\mathcal{V}}$ is a rewriting of Q_2^Σ under sound views $\mathcal{V}$. The proposition is an easy consequence of the definition of rewriting under sound views, and Proposition 11.

PROPOSITION 24. *$Q_1^{\mathcal{V}}$ is a rewriting of Q_2^{Σ} under sound views $\mathcal{V}$ if and only if $Q_1^{\mathcal{V}} \subseteq cert_{Q_2^{\Sigma},\mathcal{V}}$, if and only if $Q_1^{\mathcal{V}} \subseteq_{\mathcal{V}} Q_2^{\Sigma}$.*

By exploiting this property and the results presented in Section 5 on view-based containment of $Q_1^{\mathcal{V}}$ in Q_2^{Σ}, we derive the following complexity result.

THEOREM 25. *Checking whether $Q_1^{\mathcal{V}}$ is a rewriting of Q_2^{Σ} under sound views $\mathcal{V}$ is NP-complete for conjunctive queries, and PSPACE-complete both for 2RPQs and for RPQs.*

To address the 2nd question, we need to check both that $Q_1^{\mathcal{V}}$ is a rewriting of Q_2^{Σ} under sound views $\mathcal{V}$ (question 1), and that the perfect rewriting of Q_2^{Σ} under sound views $\mathcal{V}$ is contained in $Q_1^{\mathcal{V}}$. The latter amounts to verifying whether for every database $\mathcal{B}$, and for every $\mathcal{V}$-extension $\mathcal{E}$ with $\mathcal{E} \subseteq \mathcal{V}(\mathcal{B})$, we have that $cert_{Q_2^{\Sigma},\mathcal{V}}(\mathcal{E}) \subseteq Q_1^{\mathcal{V}}(\mathcal{E})$. The next proposition shows that we can again use view-based query containment for this purpose.

PROPOSITION 26. *The perfect rewriting of Q_2^{Σ} under sound views $\mathcal{V}$ is contained in $Q_1^{\mathcal{V}}$ if and only if $Q_2^{\Sigma} \subseteq_{\mathcal{V}} Q_1^{\mathcal{V}}$.*

From the results presented in Section 6 on view-based containment of Q_2^{Σ} in $Q_1^{\mathcal{V}}$, the theorem below directly follows.

THEOREM 27. *Checking whether the perfect rewriting of Q_2^{Σ} under sound views $\mathcal{V}$ is contained in $Q_1^{\mathcal{V}}$ can be done in Π_2^P for conjunctive queries, and is NEXPTIME-complete both for 2RPQs and for RPQs.*

Propositions 24 and 26 show that, both for conjunctive queries, and for 2RPQs, we have a method for checking whether a given query $Q_1^{\mathcal{V}}$ is the perfect rewriting of Q_2^{Σ} under sound views $\mathcal{V}$.

Turning our attention to the 3rd question, observe first that it may happen that, for a class $\mathcal{C}$, no maximally contained rewriting of Q^{Σ} coincides with the perfect rewriting of Q^{Σ} under sound views $\mathcal{V}$. A method for computing maximally contained conjunctive rewritings for conjunctive queries and views is described in [26]. Analogously, a method for computing maximally contained two-way regular rewritings for 2RPQs queries and views is described in [12]. By resorting to these results, and by referring to Theorem 27, we can conclude that, for both classes of queries, we have an upper bound for checking whether the maximally contained rewriting of a query Q_2^{Σ} under sound views $\mathcal{V}$ is perfect.

Finally, perfectness should not be confused with exactness. A rewriting of a query is called *exact* if it is equivalent to the query itself. Although every exact rewriting is also perfect, it may happen that the perfect rewriting of a query Q^{Σ} under sound views $\mathcal{V}$ is not exact. In [14], a set of views $\mathcal{V}$ is called *lossless* wrt a query Q^{Σ} if the perfect rewriting of Q^{Σ} under sound views $\mathcal{V}$ is exact. In the same paper, a method for checking losslessness is presented for RPQs. It is easy to see that checking losslessness amounts to checking containment between Q^{Σ} and $cert_{Q^{\Sigma},\mathcal{V}}$, and therefore is not captured by the notion of view-based query containment studied in this paper.

9. Conclusions

We have presented a thorough analysis of view-based query containment, showing that the problem comes in various forms, depending on whether each of the two queries is expressed over the base alphabet or the view alphabet. We have investigated all possible combinations from a semantic point of view, and studied their mutual relationships. For both settings of conjunctive queries and of two-way regular path queries, we have provided techniques and complexity bounds for the different variants of view-based query containment. The only problem left open by our analysis is the exact lower bound of checking $Q_1^{\mathcal{V}} \subseteq_{\mathcal{V}} Q_2^{\Sigma}$ for conjunctive queries. Finally, we have studied the relationship between view-based query containment and view-based query rewriting.

In our investigation, we have considered the case where views are sound, i.e., their extensions provide a subset of the results of applying the view definitions to the database. Although this is the usual assumption in most domains, e.g., data integration, there are settings where views should be considered exact, e.g., query optimization. An interesting direction for continuing our work is to study view-based query containment under exact views.

10. Acknowledgments

This work was supported in part by projects Infomix (IST-2001-33570) and Sewasie (IST-2001-34825) funded by the European Union, by project D2I ("From Data to Information") funded by MIUR (Italian Ministry of University and Research), by NSF grants CCR-9988322, CCR-0124077, IIS-9908435, IIS-9978135, and EIA-0086264, by BSF grant 9800096, and by a grant from the Intel Corporation.

11. REFERENCES

[1] S. Abiteboul. Querying semi-structured data. In *Proc. of ICDT'97*, pages 1–18, 1997.

[2] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: from Relations to Semistructured Data and XML.* Morgan Kaufmann, Los Altos, 2000.

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

[4] S. Abiteboul and V. Vianu. Regular path queries with constraints. *J. of Computer and System Sciences*, 58(3):428–452, 1999.

[5] S. Adali, K. S. Candan, Y. Papakonstantinou, and V. S. Subrahmanian. Query caching and optimization in distributed mediator systems. In *Proc. of ACM SIGMOD*, pages 137–148, 1996.

[6] M. Ajtai and R. Fagin. Reachability is harder for directed than for undirected finite graphs. *J. of Symbolic Logic*, 55(1):113–150, 1990.

[7] T. Bray, J. Paoli, and C. M. Sperberg-McQueen. Extensible Markup Language (XML) 1.0 — W3C recommendation. Technical report, World Wide Web Consortium, 1998. Available at `http://www.w3.org/TR/1998/REC-xml-19980210`.

[8] P. Buneman. Semistructured data. In *Proc. of PODS'97*, pages 117–121, 1997.

[9] P. Buneman, S. Davidson, G. Hillebrand, and D. Suciu. A query language and optimization technique for unstructured data. In *Proc. of ACM SIGMOD*, pages 505–516, 1996.

[10] D. Calvanese, G. De Giacomo, and M. Lenzerini. Representing and reasoning on XML documents: A description logic approach. *J. of Log. and Comp.*, 9(3):295–318, 1999.

[11] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Description logic framework for information integration. In *Proc. of KR'98*, pages 2–13, 1998.

[12] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi. Query processing using views for regular path queries with inverse. In *Proc. of PODS 2000*, pages 58–66, 2000.

[13] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi. View-based query processing and constraint satisfaction. In *Proc. of LICS 2000*, pages 361–371, 2000.

[14] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi. Lossless regular views. In *Proc. of PODS 2002*, pages 58–66, 2002.

[15] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi. View-based query answering and query containment over semistructured data. In G. Ghelli and G. Grahne, editors, *Revised Papers of the 8th International Workshop on Database Programming Languages (DBPL 2001)*, volume 2397 of *LNCS*, pages 40–61. Springer, 2002.

[16] S. Chaudhuri, S. Krishnamurthy, S. Potarnianos, and K. Shim. Optimizing queries with materialized views. In *Proc. of ICDE'95*, Taipei (Taiwan), 1995.

[17] T. Feder and M. Y. Vardi. The computational structure of monotone monadic SNP and constraint satisfaction. *SIAM J. on Computing*, 28:57–104, 1999.

[18] M. F. Fernandez, D. Florescu, A. Levy, and D. Suciu. Verifying integrity constraints on web-sites. In *Proc. of IJCAI'99*, pages 614–619, 1999.

[19] D. Florescu, A. Levy, and A. Mendelzon. Database techniques for the World-Wide Web: A survey. *SIGMOD Record*, 27(3):59–74, 1998.

[20] M. Friedman, A. Levy, and T. Millstein. Navigational plans for data integration. In *Proc. of AAAI'99*, pages 67–73. AAAI Press/The MIT Press, 1999.

[21] G. Grahne and A. O. Mendelzon. Tableau techniques for querying information sources through global schemas. In *Proc. of ICDT'99*, volume 1540 of *LNCS*, pages 332–347. Springer, 1999.

[22] A. Gupta and J. D. Ullman. Generalizing conjunctive query containment for view maintenance and integrity constraint verification (abstract). In *Workshop on Deductive Databases (In conjunction with JICSLP)*, page 195, Washington D.C. (USA), 1992.

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

[24] P. G. Kolaitis and M. Y. Vardi. Conjunctive-query containment and constraint satisfaction. In *Proc. of PODS'98*, pages 205–213, 1998.

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

[26] A. Y. Levy, A. O. Mendelzon, Y. Sagiv, and D. Srivastava. Answering queries using views. In *Proc. of PODS'95*, pages 95–104, 1995.

[27] A. Y. Levy and M.-C. Rousset. Verification of knowledge bases: a unifying logical view. In *Proc. of the 4th European Symposium on the Validation and Verification of Knowledge Based Systems*, Leuven, Belgium, 1997.

[28] A. Y. Levy and Y. Sagiv. Semantic query optimization in Datalog programs. In *Proc. of PODS'95*, pages 163–173, 1995.

[29] H. R. Lewis. Complexity of solvable cases of the decision problem for the predicate calculus. In *FOCS-78*, pages 35–47, 1978.

[30] C. Li and E. Chang. On answering queries in the presence of limited access patterns. In *Proc. of ICDT 2001*, pages 219–233, 2001.

[31] T. D. Millstein, A. Y. Levy, and M. Friedman. Query containment for data integration systems. In *Proc. of PODS 2000*, pages 67–75, 2000.

[32] T. Milo and D. Suciu. Index structures for path expressions. In *Proc. of ICDT'99*, volume 1540 of *LNCS*, pages 277–295. Springer, 1999.

[33] A. Motro. Panorama: A database system that annotates its answers to queries with their properties. *J. of Intelligent Information Systems*, 7(1), 1996.

[34] P. van Emde Boas. The convenience of tilings. In A. Sorbi, editor, *Complexity, Logic, and Recursion Theory*, volume 187 of *Lecture Notes in Pure and Applied Mathematics*, pages 331–363. Marcel Dekker Inc., 1997.

[35] M. Y. Vardi. A note on the reduction of two-way automata to one-way automata. *Information Processing Letters*, 30(5):261–264, 1989.

[36] H. Wang. Dominoes and the $\forall\exists\forall$ case of the decision problem. In *Symposium on the Mathematical Theory of Automata*, pages 23–55, 1962.