Querying the Deep Web: Back to the Foundations

Andrea Calí[1,4], Davide Martinenghi[2], Igor Razgon[1], and Martín Ugarte[3]

[1]Dept of Comp. Sci. and Inf. Syst.
Birkbeck, Univ. of London, UK

[2]Dip. di Elettr., Informaz. e Bioing.
Politecnico di Milano, Italy

[3]Web and Info. Technologies Lab.
Université Libre de Bruxelles

[4]Oxford-Man Inst. of Quantitative Finance
University of Oxford, UK

{andrea,igor}@dcs.bbk.ac.uk
davide.martinenghi@polimi.it
mugartec@ulb.ac.be

Abstract. The Deep Web is the large corpus of data accessible on the Web through forms and presented in dynamically-generated pages, but not indexable as static pages, and therefore invisible to search engines. Deep Web data are usually modelled as relations with so-called access limitations, that is, they can be queried only by selecting certain attributes. In this paper we give some fundamental complexity results on the problem of processing conjunctive (select-project-join) queries on relational data with access limitations.

1 Introduction

The term *Deep Web* (sometimes also called *Hidden Web*) [7, 5, 6] refers to the data content that is created dynamically as the result of a specific search on the web. For example, when we query a White Pages website, the generated output consists of one or more pages containing the result of a query posed on an underlying database; these pages cannot be indexed by search engines. When we query whitepages.com through a form, we are forced to fill in some of the fields of the form, for instance the Name field; the result is then structured as a table. A Deep Web source can be naturally modelled as a relational table (or a set of relational tables) that can be queried only according to so-called *access patterns*, each of which enforces the selection on some of the attributes (which corresponds to filling the input fields in the form with values), which are called *input* attributes. Relational tables accessible through access patterns are said to have *access limitations*.

Processing structured queries over Deep Web sources is the key problem in the integration of such sources. Interestingly, when Deep Web sources are modeled, as mentioned, as relations with access limitations, answering a simple conjunctive (select-project-join) query on such sources requires, in the worst case, the evaluation of a *recursive* Datalog query plan. In such plans, values obtained as output from a source are used as input for other sources; the compatibility

$$\rho_1 : \quad q() \leftarrow \hat{r}(X,Y), \hat{s}(Z,Y) \qquad \rho_5 : dom(Y) \leftarrow \hat{s}(X,Y)$$
$$\rho_2 : \hat{r}(X,Y) \leftarrow dom(X), r(X,Y) \qquad \rho_5 : dom(Y) \leftarrow \hat{r}(X,Y)$$
$$\rho_3 : \hat{s}(X,Y) \leftarrow dom(X), s(X,Y) \qquad \rho_6 : \ dom(a)$$

Fig. 1. Datalog program for Example 1

of values is established by assigning to each attribute of a relation a so-called *abstract domain*, which expresses the type of value (e.g. name, address etc.) as opposed to the concrete domain (e.g. string, integer etc.).

In this paper we consider conjunctive queries (CQs) on relational schemata with access limitations and the two traditional problems associated with them: *query answering* and *query containment*. Such problems have been extensively studied in the literature; however, for some cases the problem of determining the complexity is still open. We tackle some of such cases with the following results:

- We show that CQ answering under access limitations is NP-complete with respect to combined complexity; thus, the access limitations do not increase the complexity of query answering in the classic case (without access limitations).
- We consider the problem of CQ containment under access limitations, known to be co-NEXPTIME-complete in its general form [3, 4] and thought to be EXPTIME-complete in the case of queries without constants [2]. We first address the case of *input-only* predicates; we show that in such a case the problem is Π_2^{p}-complete. As for the hardness, we show that Π_2^{p}-hardness holds under stricter conditions: for predicates of arity $\leqslant 2$ and two abstract domains. Then we address CQ containment for (input-output) binary predicates; we conjecture that this problem is also in Π_2^{p}.

2 Query Answering

We assume the reader is familiar with the notions of relational schema and instance, conjunctive query and Datalog program — otherwise, see for instance the book of Abiteboul et al. [1]. We consider relational schemata whose predicates are annotated so as to express whether each argument/attribute is *input* (needs to be selected) or *output*; for instance, r^{iio}, of arity 3, has the first two attributes as input attributes, and the third as output.

In the presence of access limitations on the sources, queries cannot be usually evaluated as in the traditional case, as we show below. Given a conjunctive query q, a schema with access limitations (implicit), an instance D and a set I of initial constants, the answers to q, denoted $\mathrm{ans}(q, I, D)$, are obtained starting from the constants in I and extracting all possible tuples (by using the constants as input in all possible ways); with the newly obtained constants again all possible tuples are extracted, and so on, until no new tuple is extracted – see e.g. [5].

Example 1. Consider a schema with predicates r^{io} and s^{io} (which contain the facts of the database D), a set of initial constants $I = \{a\}$, and the Boolean CQ

$q() \leftarrow r(X,Y), s(Z,Y)$. Assume there is a single abstract domain, represented by the unary predicate *dom*, associated with all attributes (arguments). The Datalog program Π_q for q is shown in Figure 1 (facts of D omitted). The query is rewritten over the *cache* relations $\hat{r}, \hat{s}$ (rule ρ_1) defined in the cache rules ρ_2 and ρ_3, which contain the facts extracted according to rules ρ_2 and ρ_3. ∎

We now come to our result on the decision problem of CQ answering under access limitations; w.l.o.g., we consider Boolean CQs.

Theorem 1. *CQ answering under access limitations is* NP*-complete with respect to combined complexity.*

Proof (sketch). For membership we exhibit a non-deterministic algorithm that performs $\leqslant |D|$ steps, and at each step guesses one of the $\leqslant |D|^{W \cdot |\mathcal{R}|}$ possible accesses to relations. Hardness follows from CQ answering without access limitations.

3 Query Containment

Definition 1. *Consider two CQs q_1, q_2 over a schema with access limitations, as well as a set I of initial constants such that* $\text{const}(q_1) \cup \text{const}(q_2) \subseteq I \subseteq \Delta$ *(*$\text{const}(q)$ *denotes the constants in a query q, while Δ denotes the infinite domain of constants); we say that q_1 is* contained *in q_2 under access limitations with respect to I, denoted $q_1 \subseteq_I q_2$, if, for every database D for $\mathcal{R}$, we have* $\text{ans}(q_1, I, D) \subseteq \text{ans}(q_2, I, D)$.

Checking containment amounts to checking containment between two recursive Datalog programs in the special form presented in Section 2. W.l.o.g., we consider Boolean CQs as in Section 2. We first consider the case of *input-only* predicates. An input-only n-predicate r is accessed, in an instance D, with an n-tuple $\langle t \rangle$ of constants of the appropriate domain, and tells (with a Boolean result) whether $r(\langle t \rangle) \in D$. Evidently, this restricts the definition of containment to instances composed solely of constants of the initial set I. The following lemma has a rather straightforward proof. A tight hardness result follows.

Lemma 1. *CQ containment under access limitations with input-only predicates is in* Π_2^{p}.

Theorem 2. *CQ containment under access limitations with input-only predicates of arity* $\leqslant 2$ *and two abstract domains is* Π_2^{p}*-hard.*

Proof (sketch). The proof is by reduction from a tighter version of Generalised-Graph-Colouring [8], which is Σ_2^{p}-complete and is defined as follows: given a graph F and a positive integer k, is there a two-colouring of the vertices of F that does not contain a monochromatic (on vertices) clique of k vertices? We reduce Generalised-Graph-Colouring to non-containment under the above stated restrictions using a predicate $e/2$ for edges and a predicate $col/2$ to indicate by $col(v, c)$ that the vertex v has colour c.

As a corollary we get tight bounds for the input-only case.

Corollary 1. *CQ containment under access limitations with input-only predicates is* Π_2^{p}*-complete.*

Finally, we present our result on the general binary case, not restricted to input-only predicates.

Theorem 3. *CQ containment under access limitations with binary predicates is in* Π_2^{p}.

Proof (sketch). The proof uses the *crayfish-chase* technique of [4] to check $q_1 \subseteq_I q_2$ in the binary case. Relying on the fact that q_2 is "blind" to pairs of atoms that are more than $|q_2|$ steps apart in a join graph, to check the existence of a counterexample for containment we guess, by means of the crayfish-chase, a polynomially bound set of atoms representing a fragment of instance that makes q_1 true; then we check whether no homomorphism maps q_2 onto such fragment.

4 Discussion

We have presented some results on our ongoing study of the fundamentals of the complexity of CQ answering and containment under access limitations. Interestingly, some of the fundamental problems have been overlooked in the literature, for instance the complexity of CQ answering under access limitations, for which we gave a tight bound. We also presented results for the input-only case, employing techniques that, we believe, pave the way to future investigations. The binary case is interesting as most knowledge representation formalisms rely on binary relations; we plan to find a tight bound for its complexity, proving our conjecture. Finally, we shall study CQ answering and containment under access limitations as well as integrity constraints expressed as ontological rules; this has applications in the intersection between the Semantic Web and the Deep Web.

References

1. Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases.* Addison-Wesley, 1995.
2. Michael Benedikt. Personal communication, 2017.
3. Michael Benedikt, Georg Gottlob, and Pierre Senellart. Determining relevance of accesses at runtime. In *Proc. of PODS*, pages 211–222, 2011.
4. Andrea Calì and Davide Martinenghi. Conjunctive Query Containment under Access Limitations. In *Proc. of ER*, pages 326–340, 2008.
5. Andrea Calì and Davide Martinenghi. Querying data under access limitations. In *Proc. of ICDE*, pages 50–59, 2008.
6. Kevin Chen-Chuan Chang, Bin He, and Zhen Zhang. Toward large scale integration: Building a metaquerier over databases on the web. In *Proc. of CIDR*, pages 44–55, 2005.
7. Jayant Madhavan, Loredana Afanasiev, Lyublena Antova, and Alon Y. Halevy. Harnessing the deep web: Present and future. In *Proc. of CIDR*, 2009.
8. Vladislav Rutenburg. Complexity of generalized graph coloring. In *Proc. of MFCS*, pages 573–581, 1986.

A A Tight Bound

As a tighter bound, we show the following result.

Theorem 4. *CQ containment under access limitations with input-only binary predicates and one abstract domain is Π_2^p-hard.*

Proof. The proof is by reduction from the following problem, which is Σ_2^p-complete.

> Generalised-Graph-Colouring, tight version. Given a graph F and an integer $k \geqslant 2$, determine whether there is a two-colouring of the vertices of F such that F does not contain a complete graph of order k that is monochromatic.

We reduce Generalised-Graph-Colouring to non-containment under access limitations, with the stated restrictions. Without loss of generality, we assume that F has no loops — loops are irrelevant to the problem but they would complicate the reduction (easy proof, left to the reader). We use a predicate $e/2$ to represent graph edges. We construct the problem of determining $q_1 \not\subseteq_I q_2$. As set of initial constants, we choose $V = \{v_1, \ldots, v_n\}$ corresponding to the n nodes of F. We introduce a predicate $col/2$ to characterise the colour of a certain node; $col(v, c)$ means that the node v has colour c. Now, rather than using a separate abstract domain for red/green colouring, we *partition* V arbitrarily into two (non-empty) sets: $V = G \cup R$, $G \cap R = \varnothing$. The two sets serve to represent *green* and *red* with the two partitions instead of using two values; we have that $col(v, c)$ represents that v is coloured in green (resp. red) iff $c \in G$ (resp. $c \in R$). To make our query comply to this representation, we need to represent the equivalence classes G and R by means of a binary predicate $samecol/2$: for each pair of distinct values (nodes) v_1, v_2 that belong to the same class (G or R) in V, we have the atoms $samecol(v_i, v_j), samecol(v_j, v_i)$. We thus construct q_1 as follows.

(i) For each arc (v_i, v_j) in F, we have the atoms $e(v_i, v_j), e(v_j, v_i)$ (the graph is undirected, therefore symmetric).
(ii) For each $v_i \in V$, we have the atom $col(v_i, W_i)$; we use all distinct variables $W_1, \ldots, W_n$.
(iii) For each v_i, v_j such that $v_i \in G$, $v_j \in G$ and $v_i \neq v_j$, we have the atoms $samecol(v_i, v_j), samecol(v_j, v_i)$.
(iv) For each v_i, v_j such that $v_i \in R$, $v_j \in R$ and $v_i \neq v_j$, we have the atoms $samecol(v_i, v_j), samecol(v_j, v_i)$.

Notice that the predicate $samecol/2$ represents an equivalence relation between nodes of V, where two nodes are equivalent iff they have the same colour (and there are only two colours: red and green).

The right-hand side query q_2 is constructed as follows, using variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ and $\mathbf{Z} = \{Z_1, \ldots, Z_n\}$.

(i) For each pair X_i, X_j in $\mathbf{X}$ such that $X_i \neq X_j$, we have the atoms $e(X_i, X_j), e(X_j, X_i)$. This represents a complete graph of order k.
(ii) For each X_i with $1 \leqslant i \leqslant k$, we have the atom $col(X_i, Z_i)$.
(iii) For each pair Z_i, Z_{i+1} with $1 \leqslant i < k$, we add the atom $samecol(Z_i, Z_{i+1})$. This expresses the fact that the complete graph represented by q_2 is monochromatic.

Notice that the choice of G and R as a partition of V does *not* to represent which v_i are red or green; this is left to the choice of the variables W_i in q_1. The instantiation of $samecol/2$ in q_1 is merely a usage of V to represent a two-colour set, without resorting to an additional (binary) abstract domain.

Note (Andrea): The following is admittedly a sketch (the "If" part is more or less a tautology, but what do you want). Please help.

It remains to show that, if we call $\mathcal{I}_1$ the instance of Generalised-Graph-Colouring and $\mathcal{I}_2$ the non-containment instance that we have constructed, I_1 admits a solution iff I_2 does. This is now straightforwardly seen.

Only If. Assume I_1 has a positive answer; this means that there is a two-colouring with the desired property. The existence of such colouring allows for a mapping of the W_i to constants of V (no other constants are possible as all predicates are input-only); such a mapping witnesses the existence of an instance for q_1; q_2 cannot map via homomorphism onto such an instance because the colouring in question does not make F contain a complete graph of order k.

If. Assume I_2 has a positive answer. This means, with a construction as above, that there is an instance that makes q_1 true and q_2 false; such an instance witnesses the existence of a two-colouring of F such that F does not contain a complete graph of order k.

B Conjunctive Regular Path Queries

Our results extend rather naturally to the containment of conjunctive regular path queries (CRPQs) into conjunctive queries. Similarly to how we prove Π_2^{p} membership of query containment in the binary case, we can prove the following theorem.

Theorem 5. *Checking containment of a CRPQ in a CQ is in* Π_2^{p}.

Note (Andrea): This was proved with Martín on the 20th of April and the detailed proof is being written.

Proof. We show that the complement of the problem is in Σ_2^{p}. Given a CRPQ γ and a CQ Q, to show that $\gamma \not\sqsubseteq Q$ we guess a model of γ and then show that there is no homomorphism from Q to that model. The guess part can be done in NP provided that the counterexample is polynomial; then checking that there is no homomorphism from Q to the counterexample is simply done by accesing an NP oracle (recall that $\Sigma_2^{\mathrm{p}} = \mathrm{NP}^{\mathrm{NP}}$). Thus, we need to show that there always is

a polynomial counterexample. Formally, this is the existence of $k, n_0 \in \mathbb{N}$ such that for every CRPQ γ and CQ Q such that $\|\gamma\| + \|Q\| \geqslant n_0$, if $\gamma \not\subseteq Q$ then there is a model D of size at most $(\|\gamma\| + \|Q\|)^k$ such that $D \models \gamma$ but $D \not\models Q$.

We proceed by contradiction and assume that for each n_0 such k does not exist. Fix $n_0 = 4$. There must be a CRPQ γ and a CQ Q such that $\|\gamma\| + \|Q\| \geqslant 4$ and $\gamma \not\subseteq Q$, but the smallest model satisfying both $D \models \gamma$ and $D \not\models Q$ is bigger than ℓ^4.

Let $\gamma = \exists \bar{y} \bigwedge_{i=1}^{n} L_i(u_i, v_i)$ be a CRPQ and let $Q = \exists \bar{z} \bigwedge_{i=1}^{m} R_i(x_i, y_i)$ be a CQ. We assume w.l.o.g. that Q is connected and that each L_i is represented by an NFA. To show that checking $\gamma \subseteq Q$ can be done in Π_2^{p} we show that checking $\gamma \not\subseteq Q$ can be done in Σ_2^{p}.

We are also able to establish a tight lower bound for the same problem.

Theorem 6. *Checking containment of a CRPQ in a CQ, in the case of two binary predicates only, is Π_2^{p}-hard.*

Note (Andrea): The following is evidently just a sketch. However, it seems the details are all sorted out. Please check!

Proof. The proof is by reduction from the Generalised-Ramsay-Number problem. The problem is stated as follows.

> Generalised-Ramsay-Number. Given an undirected graph F, a partial two-colouring of the edges of F and an integer $k \geqslant 2$, determine whether every complete two-colouring of F contains a monochromatic clique of order k.

Given an instance I of Generalised-Ramsay-Number, we now construct two instance I_r and I_g of CRPQ-CQ containment; we will show that I has a positive answer iff I_r or I_g have positive answer. The instance I consists of $\langle F, \lambda, k \rangle$, where $\lambda : F \to \{r, g\}$ is a partial colouring function (r and g stand for red and green respectively). We construct I_r as $q_1 \subseteq q_{2r}$ where q_{1r} and q_{2r} are as below. Since we deal with an undirected graph and we use binary predicates only, we assume that whenever we have an atom $r(v, v')$ we also have the symmetric $r(v', v)$, which for brevity we do not write. We use the constants $v_1, \ldots, v_n$, exactly one for each node of F. We use the binary predicates $r(v, v')$ (resp. $g(v, v')$) to denote that the arc (v, v') is coloured in red (resp. green). Now, for each edge (v, v') in F, we have in q_1:

- the atom $r(v, v')$ if $\lambda(v, v') = r$;
- the atom $g(v, v')$ if $\lambda(v, v') = g$;
- the atom $(r + g)(v, v')$ if $\lambda(v, v')$ is not defined.

The CQ q_{2r} has k variables $X_1, \ldots, X_k$ and for each X_i, X_j, with $1 \leqslant i < j \leqslant k$, q_{2r} has atoms $r(X_i, X_j), r(X_j, X_i)$. The CQ q_{2g} is analogous, with atoms $r(X_i, X_j), r(X_j, X_i)$. Intuitively, q_{2r} (resp. q_{2g}) encodes a k-clique of all red (resp. green) arcs. The instance I_g is constructed analogously, with a green (instead of red) monochromatic k-clique represented by q_{2g}. We now show that I

has a positive answer iff I_r or I_g have a positive answer. Only if. If I has a positive answer, then every colouring that completes λ contains a k-clique that is all red or all green. Clearly every colouring λ' of F that completes λ corresponds to a choice of *either* r or g for the atoms $(r+g)(v_i, v_j)$; clearly, if for all λ' a monochromatic clique exists, therefore either q_{2r} or q_{2g} evaluates to true on all images of q_1 with the choice above. Obviously if for some arc (v_i, v_j) we choose both atoms $r(v_i, v_j), s(v_i, v_j)$, this does not prevent the existence of cliques. If. By contradiction, let us assume that I has negative answer while I_r or I_g have a positive answer. Hence, there is a counterexample, that is, a colouring λ' that completes λ such that there F has no monochromatic k-clique. Evidently, such counterexample corresponds to an instance B satisfying q_1 but neither q_{2r} nor q_{2g}; the instance B satisfies each atom $(r+g)(v_i, v_j)$ of q_1 with one atom, $r(v_i, v_j)$ or $g(v_i, v_j)$; this is a contradiction.

C The case of fixed domain size

Note (Andrea): After a chat with Igor, we identified the case of CQ containment when the domain Δ has size bounded by a constants, i.e. $|\Delta| \leqslant M$ with M constant natural number. This case implies of course that Δ is in the input. Notice that in this case each atom can assume values (arguments) *only* in Δ, which makes this case similar to the one where predicates are input-only. We also have an initial set of constants I such that $I \subseteq \Delta$ — having some $c \in I$ with $c \notin \Delta$ makes no sense as c would be completely useless. In the case of binary predicates, Igor proved that the problem is coNP-hard with $|\Delta| = 4$ and fixed right-hand side q_2. It remains to establish the complexity of the problem when q_1, q_2 are not of bounded size. My take on the case of binary, input-only predicates follows.

Theorem 7. *CQ containment in the case of binary, input-only predicates and bounded domain is complete for the class* $\text{coNP} \cup \text{NP}$.

Note (Martín): There seems to be a problem with this statement; $\text{coNP} \cup \text{NP}$ is not a class for which a problem can be complete (unless some classical assumptions break). The contradiction is the following: If a language L is complete for the class $\text{coNP} \cup \text{NP}$, then L belongs to $\text{coNP} \cup \text{NP}$. Assume w.l.o.g. that $L \in \text{coNP}$. As L is hard for $\text{coNP} \cup \text{NP}$, any language in this class should be reducible (by a reasonable notion of reduction) to L. In particular SAT has to be reducible to L, which is a reduction of an NP-complete problem into a problem in coNP.
Judging by the structure of the problem I would say it is complete for DP.

Note (Andrea): The following is admittedly a sketch, but fairly complete.

Proof. Let the number of values in Δ be bounded by M: $|\Delta| \leqslant M$; let us denote $k = |\Delta|$. We distinguish two cases: *(a)* $k < |\text{var}(q_1)|$ and *(b)* $k \geqslant |\text{var}(q_1)|$.

Case (a). In such a case, we are looking for an instance that makes q_1 true and q_2 false, as a counterexample for containment. We claim that the problem is in this case coNP-complete. *Membership.* For obvious reasons (if we try to falsify q_2 we try to have an instance as small as possible) we consider images of q_1 according to a homomorphism from $\mathrm{var}(q_1)$ to Δ, that is, we ground q_1 with values of Δ. Clearly there are $k^{|\mathrm{var}(q_1)|}$ possible ways of grounding q_1 as above; however, due to symmetries, we need not to naïvely consider such an exponential number of groundings; in fact, we only need to consider all ways of having $k-1$ variables that take distinct values, while all others take the same value (the only one left in Δ, distinct from the previous $k-1$). Therefore we consider $\binom{|\mathrm{var}(q_1)|}{k-1}$ possible instances, which is a number bounded by $|\mathrm{var}(q_1)|^{k+1}$ and therefore bounded by a constant (yay!). At this point, for each of such instances, say B, we check whether $B \models q_2$, that is, whether q_2 evaluates to true on B.

Note (Andrea): Did we assume anywhere that the queries are Boolean? It is not strictly necessary, but it spares us a bit of hassle.

The procedure is straightforwardly (at least we hope!) seen to be correct and its complexity is coNP. *Hardness.*

Note (Andrea): This is Igor's proof in the other draft; it is correct and, once copied here in civilised form, it will perfectly work.

Case (b). This case is easy as, since we are try to find a counterexample to containment, we look for groundings of q_1 with as many distinct values as possible; since we have enough values to make distinct variables map onto distinct constants, the problem amounts to the CQ containment without access limitations, which is NP-complete.

We get an interesting corollary for a more general problem which, to the best of our knowledge (and surprisingly), has never been studied in the literature.

Corollary 2. *CQ containment for CQs with binary predicates, with domain bounded by a constant, is complete for the class* $\mathrm{coNP} \cup \mathrm{NP}$.

Note (Andrea): I am not sure how to extend to the case of input-output predicates. Clues might come from the proof of the upper complexity bound of CQ containment with input-output predicates in the case of unbounded domain, which Martín and I should produce shortly. The proof of the upper bound is based on certain expansions of the canonical image of q_1 (which constitute the infamous *crayfish chase*), of which only those of up to a polynomial size are to be checked. Such expansion will have values, in the new case of bounded domain, in Δ only.