Investigating Monotone Abstractions

Gianluca Cima[2], Marco Console[1], Maurizio Lenzerini[1] and Antonella Poggi[1]

[1]Sapienza Università di Roma
[2]CNRS & University of Bordeaux

Abstract

In Ontology-based Data Management (OBDM), an abstraction of a source query q is a query over the ontology capturing the semantics of q in terms of the concepts and the relations available in the ontology. Since a perfect characterisation of a source query may not exist, the notions of best sound and complete approximations of an abstraction have been introduced and studied in the typical OBDM context, i.e., in the case where the ontology is expressed in DL-Lite, and source queries are expressed as unions of conjunctive queries (UCQs). Interestingly, if we restrict our attention to abstractions expressed as UCQs, even best approximations of abstractions are not guaranteed to exist. Thus, a natural question to ask is whether such limitations affect even larger classes of queries. In this paper, we answer this fundamental question for an essential class of queries, namely the class of monotone queries. We define a monotone query language based on disjunctive Datalog enriched with an epistemic operator, and show that its expressive power suffices for expressing the best approximations of monotone abstractions of UCQs.

Keywords

Abstraction, Disjunctive Datalog, Monotone queries, Epistemic queries

1. Introduction

In *Ontology-based Data Management (OBDM)* [1], an ontology, i.e., a formal, logic-based representation of a domain of interest, is used to provide a high-level conceptual tool for accessing and managing the data sources of an information system. Suitable mappings declaratively specify the relationship between the data at the sources and the elements in the ontology, and this enables the user to carry out many relevant tasks on data through the lens of the ontology [2, 3].

Recent papers [4, 5, 6, 7, 8] address a novel issue in OBDM: starting from a query q_S expressed over the sources, the goal is to find a so-called *abstraction* of q_S [9], i.e., an ontology-based characterization of q_S expressed in terms of the ontology elements, and whose answers coincide with the answer to the original query, modulo the ontology and the mapping. We encountered the need of abstraction during a joint project with a public statistical research institute. The institute's departments must publish subsets of the data they gather in the form of semantically described linked open data. To compute the content of the datasets the departments execute suitable queries over the data sources mapped to a shared ontology. Notably, when the dataset is published, it must be documented through a SPARQL query expressed in terms of the ontology.

SEBD 2022: The 30th Italian Symposium on Advanced Database Systems, June 19-22, 2022, Tirrenia (PI), Italy
gianluca.cima@u-bordeaux.fr (G. Cima); console@diag.uniroma1.it (M. Console); lenzerini@diag.uniroma1.it (M. Lenzerini); poggi@diag.uniroma1.it (A. Poggi)

This task is currently done manually. The notion of abstraction perfectly captures this scenario and provides the formal tool for automating the process: given the query over the sources computing the content of the dataset, the abstraction of such query with respect to the mapping and the ontology is exactly the SPARQL query to be associated to the open dataset. Besides the above use case, abstraction can be the appropriate tool in various scenarios. For additional insights we refer to the references mentioned above.

The first investigations on abstraction appear in [4, 5]. Both papers point out that the "perfect" abstraction of a union of conjunctive queries (UCQ) expressed over the data source not always exists, and present algorithms for computing such abstraction in the case where it both exists, and can be expressed as a UCQ over the ontology. In [4, 6] the notion of (sound and complete) approximations of the perfect abstraction is introduced, exactly to cope with situations in which perfectness cannot be achieved. Moreover, both papers make it clear that, for a given class of queries C, one is probably interested in two specific forms of approximations, called *C-minimally complete* and *C-maximally sound* abstractions. Based on these notions, [6] presents a thorough analysis of the verification problem (check whether a given query is a complete or sound abstraction) and the computation problem of UCQ-minimally complete and UCQ-maximally sound abstractions of UCQ source queries in OBDM systems based on *DL-Lite*. In [7] the computation problem is studied in the context of a specific class of non-monotone queries for expressing abstractions, and it is shown that this class can provide abstractions that are better than the one in the UCQ class.

Thus, with the exception of [7], all the results on abstractions have been obtained under the assumptions that abstractions are expressed as UCQs over the ontology, and many of them originate from the observations that best approximations in the UCQ class are not guaranteed to exist. Thus, a natural issue to investigate is whether such limitations affect even larger classes of queries for expressing abstractions. The main goal of this paper is to address the following question: do approximations of perfect abstraction that are best in a fundamental class of queries, namely the class of monotone queries, always exist? Obviously, a related goal is to derive algorithms for computing approximations of abstractions that are best in the class of monotone queries, if they exist. Note that the class of monotone queries includes queries expressible in First-Order Logic and is therefore extremely important.

In this paper we answer positively to the above-mentioned fundamental question. More specifically, the contributions of the paper can be summarized as follows. We present a general framework for abstraction in OBDM, based on the definition of queries as functions from the logical models of OBDM systems to sets of tuples. The framework includes a new monotone query language, called $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$, based on disjunctive Datalog enriched with inequalities and an epistemic operator (Section 2). We consider a scenario where the OBDM specification J is based on *DL-Lite*$_{RDFS}$ (i.e., the fragment of RDFS expressible in Description Logic), and show that, in the considered scenario, for any source UCQ $q_{\mathcal{S}}$, the best (sound or complete) approximations of the J-abstraction of $q_{\mathcal{S}}$ in the class of monotone queries always exists and can be expressed in $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ (Section 3). As a consequence, if the perfect abstraction exists and is in the class of monotone queries, then it can be expressed in $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ (Section 4).

This paper is an extended abstract of [10]. Hence, while we assume basic knowledge about databases [11] and Description Logics (DL) [12], for specific concepts and notations, we refer to the Preliminaries section of [10].

2. Framework

In what follows, we implicitly refer to an OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, and when we denote a query by $q_{\mathcal{O}}$ (resp., $q_{\mathcal{S}}$) we mean that the query is a query for J (resp., a source query), i.e., is over the signature of the ontology $\mathcal{O}$ (resp., the schema $\mathcal{S}$). We follow [6] for the basic definitions related to abstraction.

We say that $q_{\mathcal{O}}$ is a *perfect J-abstraction* of $q_{\mathcal{S}}$ if $q_{\mathcal{O}}^{J,D} = q_{\mathcal{S}}^{D}$, for each $\mathcal{S}$-database D consistent with J.

As the condition for an ontology query to be a perfect abstraction of a source query is a strong one, it might be very well the case that a perfect abstraction of a source query does not exist. It is then reasonable to consider weaker notions, such as sound or complete approximations, of perfectness.

We say that $q_{\mathcal{O}}$ is a *complete (resp. sound) J-abstraction* of $q_{\mathcal{S}}$ if $q_{\mathcal{S}}^{D} \subseteq q_{\mathcal{O}}^{J,D}$ (resp. $q_{\mathcal{O}}^{J,D} \subseteq q_{\mathcal{S}}^{D}$), for each $\mathcal{S}$-database D consistent with J.

Obviously, we might be interested in complete or sound abstractions that approximate $q_{\mathcal{S}}$ at best, at least in the context of a specific class of queries. If $\mathcal{L}_{\mathcal{O}}$ is a class of queries, we say that a query $q_{\mathcal{O}} \in \mathcal{L}_{\mathcal{O}}$ is an *$\mathcal{L}_{\mathcal{O}}$-minimally complete* (resp., *$\mathcal{L}_{\mathcal{O}}$-maximally sound*) J-abstraction of $q_{\mathcal{S}}$ if $q_{\mathcal{O}}$ is a complete (resp., sound) J-abstraction of $q_{\mathcal{S}}$ and there is no query $q'_{\mathcal{O}} \in \mathcal{L}_{\mathcal{O}}$ such that $q'_{\mathcal{O}}$ is a complete (resp., sound) J-abstraction of $q_{\mathcal{S}}$ and $q'_{\mathcal{O}} \sqsubset_J q_{\mathcal{O}}$ (resp., $q_{\mathcal{O}} \sqsubset_J q'_{\mathcal{O}}$).

We now introduce a new language, called $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$, for expressing queries over OBDM specifications. The language is based on disjunctive Datalog, and is used in this paper for expressing abstractions. The basic component of a $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ query is a rule. Assume two disjoint and countably infinite sets of predicates $\mathcal{E}$ and $\mathcal{P}$, called extensional and intensional, respectively. A $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ rule has one of the following forms:

- The typical form of disjunctive Datalog, i.e.,

$$\gamma(\bar{x}) \rightarrow \alpha_1(\bar{x}_1) \vee \ldots \vee \alpha_n(\bar{x}_n) \tag{1}$$

  where $\gamma(\bar{x})$ is a conjunction of relational atoms on the predicates of $\mathcal{P}$ with $\vec{x}$ as variables, and for $i = 1, \ldots, n$, $\alpha_i(\bar{x}_i)$ is a single relational atom whose predicate is in $\mathcal{P}$ such that $\bar{x}_i \subseteq \bar{x}$,
- A new form specified as follows

$$\mathbf{K}(\exists \bar{z}.\phi(\bar{x}, \bar{z}) \wedge \xi(\bar{x})) \rightarrow \psi_1(\bar{x}_1) \vee \ldots \vee \psi_n(\bar{x}_n) \tag{2}$$

  where ϕ is a conjunction of relational atoms over $\mathcal{E}$, $\xi(\bar{x})$ is a conjunction of inequality atoms involving only variables from $\bar{x}$ and for $j = 1, \ldots, n$, $\psi_j = \exists \bar{y}_j.\gamma_j(\bar{x}_j, \bar{y}_j)$, where γ_j is a conjunction of relational atoms on $\mathcal{P}$. When $\bar{x}_j$ contains only variables $\bar{x}$ occurring in $\phi(\bar{x}, \bar{z})$, for $j = 1, \ldots, n$, we say that the rule is *safe*.

An n-ary $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ query $q_{\mathcal{O}}$ over an OBDM specification J is a finite set of $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ rules whose extensional predicates coincide with the alphabet of $\mathcal{O}$, and whose intensional predicates include a special n-ary predicate Ans. We say that $q_{\mathcal{O}}$ is *safe* if all of its rules are safe. The semantics of $q_{\mathcal{O}}$ is provided relative to an OBDM system. Given an OBDM system $\langle J, D \rangle$, an interpretation for $q_{\mathcal{O}}$ is a pair $I = (mod(\langle J, D \rangle), f)$, where f is a first-order interpretation

(with domain $Const$) for the predicates in $\mathcal{P}$. As usual, we may also see f as the set of facts $\{p(\bar{c}) \mid \bar{c} \in p^f\}$. We now define when I satisfies a $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ rule.

- I satisfies a rule of the form (1) if the first-order formula $\forall \bar{x}.\gamma(\bar{x}) \rightarrow \alpha_1(\bar{x}_1) \vee \ldots \vee \alpha_n(\bar{x}_n)$ is true in f,
- I satisfies a rule of the form (2) if for all tuples $\bar{c}$ of constants in $Const$, the fact that the first-order formula $\exists \bar{z}.\phi(\bar{c}, \bar{z}) \wedge \xi(\bar{c})$ is satisfied by every model in $mod(\langle J, D\rangle)$ implies that f satisfies the first-order formula $\exists \bar{y}_j.\gamma_j(\bar{c}_j, \bar{y}_j)$, for some $j = 1, \ldots, n$. Observe that, if the rule is unsafe, $\bar{c}_j$ may contain constants of $Const$ that do not occur in $\bar{c}$.

An interpretation I for $q_{\mathcal{O}}$ is called a *model* of $q_{\mathcal{O}}$ if all the rules of $q_{\mathcal{O}}$ are satisfied by I. Finally, we define the notion of answers to an n-ary $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ query $q_{\mathcal{O}}$ w.r.t. an OBDM system $\langle J, D\rangle$, denoted by $q_{\mathcal{O}}^{J,D}$, as follows: $\{\bar{c} \in Dom(J, D, q_{\mathcal{O}})^n \mid \bar{c} \in Ans^f$ for each model $(mod(\langle J, D\rangle), f)$ of $q_{\mathcal{O}}\}$.

Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ be an OBDM specification such that $\mathcal{O} = \emptyset$, $\mathcal{S} = \{s_1/2, s_2/1, s_3/1\}$ and $\mathcal{M} = \{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4\}$, where:

$$
\begin{array}{lllll}
\mathbf{m}_1: & \forall x_1, x_2.s_1(x_1, x_2) & \rightarrow & \mathsf{E}(x_1, x_2) \\
\mathbf{m}_2: & \forall x.s_1(x, x) & \rightarrow & \mathsf{SN}(x) \\
\mathbf{m}_3: & \forall x.s_2(x) & \rightarrow & \exists z.\mathsf{E}(x, z) \\
\mathbf{m}_4: & \forall x.s_3(x) & \rightarrow & \mathsf{E}(x, x)
\end{array}
$$

The mapping $\mathcal{M}$ of J establishes how predicates in the schema $\mathcal{S}$ relate to the ontology predicates E (which stands for edge) and SN (which stands for special node).

Let us now consider the following safe $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ query $q_{\mathcal{O}}$ over J. $q_{\mathcal{O}}$ returns all the pairs (v_1, v_2) of special nodes that are *known to be distinct* and such that there is a path from v_1 to v_2 passing only for nodes *known to be special*:

$$
\begin{array}{lll}
\mathbf{K}(\mathsf{E}(x_1, x_2) \wedge \mathsf{SN}(x_1) \wedge \mathsf{SN}(x_2)) & \rightarrow & T_1(x_1, x_2) \\
\mathbf{K}(\mathsf{SN}(x_1) \wedge \mathsf{SN}(x_2) \wedge x_1 \neq x_2) & \rightarrow & T_2(x_1, x_2) \\
T_1(x_1, y) \wedge T_1(y, x_2) & \rightarrow & T_1(x_1, x_2) \\
T_1(x_1, x_2) \wedge T_2(x_1, x_2) & \rightarrow & Ans(x_1, x_2)
\end{array}
$$

□

The following proposition shows that $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ is a monotone query language.

Every $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ query over J is in $\mathfrak{M}^J$, for every OBDM specification J.

The semantics should make it clear that $\mathbf{K}$ is the knowledge operator in the S5 epistemic logic: the formula $\mathbf{K}A$ should be read as "A is known (i.e., logically implied) by the system" [13]. Therefore, when accessing the information modeled by $\langle J, D\rangle$, a $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ query extracts what is known by the system, and this characteristic is crucial for not falling into undecidability resulting from using Datalog rules jointly with Description Logics (see [14, 15]), as stated in the following proposition. Let Σ be an OBDM system. Answering safe $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ queries w.r.t. Σ is decidable if and only if answering CQs w.r.t. Σ is decidable[1].

Although our framework is general enough to consider any DL for expressing ontologies and any query language for expressing source queries, in the rest of this paper we will carry out our investigation in the following setting: (i) ontologies are expressed in *DL-Lite*$_{RDFS}$, and (ii) source queries are expressed as UCQs.

[1]With answering we implicitly refer to the associated *recognition problem*, i.e., check whether a tuple is in the answer to a query.

At this point, one may wonder whether $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ is the right language to express monotone abstractions in this setting. While a thorough analysis of the language is outside the scope of the present paper, the following proposition provides a positive answer to this question, at least from the computational point of view.

In our setting, (i) answering safe $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ queries is in coNP in data complexity, and (ii) there exists an OBDM specification J and a CQ $q_{\mathcal{S}}$ such that, given an $\mathcal{S}$-database D, answering the $\mathfrak{M}$-maximally sound J-abstraction of $q_{\mathcal{S}}$ is coNP-hard in data complexity.

To ease the presentation, from now on we assume that mappings and source queries do not mention constants. However, all our results can be straightforwardly adapted to the case where constants are allowed.

3. Monotone approximations of abstractions

In this section we investigate the problem of the existence of $\mathfrak{M}$-minimally complete and $\mathfrak{M}$-maximally sound abstractions, in the restricted setting specified at the end of the previous section. In particular, we start by showing that $\mathfrak{M}$-minimally complete abstractions of UCQ source queries always exist and can be expressed in $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$. Then, because of the lack of space, we will just mention the corresponding result on $\mathfrak{M}$-maximally sound abstractions.

Given a CQ $q_{\mathcal{S}} = \{\bar{x} \mid \exists \bar{y}.\phi(\bar{x}, \bar{y})\}$, the subroutine $\mathsf{SaturateQ}(q_{\mathcal{S}})$ computes a UCQ$^{\neq}$ in the following way: for each possible unifier μ on the variables in $\bar{x} \cup \bar{y}$ such that $\mu(x) \in \bar{x}$ for each $x \in \bar{x}$, $\mathsf{SaturateQ}(q_{\mathcal{S}})$ contains a query obtained from $\mu(q_{\mathcal{S}})$ by adding the inequality atom $t_1 \neq t_2$ for each pair of distinct variables t_1, t_2 occurring in $\mu(q_{\mathcal{S}})$. For a UCQ $q_{\mathcal{S}}$, we denote by $\mathsf{SaturateQ}(q_{\mathcal{S}})$ the UCQ$^{\neq}$ obtained by applying $\mathsf{SaturateQ}(q)$ to each disjunct q of $q_{\mathcal{S}}$. We write each CQ$^{\neq}$ q generated by $\mathsf{SaturateQ}(q_{\mathcal{S}})$ as $q = \{\bar{x} \mid \exists \bar{y}.\phi(\bar{x}, \bar{y}) \wedge \xi(\bar{x}, \bar{y})\}$, where $\phi(\bar{x}, \bar{y})$ and $\xi(\bar{x}, \bar{y})$ are the conjunctions of the relational atoms over $\mathcal{S}$ and of the inequality atoms, respectively, occurring in the body of q.

Moreover, for an OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and a CQ$^{\neq}$ $q = \{\bar{x} \mid \exists \bar{y}.\phi(\bar{x}, \bar{y}) \wedge \xi(\bar{x}, \bar{y})\}$ over $\mathcal{S}$, we denote by r_q the following $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ rule of form (2):

$$r_q = \mathbf{K}(\exists \bar{z}.\mathcal{M}(q) \wedge \xi(\bar{x}, \bar{y})) \rightarrow Ans(\bar{x}),$$

where (*i*) $\mathcal{M}(q)$ is computed by simply ignoring the inequality atoms and chasing the set of relational atoms occurring in the body of q; (*ii*) $\bar{y} \subseteq \bar{y}$ is the subset of the existential variables of q occurring in $\mathcal{M}(q)$; (*iii*) $\bar{z}$ are the fresh variables introduced when computing $\mathcal{M}(q)$; and (*iv*) $\xi(\bar{x}, \bar{y})$ is the conjunction of the inequality atoms obtained from $\xi(\bar{x}, \bar{y})$ by removing all those atoms of the form $y \neq t$ and $t \neq y$ in which y is an existential variable occurring in $\bar{y}$ but not in $\bar{y}$ (i.e., not in $\mathcal{M}(q)$) and t is any other possible variable. Observe that the epistemic operator is exploited to bind the existential variables coming from q. This is achieved by pushing the subset $\bar{y}$ of the existential variables $\bar{y}$ of q occurring in $\mathcal{M}(q)$ inside the $\mathbf{K}$ operator.

We are now ready to present the algorithm $\mathfrak{M}$-MinComplete for computing the $\mathfrak{M}$-minimally complete J-abstractions. Given an OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and a UCQ $q_{\mathcal{S}}$ over $\mathcal{S}$ such that $\mathsf{SaturateQ}(q_{\mathcal{S}}) = q_1 \cup \ldots \cup q_n$, $\mathfrak{M}$-$\mathsf{MinComplete}(J, q_{\mathcal{S}})$ outputs the $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ query $q_{\mathcal{O}} = \{r_{q_1}, \ldots, r_{q_n}\}$ over J.

Consider the OBDM specification J illustrated in Example 2 and the CQ $q_{\mathcal{S}} = \{(x) \mid \exists y.s_1(x,y)\}$ over $\mathcal{S}$. One can verify that $\mathfrak{M}$-MinComplete$(J, q_{\mathcal{S}})$ returns the following safe $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ query $q_{\mathcal{O}}$ over J asking for all those nodes v such that either v is connected to a node v' known to be different from v or v is a special node with a self-loop:

$$\begin{array}{lcl} \mathbf{K}(\mathsf{E}(x,y) \wedge x \neq y) & \rightarrow & Ans(x) \\ \mathbf{K}(\mathsf{E}(x,x) \wedge \mathsf{SN}(x)) & \rightarrow & Ans(x) \end{array}$$

Note that $q_{\mathcal{O}}$ is a better complete approximation than the query $\{(x) \mid \exists y.\mathsf{E}(x,y)\}$, which is the UCQ-minimally complete J-abstraction of $q_{\mathcal{S}}$ [6]. □

$\mathfrak{M}$-MinComplete$(J, q_{\mathcal{S}})$ terminates and returns the unique (up to J-equivalence) $\mathfrak{M}$-minimally complete J-abstraction of $q_{\mathcal{S}}$. Furthermore, we observe that the result of $\mathfrak{M}$-MinComplete$(J, q_{\mathcal{S}})$ is independent from the assertions occurring in the ontology of the OBDM specification J. Similar results can be obtained for OBDM specifications based on more expressive Horn DL ontologies.

Before concluding this section, we observe that $\mathfrak{M}$-MinComplete$(J, q_{\mathcal{S}})$ may return unsafe $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ queries. Nevertheless, these queries enjoy nice computational properties in our setting.

Let $q_{\mathcal{O}}$ be a $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ query with only rules of the form $\mathbf{K}(\exists \bar{z}.\phi(\bar{x},\bar{z}) \wedge \xi(\bar{x})) \rightarrow Ans(\bar{x}_a)$. In our setting, (*i*) answering $q_{\mathcal{O}}$ is in PTime in data complexity, and (*ii*) if $q_{\mathcal{O}}$ is safe, then it is possible to compute a $\mathrm{UCQ}^{\neq}$ q_r over $\mathcal{S}$ such that $q_{\mathcal{O}}^{J,D} = q_r^D$, for each $\mathcal{S}$-database D.

As for best sound approximations of abstractions, it can be shown (cf. details in [10]) that, similarly to the case of best complete abstractions, an algorithm exists that, given a UCQ $q_{\mathcal{S}}$, terminates and computes the unique (up to J-equivalence) $\mathfrak{M}$-maximally sound J-abstraction of $q_{\mathcal{S}}$, expressed as a $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ query. Thus, we have the following:

Given a UCQ $q_{\mathcal{S}}$, both the unique (up to J-equivalence) $\mathfrak{M}$-minimally complete and the $\mathfrak{M}$-maximally sound J-abstraction of $q_{\mathcal{S}}$ always exist and can be expressed as $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ queries. Consider the OBDM specification J illustrated in Example 2. One can verify that the following set $\mathcal{R}$ of safe $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$ rules is the unique (up to J-equivalence) $\mathfrak{M}$-maximally sound J-abstraction of $q_{\mathcal{S}}$:

$$\begin{array}{lcl} \mathbf{K}(\mathsf{E}(x_1,x_2) \wedge x_1 \neq x_2) & \rightarrow & s_1(x_1,x_2) \\ \mathbf{K}(\mathsf{E}(x,x)) & \rightarrow & s_3(x) \vee s_1(x,x) \\ \mathbf{K}(\exists z.\mathsf{E}(x,z)) & \rightarrow & s_2(x) \vee \exists y.s_1(x,y) \vee s_3(x) \\ \mathbf{K}(\mathsf{SN}(x)) & \rightarrow & s_1(x,x) \end{array}$$

□

4. Perfect Abstractions

It follows from Theorem 3 that either the perfect abstraction of a source UCQ can be expressed in $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$, or it cannot be expressed as a monotone query (if it exists at all). We now present an algorithm that, given an OBDM specification J and a source UCQ $q_{\mathcal{S}}$, returns the perfect J-abstraction of $q_{\mathcal{S}}$, if and only if it exists and is in $\mathfrak{M}$. To this aim, we make use of Proposition 3, and refer to the q_r defined in that proposition as the *rewriting of* $q_{\mathcal{O}}$ *w.r.t.* J.

Our algorithm, that we call $\mathfrak{M}$-Perfect, goes as follows. Given an OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and a UCQ $q_{\mathcal{S}}$ over $\mathcal{S}$ as input: if (*i*) $q_{\mathcal{O}} = \mathfrak{M}$-MinComplete$(J, q_{\mathcal{S}})$ is safe and

(*ii*) $q_r \sqsubseteq_{\mathcal{S}} q_{\mathcal{S}}$; then **return** $q_{\mathcal{O}}$; otherwise, **report** "no perfect J-abstraction of $q_{\mathcal{S}}$ is in $\mathfrak{M}$".

In Example 3, $\mathfrak{M}$-Perfect$(J, q_{\mathcal{S}})$ returns $q_{\mathcal{O}}$, which is the perfect J-abstraction of $q_{\mathcal{S}}$.

We conclude this section by establishing termination and correctness of the $\mathfrak{M}$-Perfect algorithm.

$\mathfrak{M}$-Perfect$(J, q_{\mathcal{S}})$ terminates and returns the unique (up to J-equivalence) perfect J-abstraction of $q_{\mathcal{S}}$ if and only if such an abstraction can be expressed in $\mathfrak{M}$.

5. Conclusion

We presented a thorough study of monotone abstractions of UCQs in OBDM systems. We proved that best approximations of such abstractions always exist and introduced a query language, $\mathsf{Datalog}^{\vee}_{\mathbf{K}}$, that captures them. Directions for future work are many. In the context of monotone abstractions, we would like to investigate the case of more expressive ontology languages, e.g., *DL-Lite*$_{\mathcal{R}}$, as well as more expressive source query languages, e.g., unions of conjunctive queries with inequalities and disjunctive Datalog. Finally, the problem of checking whether given best approximations are expressible in simpler and more user friendly languages remains open.

Acknowledgements

This work has been partially supported by the ANR AI Chair INTENDED (ANR-19-CHIA-0014), by MIUR under the PRIN 2017 project "HOPE" (prot. 2017MMJJRE), and by the EU under the H2020-EU.2.1.1 project TAILOR, grant id. 952215.

References

[1] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, R. Rosati, Linking data to ontologies, Journal on Data Semantics X (2008) 133–173. doi:10.1007/978-3-540-77688-8_5.

[2] M. Lenzerini, Managing data through the lens of an ontology, AI Magazine 39 (2018) 65–74.

[3] G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, R. Rosati, Using ontologies for semantic data integration, in: A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years, 2018, pp. 187–202.

[4] G. Cima, Preliminary results on ontology-based open data publishing, in: Proceedings of the Thirtieth International Workshop on Description Logics (DL 2017), volume 1879 of *CEUR Electronic Workshop Proceedings*, 2017.

[5] C. Lutz, J. Marti, L. Sabellek, Query expressibility and verification in ontology-based data access, in: Proceedings of the Sixteenth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2018), 2018, pp. 389–398.

[6] G. Cima, M. Lenzerini, A. Poggi, Semantic characterization of data services through ontologies, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019), 2019, pp. 1647–1653.

[7] G. Cima, M. Lenzerini, A. Poggi, Non-monotonic ontology-based abstractions of data services, in: Proceedings of the Seventeenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2020), 2020, pp. 243–252.
[8] G. Cima, M. Console, M. Lenzerini, A. Poggi, Abstraction in Data Integration, in: Proceedings of the Thirty-Sixth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2021), 2021, pp. 1–11.
[9] G. Cima, Abstraction in Ontology-based Data Management, Ph.D. thesis, Sapienza University of Rome, 2020.
[10] G. Cima, M. Console, M. Lenzerini, A. Poggi, Monotone abstractions in ontology-based data management, in: Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI 2022), 2022.
[11] S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison Wesley Publ. Co., 1995.
[12] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. F. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation and Applications, Cambridge University Press, 2003.
[13] H. J. Levesque, G. Lakemeyer, The Logic of Knowledge Bases, The MIT Press, 2001.
[14] A. Y. Levy, M.-C. Rousset, Combining Horn rules and description logics in CARIN, Artificial Intelligence 104 (1998) 165–209.
[15] D. Calvanese, R. Rosati, Answering recursive queries under keys and foreign keys is undecidable, in: Proceedings of the Tenth International Workshop on Knowledge Representation meets Databases (KRDB 2003), volume 79 of *CEUR Electronic Workshop Proceedings*, 2003.