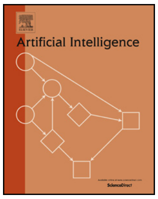
Artificial Intelligence 323 (2023) 103976

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



The notion of Abstraction in Ontology-based Data Management ☆

Gianluca Cima *, Antonella Poggi, Maurizio Lenzerini

Sapienza University of Rome, Italy

ARTICLE INFO

Article history:
Received 31 December 2021
Received in revised form 15 January 2023
Accepted 13 July 2023
Available online 20 July 2023

Keywords:
Abstraction
Ontology-based Data Management
Computational Complexity

ABSTRACT

We study a novel reasoning task in Ontology-based Data Management (OBDM), called Abstraction, which aims at associating formal semantic descriptions to data services. In OBDM a domain ontology is used to provide a semantic layer mapped to the data sources of an organization. The basic idea of the work presented in this paper is to explain the semantics of a data service in terms of a query over the ontology. We illustrate a formal framework for this problem, based on three different notions of abstraction, called sound, complete, and perfect, respectively. We present a thorough complexity analysis of two computational problems, namely verification (checking whether a query is an abstraction of a given data service), and computation (computing an abstraction of a given data service).

1. Introduction

In the last years the interest in using Artificial Intelligence (AI) for Data Management has grown considerably. One obvious area where AI is greatly effective is data analytics, which is the process of identifying, modeling, and communicating meaningful patterns of data. In particular, machine learning is being used in a plethora of ways for discovering insights, finding new patterns, and detecting novel relationships in the data. But data analytics is not the only area where AI can be of great value for Data Management: data modeling can also substantially benefit from AI. As all data scientists know, the quality of data analytics and data-driven decision-making heavily depends on the quality of the data that are input to the discovery process, and this in turn depends on the process used for gathering, structuring and making sense of available data. During such modeling process, data at the relevant sources must be collected, classified, interpreted, and their semantics must be characterized, so as to integrate them with data coming from other sources, cleansed, formatted, and categorized using a precise specification language.

Ontology-based Data Management (OBDM) [2] is a paradigm introduced with the goal of coping with the above issues. The key idea of OBDM is to apply suitable techniques from the area of Knowledge Representation and Reasoning for a new way to carry out Data Management and Data Governance, based on the principle of managing heterogeneous data through the lens of an ontology. OBDM resorts to a three-level architecture, constituted by the ontology, the data layer, and the mapping between the two. The *data layer* is constituted by the existing data sources that are relevant for the domain of interest. The *ontology* is a declarative and explicit representation of such domain, given in terms of formal and high-level assertions which describe the domain in terms of classes of objects, called *concepts*, and relationships among objects,

☆ This article is an expanded version of our IJCAI-19 conference publication [1].

* Corresponding author.

E-mail addresses: cima@diag.uniroma1.it (G. Cima), poggi@diag.uniroma1.it (A. Poggi), lenzerini@diag.uniroma1.it (M. Lenzerini).

https://doi.org/10.1016/j.artint.2023.103976

called *roles*. The *mapping* is a set of declarative assertions specifying how the available sources in the data layer relate to the concepts and the roles in the ontology. One of the distinguishing features of the whole approach is that users of the information system will be freed from all the details of how to use the data at the sources, as they will express their needs, such as queries, quality checks or other governance tasks, in terms of the concepts and the roles described in the domain model. The system will reason about the ontology and the mappings, and will reformulate the needs in terms of appropriate calls to services provided for accessing the data sources.

In order to translate the users' needs expressed over the ontology into correct and efficient computations over the data sources, techniques typical of the two areas of Knowledge Representation and Automated Reasoning are crucial. Indeed, most of the literature about managing data sources through an ontology [3–5,2,6,7] deals with user queries expressed over the ontology, and studies the problem of answering such queries, by finding a so-called *ontology-to-source rewriting*, i.e., a query over the source schema that, once executed over the data, provides the *certain answers* to the original query, i.e., those answers that are entailed from the logical specification of the OBDM system. Hence, Automated Reasoning techniques have played a prominent role in such investigation.

However, translating queries from the ontology level to the data layer is not the only relevant task in the context of OBDM. In this paper we argue that another important reasoning task in the governance of Information Systems is what we call *Abstraction*, whose goal is to capture and express the semantics of a data service in terms of the ontology. The architecture of many modern Information Systems is in fact based on data services [8], i.e., services deployed on top of data sources, other services, and/or applications to encapsulate a wide range of data-centered operations. Such architecture is crucial for the *Data-As-A-Service* [9] paradigm. However, in order to realize such paradigm, in particular to foster the adoption and the reuse of data services, it is of vital importance to well document and clearly specify their semantics, bringing them into compliance with the FAIR guiding principles [10], i.e., make them Findable, Accessible, Interoperable, and Reusable (FAIR). While most current techniques manually associate APIs (Application Programming Interface) to data services, and describe their intended meaning with ad-hoc methods, often using natural language or complex metadata [11], we propose a new approach, whose goal is to automatically associate formal semantic descriptions to data services. Since we base our proposal on OBDM, we envision a method by which the semantics of data services is expressed using the elements of the domain ontology, which is assumed to be familiar to the consumers of data services. But how can we automatically produce a semantic characterization of a data service, having an OBDM specification available? The idea is to exploit Abstraction, a new reasoning task that works as follows: we express the data service in terms of a query over the sources, and we aim at automatically deriving the query over the ontology that best describes the data service, given the mapping. The following example illustrates this idea.

Example 1.1. Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \{\ \exists \mathsf{TeachesTo} \sqsubseteq \mathsf{Professor},\ \ \mathsf{Student} \sqsubseteq \neg\mathsf{Professor},\ \ \mathsf{Student} \sqsubseteq \exists\mathsf{HasGuarantor},\ \ \exists\mathsf{RegisteredTo} \sqsubseteq \mathsf{Student}\ \}$
- $\mathcal{S} = \{\ s_1, s_2, s_3, s_4, s_5\ \}$
- $\mathcal{M} = \{\ m_1, m_2, m_3, m_4, m_5\ \}$, where:

$$\begin{array}{lrcl} m_1: & s_1(x,y) & \rightarrow & \mathsf{TeachesTo}(x,y) \\ m_2: & s_2(x) & \rightarrow & \mathsf{Professor}(x) \\ m_3: & s_3(x) & \rightarrow & \mathsf{RegisteredTo}(x,\mathsf{Sapienza}) \\ m_4: & s_3(x) \wedge s_4(x) & \rightarrow & \mathsf{HasGuarantor}(x,x) \\ m_5: & \exists y. s_4(x) \wedge s_5(y,x) & \rightarrow & \exists z.\mathsf{HasGuarantor}(z,x) \end{array}$$

where x, y, z are variables, and Sapienza is a constant.

Let the data service be expressed as the union of conjunctive query (UCQ) $q_{\mathcal{S}} = \{(x) \mid s_2(x)\} \cup \{(x) \mid \exists y. s_1(x,y)\}$ over the source schema $\mathcal{S}$. Conceivably, by inspecting the mapping assertions in $\mathcal{M}$ and the ontology assertions in $\mathcal{O}$, one can argue that the query $q_{\mathcal{O}}$ over the ontology $\mathcal{O}$ that characterizes the data service $q_{\mathcal{S}}$ at best w.r.t. the OBDM specification J is $q_{\mathcal{O}} = \{(x) \mid \mathsf{Professor}(x)\}$. △

Note that the problem of Abstraction is somehow reversed with respect to query answering: while in the latter we start with a query $q_{\mathcal{O}}$ over the ontology and we aim at a query over the sources computing the certain answers to $q_{\mathcal{O}}$, here we start with a source query $q_{\mathcal{S}}$ and we aim at deriving its abstraction, i.e., a corresponding query over the ontology, the one providing the semantics of $q_{\mathcal{S}}$ in terms of the ontology $\mathcal{O}$. Thus, Abstraction is a sort of reverse engineering problem, which is a novel aspect in the research on OBDM.

Motivations Besides for semantically characterizing data services, the notions introduced in this paper are relevant in a plethora of other application scenarios, among which we mention:

- *Open data publishing*: Current practices for publishing open data focus essentially on providing extensional information (often in very simple forms, such as CSV files), and carry out the task of documenting data mostly by using metadata expressed in natural languages, or in terms of record structures. As a consequence, the semantics of nowadays most available datasets is not formally expressed in a machine-readable form. However, following the ideas in [12], the notion of Abstraction can be used to automatically provide the semantics of open datasets and open APIs published by public or private organizations, which is a key aspect for unchaining all the potentials of open data [13]. Indeed, we encountered the need of Abstraction during a joint project on OBDM with a public statistical research institute [14]. 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. 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. In order to illustrate this scenario, consider the OBDM specification J introduced in Example 1.1 and assume it belongs to an organization that aims at publishing the dataset obtained by evaluating $q_{\mathcal{S}}$ over the current state of the $\mathcal{S}$-database D. Suppose also that $\mathcal{O}$ is shared among the organization stakeholders. The dataset can then be annotated with the abstraction of $q_{\mathcal{S}}$, which explicits that the dataset contains all professors, where the concept "Professor" is described in $\mathcal{O}$. Note that this is compliant with the FAIR principles according to which the dataset has to be Interoperable and Reusable, since the abstraction provides the semantics of the dataset content in terms of a shared vocabulary, i.e., the ontology.

- *Checking the quality of mappings*: In [15], the concept of *realization* of source queries, similar to one of the notions studied here, is used for checking whether the mapping provides the right coverage for expressing the relevant data services at the ontology level. Thus, Abstraction can be used as a tool to help decide whether or not the existing data sources and/or the ontology need to be updated [16]. Looking at Example 1.1 and assuming that $q_{\mathcal{S}}$ expresses a data service particularly relevant for the organization to whom J belongs to, it is crucial that the mapping $\mathcal{M}$ provides the right coverage for expressing $q_{\mathcal{S}}$ using $\mathcal{O}$. Then, the fact that an abstraction exists for $q_{\mathcal{S}}$ provides the guarantee that the mapping is adequate, as far as it concerns $q_{\mathcal{S}}$.
- *Source profiling*: Our notions are useful for a semantic-based approach to source profiling [17,18], in particular for describing the structure and the content of a data source in terms of the business vocabulary. Consider again the OBDM specification of Example 1.1 and assume that, with the aim of equipping data with an adequate documentation (e.g., before a system migration), the data owner wishes to describe the content of the source s_1 in terms of the business domain, i.e., of the ontology $\mathcal{O}$. He/she can achieve this by computing the abstraction of the query $\{(x) \mid \exists y.s_1(x,y)\}$, which, in fact, leads to claim that s_1 contains professors who teach.
- *Explaining classifiers*: Abstraction can also be used in the Explainable Artificial Intelligence field. Indeed, suppose to acquire the outcome of a binary classifier over tuples of data sources in the information system; then, it is possible to semantically describe the choices taken by such a classifier by deriving an ontology-mediated query whose answers include all the tuples classified positively, and none of the tuples classified negatively [19,20].
- *Synthetizing suitable specifications of processes in a microservice architecture*: Finally, if the databases that are local to microservices are mapped to an ontology, Abstraction can be used to provide a semantic description of processes orchestrating several data-driven microservices, thus obtaining a virtual, unified characterization of a set of distributed, autonomous computations [21].

Contributions The contributions provided by this paper can be summarized as follows.

- We propose a formal framework for the problem of semantically characterizing a data service through an ontology. We introduce the notions of *perfect*, *sound*, and *complete* abstraction, and we define two basic reasoning tasks, namely *verification* and *computation*. The former checks whether a given query is an abstraction of a data service, whereas the latter computes one such abstraction. We show that, although the ideal notion is the one of perfect abstraction, there are cases where, with the given mapping, no query over the ontology can precisely characterize the data service at hand. Thus, we also introduce *maximally sound* and *minimally complete* abstractions, which intuitively aim at approximating a perfect abstraction of a data service at best, with the goal of either precision (maximally sound abstraction), or recall (minimally complete abstraction).
- We study both the verification, and the computation problem for complete, sound, and perfect abstractions in one of the most popular OBDM setting considered in the literature, namely where the ontology language is *DL-Lite*$_{\mathcal{R}}$ [22], the mapping language follows the *Global-and-Local-as-Views (GLAV)* approach, i.e., each mapping assertion maps a conjunctive query (CQ) over the source to a CQ over the ontology [23,24], and when both the data service and the abstraction are expressed as unions of CQs. In particular, for perfect and complete abstractions we present algorithms for verification and computation, and characterize the complexity of both tasks. For the case of sound abstractions, we do the same for verification, and then we precisely determine the cases where a maximally sound abstraction is not guaranteed to exist.
- We single out a restricted scenario that is still meaningful from the point of view of expressive power, and guarantees the existence of maximally sound abstractions. The restricted scenario is obtained from the general one by (i) introducing a specific setting for OBDM specifications by limiting the ontology language to *DL-Lite*$_{\mathsf{RDFS}}$ [25,26] as well as

limiting the mapping assertions to *Pure Global-as-View* (i.e., GAV mapping without constants and repeated variables in the head, which is how GAV was originally defined in the data integration literature [23]), and (*ii*) limiting the data service to be expressed as a union of CQs with join-free existential variables (UCQJFEs). In such restricted scenario, we provide algorithms and complexity results for verification and computation of maximally sound abstractions.

Related work This paper reports and extends the results previously presented in [27,12,1]. More specifically, with respect to [12,1], besides providing a comprehensive overview of the relevance of the Abstraction reasoning task (see the current Introduction section), the present work further generalizes the source query language for specifying data services, by allowing constants and repeated variables to appear in the target list, and provides all the complete proofs as well as several additional examples. Apart from [12,1,27], to the best of our knowledge, Abstraction has been (partially) addressed only in [15], under the name of *realization*. In particular, [15] focuses on both *DL-Lite*$_{\mathcal{R}}$ and the $\mathcal{EL}$ family [28] of ontology languages, and studies only the case of perfect abstractions and only with GAV mapping assertions, under a slightly different semantics with respect to the one proposed here (cf. discussion at the beginning of Section 3). Importantly, due to the differences in the semantics, results of [15] on perfect abstractions over OBDM specifications with a *DL-Lite*$_{\mathcal{R}}$ ontology and GAV mapping assertions, implicitly hold only for consistent OBDM specifications.

Related to the problem studied here are also the work dealing with the problem of computing *(source-to-ontology) rewritings* over OBDM specifications with a *DL-Lite*$_{\mathcal{R}}$ ontology and (GLAV) mapping assertions [29]. In particular, we will show in Section 3, that given an OBDM specification J, a query $q_{\mathcal{O}}$ over the ontology, and a query $q_{\mathcal{S}}$ over the sources, $q_{\mathcal{O}}$ is a complete abstraction of $q_{\mathcal{S}}$ with respect to J if and only if $q_{\mathcal{S}}$ is a sound rewriting of $q_{\mathcal{O}}$ with respect to J. Hence, these works are strongly related to just one of the problems studied here, namely the verification problem for complete abstractions.

Finally, our investigation is related to *view-based query rewriting* [30,31]. In particular, given an OBDM specification with an empty ontology and a set of pure GAV mapping assertions, $q_{\mathcal{O}}$ is a sound abstraction of $q_{\mathcal{S}}$ with respect to J if and only if $q_{\mathcal{O}}$ is a rewriting of $q_{\mathcal{S}}$ over a set of disjunctive views including one view for each element of the ontology specified in the head of a mapping, where the definition of the view coincides with the body of the mapping. For more details on this relation between Abstraction and view-based query rewriting, we refer the interested reader to [32]. Interestingly, there are only few works [33,34] tackling view-based query rewriting in the presence of disjunctive views. None of these works, however, focuses on the problem of computing rewritings expressed as unions of CQs.

Plan of the paper The paper is organized as follows. After presenting in Section 2 notions preliminary to our investigation, in Section 3 we present our formal framework for Abstraction. Then, in Section 4, 5, and 6, we study verification and computation of complete, sound, and perfect abstractions, respectively. Section 7 first focuses on verification and computation of sound abstractions in a restricted scenario, and then mention the key results for a further restricted scenario. Finally, Section 8 concludes the paper, while Appendix A provides more proofs and Appendix B tackles the further restricted scenario mentioned at the end of Section 7.

2. Preliminaries

We assume basic knowledge about databases [35] and Description Logics (DLs) [36,37]. In what follows, we use $\sigma(x)$ to denote the *size* of object x.

Database and queries A *relational database schema* (or simply *schema*) $\mathcal{S}$ is a finite set of predicate symbols, each with a specific arity $n \geq 1$. Given a schema $\mathcal{S}$, an $\mathcal{S}$*-database* D is a finite set of *facts* of the form $s(\vec{c})$, where s is an n-ary predicate symbol of $\mathcal{S}$, and $\vec{c} = (c_1, \ldots, c_n)$ is an n-tuple of terms, where each term in a fact is a constant taken from a denumerable infinite set of symbols Const. Note that, when convenient, we treat tuples of terms as sets, in which cases we implicitly refer to the set composed by all the terms occurring in the tuple.

We sometime refer to a finite set of *atoms* $s(t_1, \ldots, t_n)$ over $\mathcal{S}$, often denoted as $\mathcal{K}$, where s is an n-ary predicate symbol of $\mathcal{S}$, and, for each $i = 1, \ldots, n$, the term t_i is either a constant in Const or a variable (such variables represent *unknown values* [38]). Each variable is taken from a denumerable infinite set of symbols denoted by Var, where $\mathsf{Const} \cap \mathsf{Var} = \emptyset$. We denote by $\mathsf{dom}(\mathcal{K})$ the set of all terms (i.e., constants and variables) occurring in $\mathcal{K}$.

In its general form, a *query* q over a schema $\mathcal{S}$ is a function that can be *evaluated* over an $\mathcal{S}$-database D to return a set of *answers* q^D, each answer being a tuple of constants. All such tuples have the same arity, which is the *arity* of q, denoted by $ar(q)$. When $ar(q) = 0$, the query is called *boolean*. We assume to deal with databases supporting at least a particular class of queries, named *conjunctive queries (CQs)* and unions thereof, which we express adopting the standard relational calculus notation. Specifically, the syntax of a CQ q of arity $n \geq 0$ over a schema $\mathcal{S}$ can be either $\texttt{false}/n$, or an expression of the form $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$, also denoted $q(\vec{t})$, where (i) $\vec{t}$, called the *target list* of q, is an n-tuple of terms; (ii) $\vec{x}$ is the tuple of variables occurring in $\vec{t}$, called the *distinguished variables* of q; (iii) $\vec{y}$ is a tuple of variables, called the *existential variables* of q; and (iv) $\phi(\vec{x}, \vec{y})$, called the *body* of q, is a finite conjunction of atoms, each one of the form $s(t'_1, \ldots, t'_k)$ where s is an k-ary predicate symbol of $\mathcal{S}$, and for each $j = 1, \ldots, k$ the term t'_j is either a constant or a variable occurring in $\vec{x}$ or $\vec{y}$. As usual, we impose that each distinguished variable and each existential variable appears in some atom of $\phi(\vec{x}, \vec{y})$. Given a CQ $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ over a schema $\mathcal{S}$, we call *freezing* of q, denoted by D_q, an $\mathcal{S}$-database associated to q, i.e., a set of facts over $\mathcal{S}$ obtained from ϕ by replacing each variable z with a different fresh constant denoted by c_z.

Example 2.1. Let $q = \{(x) \mid \exists y.s_1(x, y) \wedge s_2(x) \wedge s_3(y) \wedge s_4(y)\}$. Then a freezing of q is the following database $D_q = \{s_1(c_x, c_y), s_2(c_x), s_3(c_y), s_4(c_y)\}$, where c_x and c_y are distinct constants. △

Given a CQ $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$, we say that an existential variable $y \in \vec{y}$ is a *join existential variable* if it occurs more than once in the atoms of $\phi(\vec{x}, \vec{y})$. In what follows, we also consider the subclass of CQs without join existential variables, namely *conjunctive queries with join-free existential variables (CQJFEs)*. Obviously, a CQ q is also a CQJFE if there is no join existential variable occurring in q. Observe that CQJFEs subsumes the class of *full conjunctive queries (full CQs)* since it allows for non-join existential variables occurring in their bodies. Full CQs form a well-known class of queries studied in relational database theory (see, e.g., [39–41]), corresponding to the Select-Join fragment of *Relational Algebra* [42], i.e., the fragment of CQs without the *projection* operator. Finally, a UCQ (respectively, UCQJFE) is a *union* of a finite set of CQs (respectively, CQJFEs) with same arity, called its *disjuncts*.[1]

Example 2.2. The query $q_{\mathcal{S}} = \{(x) \mid s_2(x)\} \cup \{(x) \mid \exists y.s_1(x, y)\}$ is a UCQJFE since it is the union of two CQs, the former being a full CQ and the latter a CQJFE. Note, in particular, that the latter CQ contains an existentially quantified variable y occurring only once within the atoms of the disjunct. On the contrary, the query obtained by replacing the second disjunct of $q_{\mathcal{S}}$ with the CQ $\{(x) \mid \exists y.s_1(x, y) \wedge s_3(y)\}$ is a UCQ that is not a UCQJFE because its second disjunct contains an existentially quantified variable y that occurs twice within the atoms of the disjunct. △

To define the evaluation of CQs and UCQs over $\mathcal{S}$-databases, we resort to the notion of homomorphism. Given two (possibly infinite) sets of atoms $\mathcal{K}$ and $\mathcal{K}'$, a *homomorphism* from $\mathcal{K}$ to $\mathcal{K}'$ is a function $h : \mathsf{dom}(\mathcal{K}) \to \mathsf{dom}(\mathcal{K}')$ for which:

- $h(c) = c$ for each $c \in \mathsf{Const}$; and
- $h(\mathcal{K}) \subseteq \mathcal{K}'$,

where $h(\mathcal{K})$ is the image of $\mathcal{K}$ under h, i.e., $h(\mathcal{K}) = \{h(\alpha) \mid \alpha \in \mathcal{K}\}$, where, for each atom $\alpha = s(t_1, \ldots, t_n)$, $h(\alpha) = s(h(t_1), \ldots, h(t_n))$.

Given a CQ q of arity n, the evaluation of q over a (possibly infinite) set of atoms $\mathcal{K}$, denoted $q^{\mathcal{K}}$, is $\emptyset$ if q is $\texttt{false}/n$, otherwise, if q is of the form $\{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$, $q^{\mathcal{K}}$ is the set of n-tuples of terms $\vec{c}$ such that there exists a function h from $\mathsf{dom}(\phi) \cup \vec{c}_{\vec{t}}$ to $\mathsf{dom}(\mathcal{K}) \cup \vec{c}_{\vec{t}}$, where $\vec{c}_{\vec{t}}$ denotes the set of constants occurring in the target list $\vec{t}$, for which (i) $h(c) = c$ for each $c \in \vec{c}_{\vec{t}}$, (ii) the restriction of h to $\mathsf{dom}(\phi)$ is a homomorphism from ϕ to $\mathcal{K}$, and (iii) $h(\vec{t}) = \vec{c}$. As usual, for a tuple of terms $\vec{t} = (t_1, \ldots, t_n)$, $h(\vec{t})$ denotes $(h(t_1), \ldots, h(t_n))$. In what follows, we also say that this is a homomorphism from q to $\mathcal{K}$ (or also a homomorphism from $\phi(\vec{x}, \vec{y})$ to $\mathcal{K}$) with $h(\vec{t}) = \vec{c}$, and write $h(q)$ (or also $h(\phi(\vec{x}, \vec{y}))$). Finally, the evaluation of a UCQ over a set of atoms $\mathcal{K}$ is simply the union of the evaluation of its disjuncts over $\mathcal{K}$. As an usual convention, for a boolean CQ q, the evaluation of q over a set of atoms $\mathcal{K}$ amounts to $q^{\mathcal{K}} = \{()\}$ (also denoted by $\mathcal{K} \models q$) if there is a homomorphism from q to $\mathcal{K}$, and $\emptyset$ otherwise (also denoted $\mathcal{K} \not\models q$).

For two queries q_1 and q_2 of the same arity over a schema $\mathcal{S}$, we write $q_1 \sqsubseteq_{\mathcal{S}} q_2$ (or simply $q_1 \sqsubseteq q_2$ when $\mathcal{S}$ is clear) if $q_1^D \subseteq q_2^D$ for every $\mathcal{S}$-database D. Furthermore, we write $q_1 \equiv_{\mathcal{S}} q_2$ (or simply $q_1 \equiv q_2$ when $\mathcal{S}$ is clear) if both $q_1 \sqsubseteq q_2$ and $q_2 \sqsubseteq q_1$ hold. It is well-known that, if $q_1 = \{\vec{t_1} \mid \exists \vec{y}_1.\phi_1(\vec{x_1}, \vec{y_1})\}$ and $q_2 = \{\vec{t_2} \mid \exists \vec{y}_2.\phi_2(\vec{x_2}, \vec{y_2})\}$ are CQs over $\mathcal{S}$, then $q_1 \sqsubseteq q_2$ if and only if $\vec{t_1} \in q_2^{\phi_1}$, i.e., if and only if there is a homomorphism h from q_2 to ϕ_1 with $h(\vec{t_2}) = \vec{t_1}$ [44], and if both q_1 and q_2 are UCQs over $\mathcal{S}$, then $q_1 \sqsubseteq q_2$ if and only if for each disjunct q of q_1 there is a disjunct q' of q_2 such that $q \sqsubseteq q'$ [45]. The containment check between (U)CQs by means of homomorphisms can be trivially extended to deal with the special CQ $\texttt{false}/n$, too.

Given a CQ $q = \{(t_1, \ldots, t_n) \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ of arity n and an n-tuple of constants $\vec{c} = (c_1, \ldots, c_n)$, we denote by $q(\vec{c})$ the special CQ $\texttt{false}/n$ if there is some $i \in [1, n]$ for which $t_i \neq c_i$ and t_i is a constant, otherwise it is the boolean CQ $\{() \mid \exists \vec{y}.\phi(\vec{x}/\vec{c}, \vec{y})\}$, where $\phi(\vec{x}/\vec{c}, \vec{y})$ denotes the formula obtained from $\phi(\vec{x}, \vec{y})$ by replacing every occurrence of the term t_i with the constant c_i, for each $i \in [1, n]$.

Given a UCQ $q = q_1 \cup \ldots \cup q_m$ of arity n and an n-tuple of constants $\vec{c} = (c_1, \ldots, c_n)$, we denote by $q(\vec{c}) = q_1(\vec{c}) \cup \ldots \cup q_m(\vec{c})$ the boolean UCQ obtained from q by replacing the disjunct q_i with $q_i(\vec{c})$, for each $i \in [1, m]$.

Ontologies A DL *ontology* $\mathcal{O}$ is simply a TBox expressed in a specific DL. Sometimes we also need to view $\mathcal{O}$ as a schema, in which cases we implicitly refer to the finite set of unary and binary predicates corresponding to *atomic concepts* and *atomic roles*, respectively, which constitute the *alphabet* of $\mathcal{O}$. We assume that every ontology $\mathcal{O}$ comprises the atomic concepts $\top$ and $\bot$, called *universal concept* and *bottom concept*, respectively. Formally, a DL ontology $\mathcal{O}$ consists in a finite set of assertions over its alphabet built according to the syntax rules of the specific DL. In particular, we are interested in DL

[1] Observe that we allow for different target lists in the disjuncts of a UCQ. This class of queries is called *disjunction of CQs (DCQs)* in [43].

ontologies expressed in *DL-Lite*$_\mathcal{R}$, a member of the *DL-Lite* family [22] of DLs which underpins OWL 2 QL [46], i.e., the OWL 2 profile especially designed for the OBDM scenarios. In *DL-Lite*$_\mathcal{R}$, assertions have the following forms:

$$B_1 \sqsubseteq B_2 \qquad R_1 \sqsubseteq R_2 \qquad \text{(concept/role inclusion assertion)}$$

$$B_1 \sqsubseteq \neg B_2 \qquad R_1 \sqsubseteq \neg R_2 \qquad \text{(concept/role disjointness assertion)}$$

where B_1, B_2 are *basic concepts*, i.e., expressions of the form A, $\exists P$, or $\exists P^-$, with A and P denoting *atomic concepts*, and R_1, R_2 are *basic roles*, i.e., expressions of the form P, or P^-. We assume that $\bot$ never occurs in the right-hand side of inclusion assertions. This is without loss of generality, since each inclusion assertion of the form $B \sqsubseteq \bot$ is equivalent to $B \sqsubseteq \neg B$. Furthermore, we implicitly assume that each *DL-Lite*$_\mathcal{R}$ ontology $\mathcal{O}$ contains the inclusion assertion $B \sqsubseteq \top$, for each basic concept B built from the alphabet of $\mathcal{O}$.

We will also consider one sublanguage of *DL-Lite*$_\mathcal{R}$, namely *DL-Lite*$_{\mathsf{RDFS}}$ [25,26] (i.e., the DL-like part of RDFS [47]), where both disjointness assertions, and concepts of the forms $\exists P$ or $\exists P^-$ in the right-hand side of inclusion assertions, are ruled out.

The semantics of DL ontologies is specified through the notion of interpretation: an *interpretation* $\mathcal{I}$ for an ontology $\mathcal{O}$ is a pair $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$, where the *interpretation domain* $\Delta^\mathcal{I}$ is a non-empty, possibly infinite set of objects, and the *interpretation function* $\cdot^\mathcal{I}$ assigns to each atomic concept A a set of domain objects $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ (with $\top^\mathcal{I} = \Delta^\mathcal{I}$ and $\bot^\mathcal{I} = \emptyset$) and to each atomic role P a set of pairs of domain objects $P^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. For the constructs of *DL-Lite*$_\mathcal{R}$, the interpretation function extends to basic concepts and roles as follows: (*i*) $(\exists P)^\mathcal{I} = \{o \mid \exists o'.\ (o, o') \in P^\mathcal{I}\}$, (*ii*) $(\exists P^-)^\mathcal{I} = \{o \mid \exists o'.\ (o', o) \in P^\mathcal{I}\}$, and (*iii*) $(P^-)^\mathcal{I} = \{(o, o') \mid (o', o) \in P^\mathcal{I}\}$.

An interpretation $\mathcal{I}$ *satisfies* (*i*) a concept inclusion assertion $B_1 \sqsubseteq B_2$ (respectively, role inclusion assertion $R_1 \sqsubseteq R_2$) if $B_1^\mathcal{I} \subseteq B_2^\mathcal{I}$ (respectively, $R_1^\mathcal{I} \subseteq R_2^\mathcal{I}$), and (*ii*) a concept disjointness assertion $B_1 \sqsubseteq \neg B_2$ (respectively, role disjointness assertion $R_1 \sqsubseteq \neg R_2$) if $B_1^\mathcal{I} \cap B_2^\mathcal{I} = \emptyset$ (respectively, $R_1^\mathcal{I} \cap R_2^\mathcal{I} = \emptyset$). Also, an interpretation $\mathcal{I}$ satisfies a *DL-Lite*$_\mathcal{R}$ ontology $\mathcal{O}$, or, equivalently, $\mathcal{I}$ is a model of $\mathcal{O}$, denoted by $\mathcal{I} \models \mathcal{O}$, if $\mathcal{I}$ satisfies every assertion in $\mathcal{O}$. Finally, an ontology $\mathcal{O}$ logically implies a concept/role inclusion/disjointness assertion α if $\mathcal{I} \models \alpha$, for every model $\mathcal{I}$ of $\mathcal{O}$.

Note that, when convenient, we treat interpretations $\mathcal{I}$ for $\mathcal{O}$ as the (possibly infinite) set of *facts* of the form $A(c)$ or $P(c_1, c_2)$ that are *true according to* $\mathcal{I}$, i.e., are such that $c \in A^\mathcal{I}$ or, respectively, $(c_1, c_2) \in P^\mathcal{I}$, where $c, c_1, c_2 \in \Delta^\mathcal{I}$ and A and P denote an atomic concept and a atomic role of $\mathcal{O}$. Thus, we immediately obtain the notion of evaluation of a UCQ q over an interpretation $\mathcal{I}$: $q^\mathcal{I}$ is the evaluation of q over the set of facts in $\mathcal{I}$.

OBDM specification An *Ontology-based Data Management (OBDM) specification* [3,48] is a triple $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, where:

- $\mathcal{O}$ is a DL ontology;
- $\mathcal{S}$ is a relational database schema, also called *source schema*;
- $\mathcal{M}$ is a *mapping*, i.e., a finite set of mapping assertions relating $\mathcal{S}$ to $\mathcal{O}$, where each mapping assertion is a statement of the form $\forall\vec{x}.(\exists\vec{y}.\phi_\mathcal{S}(\vec{x}, \vec{y}) \rightarrow \exists\vec{z}.\psi_\mathcal{O}(\vec{x}, \vec{z}))$, with $\phi_\mathcal{S}(\vec{x}, \vec{y})$ and $\psi_\mathcal{O}(\vec{x}, \vec{z})$ finite conjunctions of atoms over $\mathcal{S}$ and $\mathcal{O}$, respectively [23,24]. We further assume that in every mapping assertion of the above form, the atomic concept $\bot$ does not occur in $\psi_\mathcal{O}(\vec{x}, \vec{z})$.[2]

Mapping assertions of the above general form are also called *GLAV (Global-and-Local-as-View)* mapping assertions. Special cases of GLAV mapping assertions are *GAV (Global-as-View)* and *LAV (Local-as-Views)* mapping assertions. A GAV mapping assertion is a GLAV mapping assertion in which the right-hand side of the implication is simply an atom without existential variables. Furthermore, a GAV mapping assertion is called *pure* if the atom $\psi_\mathcal{O}(\vec{x})$ does not have constants nor variables that are repeated more than once. A LAV mapping assertion is a GLAV mapping assertion in which the left-hand side of the implication is simply an atom without repeated variables, and all universally quantified variables appear in the right-hand side of the implication, i.e., it is an assertion of the form $\forall\vec{x}.(s(x_1, \ldots, x_n) \rightarrow \exists\vec{z}.\psi_\mathcal{O}(\vec{x}, \vec{z}))$, where s is an n-ary predicate symbol of $\mathcal{S}$ and $x_1, \ldots, x_n$ are pairwise different variables. Finally, we say that a mapping $\mathcal{M}$ is a GLAV (respectively, LAV, GAV, pure GAV) mapping if it consists of a finite set of GLAV (respectively, LAV, GAV, pure GAV) mapping assertions. For readability purposes, from now on we will drop the universal quantifiers in front of mapping assertions. The following example illustrates the definitions above.

Example 2.3. Consider again the OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ of Example 1.1. The mapping $\mathcal{M}$ is GLAV. In particular, it contains four mapping assertions, m_1, m_2, m_3, and m_4 that are GAV, of which two, m_1 and m_2 are also pure GAV, and three, m_1, m_2, and m_3, are also LAV. $\triangle$

We will make use of the notion of chase of a set of atoms with respect to a mapping. The *chase* is a fixpoint algorithm typically used to reason about data dependencies [49,50].[3] Formally, given a set of atoms $\mathcal{K}$ over a schema $\mathcal{S}$ and a mapping

[2] We could easily extend our work to the case where such assumption does not hold, but we would be forced to introduce some technicalities which are not interesting with respect to the goal of the paper.

[3] Here we implicitly refer to the *oblivious chase* [51] (also known as the *naive chase* [52]) rather than to the *standard chase* [53].

$\mathcal{M}$ relating $\mathcal{S}$ to an ontology $\mathcal{O}$, the chase of $\mathcal{K}$ with respect to $\mathcal{M}$, denoted by $\mathcal{M}(\mathcal{K})$, is a set of atoms over $\mathcal{O}$ computed as follows: (*i*) $\mathcal{M}(\mathcal{K})$ is initially set to the empty set, then (*ii*) for every GLAV assertion $\exists\vec{y}.\phi_{\mathcal{S}}(\vec{x},\vec{y}) \to \exists\vec{z}.\varphi_{\mathcal{O}}(\vec{x},\vec{z})$ in $\mathcal{M}$ and for every homomorphism h from $\phi_{\mathcal{S}}(\vec{x},\vec{y})$ to $\mathcal{K}$, we add to $\mathcal{M}(\mathcal{K})$ the image of the set of all atoms occurring in $\varphi_{\mathcal{O}}(\vec{x},\vec{z})$ under h', i.e., we set $\mathcal{M}(\mathcal{K}) := \mathcal{M}(\mathcal{K}) \cup h'(\varphi_{\mathcal{O}}(\vec{x},\vec{z}))$, where h' extends h by assigning to each variable $z \in \vec{z}$ a different fresh variable in Var not present in $\mathcal{M}(\mathcal{K})$. With a slight abuse of notation, given a CQ $q = \{\vec{t} \mid \exists\vec{y}.\phi(\vec{x},\vec{y})\}$ over a schema $\mathcal{S}$ and a mapping $\mathcal{M}$ relating $\mathcal{S}$ to an ontology $\mathcal{O}$, we denote by $\mathcal{M}(q)$ the conjunction of all the atoms obtained by chasing the set of atoms occurring in the body of q with respect to $\mathcal{M}$.

Example 2.4. Let $J = \langle\mathcal{O},\mathcal{S},\mathcal{M}\rangle$ be the OBDM specification introduced in Example 1.1 and $q = \{(x) \mid \exists y.s_1(x,y) \wedge s_2(x) \wedge s_3(y) \wedge s_4(y)\}$ be the query already discussed in Example 2.1. Then $\mathcal{M}(q)$ is the following conjunction of atoms: $\mathsf{TeachesTo}(x,y) \wedge \mathsf{Professor}(x) \wedge \mathsf{RegisteredTo}(y,\mathsf{Sapienza}) \wedge \mathsf{HasGuarantor}(y,y)$. △

Semantics of OBDM The semantics of an OBDM specification $J = \langle\mathcal{O},\mathcal{S},\mathcal{M}\rangle$ is given w.r.t. an $\mathcal{S}$-database D. Specifically, an interpretation $\mathcal{I} = \langle\Delta^{\mathcal{I}},\cdot^{\mathcal{I}}\rangle$ for $\langle J,D\rangle$ is an interpretation for $\mathcal{O}$ whose *interpretation domain* $\Delta^{\mathcal{I}}$ is equal to Const, and whose *interpretation function* $\cdot^{\mathcal{I}}$ further assigns to each constant $c \in \mathsf{Const}$ itself.[4]

Given an OBDM specification $J = \langle\mathcal{O},\mathcal{S},\mathcal{M}\rangle$, an $\mathcal{S}$-database D, and an interpretation $\mathcal{I}$ for $\langle J,D\rangle$, we say that the pair $\langle D,\mathcal{I}\rangle$ satisfies a mapping assertion $m = \exists\vec{y}.\phi_{\mathcal{S}}(\vec{x},\vec{y}) \to \exists\vec{z}.\psi_{\mathcal{O}}(\vec{x},\vec{z})$ occurring in $\mathcal{M}$, denoted by $\langle D,\mathcal{I}\rangle \models m$, if $\{\vec{x} \mid \phi_{\mathcal{S}}(\vec{x},\vec{y})\}^D \subseteq \{\vec{x} \mid \phi_{\mathcal{O}}(\vec{x},\vec{y})\}^{\mathcal{I}}$. Furthermore, we say that $\langle D,\mathcal{I}\rangle$ satisfies $\mathcal{M}$, denoted by $\langle D,\mathcal{I}\rangle \models \mathcal{M}$, if $\langle D,\mathcal{I}\rangle \models m$ for each $m \in \mathcal{M}$.

We are now ready to formalize the notion of model of an OBDM specification $J = \langle\mathcal{O},\mathcal{S},\mathcal{M}\rangle$ w.r.t. an $\mathcal{S}$-database D. An interpretation $\mathcal{I}$ for $\langle J,D\rangle$ is a *model for* J *relative to* D if (*i*) $\mathcal{I} \models \mathcal{O}$, and (*ii*) $\langle D,\mathcal{I}\rangle \models \mathcal{M}$. The set of models for J relative to D is denoted by $Mod_D(J)$, and D is said to be *consistent with* J if $Mod_D(J) \neq \emptyset$, *inconsistent with* J otherwise.

Example 2.5. Let us consider again the OBDM specification $J = \langle\mathcal{O},\mathcal{S},\mathcal{M}\rangle$ of Example 1.1, and let D be the following $\mathcal{S}$-database: $D = \{s_1(\mathsf{John},\mathsf{Max}), s_2(\mathsf{Julie}), s_3(\mathsf{Alfred}), s_3(\mathsf{Jim}), s_4(\mathsf{Gilda}), s_4(\mathsf{Diane}), s_4(\mathsf{Alfred}), s_5(\mathsf{Paris},\mathsf{Gilda})\}$. It is easy to verify that the following interpretation M (seen as a set of facts over $\mathcal{O}$) is a model for J relative to D:

$$\begin{aligned} M = \{&\mathsf{TeachesTo}(\mathsf{John},\mathsf{Max}), \mathsf{Professor}(\mathsf{John}), \mathsf{Professor}(\mathsf{Julie}), \mathsf{RegisteredTo}(\mathsf{Alfred},\mathsf{Sapienza}),\\ &\mathsf{RegisteredTo}(\mathsf{Jim},\mathsf{Sapienza}), \mathsf{HasGuarantor}(\mathsf{Alfred},\mathsf{Alfred}), \mathsf{HasGuarantor}(\mathsf{Steve},\mathsf{Gilda}),\\ &\mathsf{Student}(\mathsf{Alfred}), \mathsf{Student}(\mathsf{Jim})\}\end{aligned}$$

where Steve is a constant in Const not occurring in D. Thus, since $M \in Mod_D(J)$, we have $Mod_D(J) \neq \emptyset$, and therefore D is consistent with J. △

Certain answers In OBDM one of the main service of interest is query answering, i.e., computing the certain answers to queries posed over the ontology. Given an OBDM specification $J = \langle\mathcal{O},\mathcal{S},\mathcal{M}\rangle$, an $\mathcal{S}$-database D, and a query $q_{\mathcal{O}}$ over $\mathcal{O}$ of arity n, we denote by $cert^D_{q_{\mathcal{O}},J}$ the set of *certain answers* of $q_{\mathcal{O}}$ with respect to J and D, i.e., the set of n-tuples of constants $(c_1,\ldots,c_n)$ occurring in D, in $\mathcal{M}$, or in $q_{\mathcal{O}}$ such that $(c_1^{\mathcal{I}},\ldots,c_n^{\mathcal{I}}) \in q_{\mathcal{O}}^{\mathcal{I}}$, for each $\mathcal{I} \in Mod_D(J)$. Observe that, if D is inconsistent with J (i.e., $Mod_D(J) = \emptyset$), then the set of certain answers of any query $q_{\mathcal{O}}$ (over $\mathcal{O}$) with respect to J and D is trivially the set of all possible n-tuples of constants occurring in D, in $\mathcal{M}$, or in $q_{\mathcal{O}}$ (*ex falso quodlibet*).

Given two queries $q^1_{\mathcal{O}}, q^2_{\mathcal{O}}$ over $\mathcal{O}$, we write $cert_{q^1_{\mathcal{O}},J} \sqsubseteq cert_{q^2_{\mathcal{O}},J}$ if $cert^D_{q^1_{\mathcal{O}},J} \subseteq cert^D_{q^2_{\mathcal{O}},J}$ for every $\mathcal{S}$-database D, and we write $cert_{q^1_{\mathcal{O}},J} \sqsubset cert_{q^2_{\mathcal{O}},J}$ if (*i*) $cert_{q^1_{\mathcal{O}},J} \sqsubseteq cert_{q^2_{\mathcal{O}},J}$, and in addition (*ii*) $cert^D_{q^1_{\mathcal{O}},J} \subset cert^D_{q^2_{\mathcal{O}},J}$ for at least one $\mathcal{S}$-database D. Then, $q^1_{\mathcal{O}}$ and $q^2_{\mathcal{O}}$ are *equivalent with respect to* J, denoted by $cert_{q^1_{\mathcal{O}},J} \equiv cert_{q^2_{\mathcal{O}},J}$, if both $cert_{q^1_{\mathcal{O}},J} \sqsubseteq cert_{q^2_{\mathcal{O}},J}$ and $cert_{q^2_{\mathcal{O}},J} \sqsubseteq cert_{q^1_{\mathcal{O}},J}$ hold.

Also, given an OBDM specification $J = \langle\mathcal{O},\mathcal{S},\mathcal{M}\rangle$ and a query $q_{\mathcal{O}}$ over $\mathcal{O}$, a query $q_{\mathcal{S}}$ over $\mathcal{S}$ is a *sound* J*-rewriting of* $q_{\mathcal{O}}$, if for every $\mathcal{S}$-database D, $q^D_{\mathcal{S}} \subseteq cert^D_{q_{\mathcal{O}},J}$, while it is a *perfect* J*-rewriting of* $q_{\mathcal{O}}$, if for every $\mathcal{S}$-database D, $q^D_{\mathcal{S}} = cert^D_{q_{\mathcal{O}},J}$ [54].

Till the end of the section, we focus on OBDM specifications whose ontology is expressed in *DL-Lite*$_{\mathcal{R}}$. Also, to simplify the presentation and without loss of generality, we assume that neither mappings nor queries over the ontology mention constants. Finally, to correctly take into account the possible presence of atoms with predicate symbol $\top$ in the body of CQs, we assume that for each predicate symbol $s \in \mathcal{S}$ of arity n and for each $i = 1,\ldots,n$, $\mathcal{M}$ contains the mapping assertion $\exists y_1,\ldots y_{i-1},x,y_{i+1},\ldots,y_n.s(y_1,\ldots,y_{i-1},x,y_{i+1},\ldots,y_n) \to \top(x)$.[5] Importantly, under these assumptions and based on results of [55,29], if J is such that $\mathcal{O} = \emptyset$ and $\mathcal{M}$ is a GLAV mapping, then by splitting the GLAV mapping $\mathcal{M}$ into a GAV

[4] Thus, we adopt the *standard name assumption* and therefore the *unique name assumption*. Note, however, that all our results can be easily reformulated in a setting where those assumptions do not hold, as usual in OBDM [3].

[5] It is easy to verify that all such mapping assertions are pure GAV. Moreover, for each OBDM specification $J = \langle\mathcal{O},\mathcal{S},\mathcal{M}\rangle$ and $\mathcal{S}$-database D, one can build an OBDM specification $J' = \langle\mathcal{O},\mathcal{S},\mathcal{M}'\rangle$ such that $Mod_D(J) = Mod_D(J')$, where $\mathcal{M}'$ is obtained from $\mathcal{M}$ by adding such mapping assertions.

mapping followed by a LAV mapping over an intermediate alphabet, we can rewrite $q_{\mathcal{O}}$ into a UCQ over $\mathcal{S}$, denoted by $\mathsf{REW}_{\mathcal{M}}(q_{\mathcal{O}})$, such that $\mathsf{REW}_{\mathcal{M}}(q_{\mathcal{O}}) \equiv cert_{q_{\mathcal{O}},J}$, i.e., $\mathsf{REW}_{\mathcal{M}}(q_{\mathcal{O}})^D = cert^D_{q_{\mathcal{O}},J}$ for every $\mathcal{S}$-database D.[6]

We are now ready to provide the last preliminary notions and notations related to certain answers in our setting. Specifically, the *canonical structure of* $\mathcal{O}$ *with respect to* $\mathcal{M}$ *and* D, denoted $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$, is the (possibly infinite) set of atoms over $\mathcal{O}$ obtained by first chasing D with respect to $\mathcal{M}$, and then by chasing, possibly ad infinitum, the resulting set of atoms $\mathcal{M}(D)$ with respect to $\mathcal{O}$ as described in [22, Definition 8][7] but using the alphabet Var of variables whenever a new element is needed in the chase. By combining results of [53, Proposition 4.2] with results of [22, Theorem 29], it is well-known that, if D is consistent with J and $q_{\mathcal{O}}$ is a UCQ over $\mathcal{O}$, then, for every tuple of constants $\vec{c}$ in D, we have that $\vec{c} \in cert^D_{q_{\mathcal{O}},J}$ if and only if $\vec{c} \in q_{\mathcal{O}}^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$, i.e., if and only if $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)} \models q_{\mathcal{O}}(\vec{c})$. The following example illustrates the above recalled notion of canonical structure.

Example 2.6. Consider the OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ introduced in Example 1.1 and the $\mathcal{S}$-database D introduced in Example 2.5. The canonical structure of $\mathcal{O}$ with respect to $\mathcal{M}$ and D, $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$, is obtained by first computing the set $\mathcal{M}(D)$, which leads to the following:

$$\begin{aligned}\mathcal{M}(D) = \{&\mathsf{TeachesTo}(\mathsf{John}, \mathsf{Max}), \mathsf{Professor}(\mathsf{Julie}), \mathsf{RegisteredTo}(\mathsf{Alfred}, \mathsf{Sapienza}),\\ &\mathsf{RegisteredTo}(\mathsf{Jim}, \mathsf{Sapienza}), \mathsf{HasGuarantor}(\mathsf{Alfred}, \mathsf{Alfred}), \mathsf{HasGuarantor}(s, \mathsf{Gilda})\}\end{aligned}$$

where $s \in \mathsf{Var}$, and then by chasing $\mathcal{M}(D)$ with respect to $\mathcal{O}$, thus obtaining the set of atoms $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ that is identical to the model M of Example 2.5 except for the fact that it contains the variable s instead of the constant Steve. Also, if $q_{\mathcal{O}}$ is the query $q_{\mathcal{O}} = \{(x) \mid \mathsf{Professor}(x)\}$ introduced in Example 1.1, then $cert^D_{q_{\mathcal{O}},J} = q_{\mathcal{O}}^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}} = \{(\mathsf{John}), (\mathsf{Julie})\}$. △

Furthermore, given a UCQ $q_{\mathcal{O}}$ over $\mathcal{O}$, we denote by $\mathsf{PerfRef}_{q_{\mathcal{O}},\mathcal{O}}$ the UCQ over $\mathcal{O}$ computed by executing the algorithm $\mathsf{PerfectRef}$ described in [22] on $q_{\mathcal{O}}$ and $\mathcal{O}$, which, intuitively, encodes all inclusion assertions of $\mathcal{O}$, so that given an OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, $cert^D_{q_{\mathcal{O}},J} = cert^D_{\mathsf{PerfRef}_{q_{\mathcal{O}},\mathcal{O}},J'}$, for every $\mathcal{S}$-database D consistent with J, where J' is obtained from J be replacing $\mathcal{O}$ with an empty ontology over the same alphabet. Thus, if we denote by $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ the UCQ $\mathsf{REW}_{\mathcal{M}}(\mathsf{PerfRef}_{q_{\mathcal{O}},\mathcal{O}})$ over $\mathcal{S}$, by combining results of [22] with the definition of perfect J-rewriting, we obtain that if $q_{\mathcal{S}}$ is a perfect J-rewriting of $q_{\mathcal{O}}$, then $q^D_{\mathcal{S}} = \mathsf{PerfRef}^D_{q_{\mathcal{O}},J}$, for every $\mathcal{S}$-database D consistent with J. Note, in particular, that the previous claim holds only for $\mathcal{S}$-databases consistent with J, since $\mathsf{PerfectRef}$ ignores the disjointness assertions. Let us illustrates all this by an example.

Example 2.7. Consider the ontology $\mathcal{O}$ of the OBDM specification J introduced in Example 1.1 and the query $q_{\mathcal{O}} = \{(x) \mid \mathsf{Professor}(x)\}$. By applying the algorithm $\mathsf{PerfectRef}$ to $q_{\mathcal{O}}$ and $\mathcal{O}$ we obtain the following UCQ: $\mathsf{PerfRef}_{q_{\mathcal{O}},\mathcal{O}} = \{(x) \mid \mathsf{Professor}(x)\} \cup \{(x) \mid \exists y.\mathsf{TeachesTo}(x, y)\}$, and then $\mathsf{PerfRef}_{q_{\mathcal{O}},J} = \mathsf{REW}_{\mathcal{M}}(\mathsf{PerfRef}_{q_{\mathcal{O}},\mathcal{O}}) = \{(x) \mid s_2(x)\} \cup \{(x) \mid \exists y.s_1(x, y)\}$.

Note, however, that $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ is not a perfect J-rewriting of $q_{\mathcal{O}}$, since, clearly, for every $\mathcal{S}$-database D inconsistent with J, $\mathsf{PerfRef}^D_{q_{\mathcal{O}},J} \neq cert^D_{q_{\mathcal{O}},J}$. Let us consider, for example, the $\mathcal{S}$-database $D' = \{s_2(\mathsf{Julie}), s_3(\mathsf{Julie}), s_3(\mathsf{Jim})\}$. It is easy to verify that D' is inconsistent with J. Thus, $cert^{D'}_{q_{\mathcal{O}},J} = \{(c) \mid c \text{ occurs in } D', \text{ in } \mathcal{M}, \text{ or in } q_{\mathcal{O}}\} = \{(\mathsf{Julie}), (\mathsf{Jim}), (\mathsf{Sapienza})\}$. On the other hand, we have that $\mathsf{PerfRef}^{D'}_{q_{\mathcal{O}},J} = \{(\mathsf{Julie})\}$. △

The previous example highlights the impact of dealing with inconsistent $\mathcal{S}$-databases when using $\mathsf{PerfectRef}$ to compute certain answers. To face such an issue, one can resort to the notion of violation query. Specifically, the $\mathcal{O}$*-violation query*, denoted by $\mathcal{V}_{\mathcal{O}}$, is the boolean UCQ constituted by the disjunct $\{() \mid \exists y.\bot(y)\}$ and a set v of disjuncts defined as follows. The set v contains one disjunct of the form $\{() \mid \exists y.A_1(y) \wedge A_2(y)\}$ (respectively, $\{() \mid \exists y_1, y_2.A_1(y_1) \wedge R(y_1, y_2)\}$, $\{() \mid \exists y_1, y_2, y_3.R_1(y_1, y_2) \wedge R_2(y_1, y_3)\}$, and $\{() \mid \exists y_1, y_2.R_1(y_1, y_2) \wedge R_2(y_1, y_2)\}$) for each disjointness assertion $A_1 \sqsubseteq \neg A_2$ (respectively, $A_1 \sqsubseteq \neg \exists R$ or $\exists R \sqsubseteq \neg A_1$, $\exists R_1 \sqsubseteq \neg \exists R_2$, and $R_1 \sqsubseteq \neg R_2$), where an atom of the form $R(y, y')$ stands for either $P(y, y')$ if R denotes an atomic role P, or $P(y', y)$ if R denotes the inverse of an atomic role, i.e., $R = P^-$. From results of [3] easily follows that, given an OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and an $\mathcal{S}$-database D, D is consistent with J if and only if $\mathsf{PerfRef}^D_{\mathcal{V}_{\mathcal{O}},J} = \emptyset$.

Example 2.8. Consider the OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ introduced in Example 1.1. Since $\mathcal{O}$ contains only one disjointness assertion $\mathsf{Student} \sqsubseteq \neg\mathsf{Professor}$, the $\mathcal{O}$*-violation query* is the following:

$$\mathcal{V}_{\mathcal{O}} = \{() \mid \exists y.\bot(y)\} \cup \{() \mid \exists y.\mathsf{Student}(y) \wedge \mathsf{Professor}(y)\}$$

[6] Observe that, if $q_{\mathcal{O}}$ is a CQ over $\mathcal{O}$ of arity n having an atom with predicate symbol $\bot$ in its body, then $\mathsf{REW}_{\mathcal{M}}(q_{\mathcal{O}})$ is $\mathtt{false}/n$ for every GLAV mapping $\mathcal{M}$.

[7] In fact, [22, Definition 8] defines how to construct a canonical interpretation of a *DL-Lite*$_{\mathcal{R}}$ knowledge base. Once one treats an interpretation as the (possibly infinite) set of facts that are satisfied by the interpretation, it is immediate to apply such construction in our setting.

and $\mathsf{PerfRef}_{\mathcal{V}_{\mathcal{O}},J} = \{() \mid \exists y.s_3(y) \wedge s_2(y)\} \cup \{() \mid \exists y,z.s_3(y) \wedge s_1(y,z)\}$.

Now, consider again the database D' introduced in Example 2.7. One can easily verify that $\mathsf{PerfRef}^{D'}_{\mathcal{V}_{\mathcal{O}},J} = \{()\}$, thus confirming that D' is inconsistent with J. △

With all these notions and results in place, it is straightforward to show the following.

Proposition 2.1. *Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification, where $\mathcal{O}$ is a DL-Lite$_{\mathcal{R}}$ ontology and $\mathcal{M}$ is a GLAV mapping. If $q_{\mathcal{O}}$ is a UCQ over $\mathcal{O}$ of arity n, then the UCQ ($\mathsf{PerfRef}_{q_{\mathcal{O}},J} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J}$) over $\mathcal{S}$ is a perfect J-rewriting of $q_{\mathcal{O}}$, where $\mathcal{V}^n_{\mathcal{O}}$ denotes the UCQ over $\mathcal{O}$ of arity n containing a disjunct $\{(x_1,\ldots,x_n) \mid \exists\vec{y}.\phi(\vec{y}) \wedge \top(x_1)\ldots\wedge\top(x_n)\}$ for each disjunct in $\mathcal{V}_{\mathcal{O}}$ of the form $\{() \mid \exists\vec{y}.\phi(\vec{y})\}$.*

Example 2.9. Consider the OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ of Example 1.1. Also, recall the queries $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ and $\mathsf{PerfRef}_{\mathcal{V}_{\mathcal{O}},J}$ computed in Examples 2.7 and 2.8, respectively. According to the above proposition, a perfect J-rewriting of $q_{\mathcal{O}}$ is the query $q' = (\mathsf{PerfRef}_{q_{\mathcal{O}},J} \cup \mathsf{PerfRef}_{\mathcal{V}^1_{\mathcal{O}},J})$.

Thus, for every $\mathcal{S}$-database D, $cert^D_{q_{\mathcal{O}},J}$ can be computed by evaluating q' over D. Now, consider the query $q_{\mathcal{S}}$ of Example 1.1. The fact that $q_{\mathcal{S}}$ coincides with $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ confirms the intuition that $q_{\mathcal{O}}$ is the query over $\mathcal{O}$ that best characterizes $q_{\mathcal{S}}$ with respect to J. △

3. Framework

In the rest of this paper, we implicitly use $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ to denote an OBDM specification, $q_{\mathcal{S}}$ to denote a query over the schema $\mathcal{S}$, and $q_{\mathcal{O}}$ to denote a query over the ontology $\mathcal{O}$.

In the context of abstraction, given $q_{\mathcal{S}}$, we aim at finding the query over $\mathcal{O}$ that precisely characterizes $q_{\mathcal{S}}$ w.r.t. the OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$. Since the evaluation of queries over $\mathcal{O}$ is based on certain answers, this means that we aim at finding a query over $\mathcal{O}$ whose certain answers w.r.t. J and D exactly capture the answers of $q_{\mathcal{S}}$ over D, for every $\mathcal{S}$-database D. Therefore, we are naturally led to the notion of perfect abstraction.

Definition 3.1. $q_{\mathcal{O}}$ is a *perfect J-abstraction of $q_{\mathcal{S}}$* if for every $\mathcal{S}$-database D, $Mod_D(J) \neq \emptyset$ implies $cert^D_{q_{\mathcal{O}},J} = q^D_{\mathcal{S}}$. If in addition $q_{\mathcal{O}} \in \mathcal{L}_{\mathcal{O}}$ for a query language $\mathcal{L}_{\mathcal{O}}$, then we say that $q_{\mathcal{O}}$ is an *$\mathcal{L}_{\mathcal{O}}$-perfect J-abstraction of $q_{\mathcal{S}}$*.

The following proposition states that perfect abstractions are always unique, up to equivalence w.r.t. the underlying OBDM specification J.

Proposition 3.1. *If q_1 and q_2 are perfect J-abstractions of $q_{\mathcal{S}}$, then they are equivalent w.r.t. J.*

Proof. Following Definition 3.1, since q_1 and q_2 are perfect J-abstractions of $q_{\mathcal{S}}$, we have that $cert^D_{q_1,J} = q^D_{\mathcal{S}} = cert^D_{q_2,J}$ for all $\mathcal{S}$-databases D consistent with J. For all the $\mathcal{S}$-databases D that are not consistent with J, however, by definition of certain answers, we have that $cert^D_{q_1,J} = cert^D_{q_2,J}$ as well. So, $cert^D_{q_1,J} = cert^D_{q_2,J}$ for all $\mathcal{S}$-databases D, i.e., $cert_{q_1,J} \equiv cert_{q_2,J}$. □

The above notion is similar, but not equivalent, to the notion of *realization* in [15]. Indeed, while the latter sanctions that $q^D_{\mathcal{S}} = cert^D_{q_{\mathcal{O}},J}$ for *all* $\mathcal{S}$-databases D, in Definition 3.1 the condition is limited only to the $\mathcal{S}$-databases D that are consistent with J. Obviously, every query that is a realization of a source query $q_{\mathcal{S}}$ is also an abstraction of $q_{\mathcal{S}}$. However, the converse is not necessarily true, as shown in the following example.

Example 3.1. Consider the OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, and queries $q_{\mathcal{S}} = \{(x) \mid s_2(x)\} \cup \{(x) \mid \exists y.s_1(x,y)\}$ and $q_{\mathcal{O}} = \{(x) \mid \mathsf{Professor}(x)\}$ of Example 1.1.

Recall the $\mathcal{S}$-database $D' = \{s_2(\mathsf{Julie}), s_3(\mathsf{Julie}), s_3(\mathsf{Jim})\}$ considered in Example 2.7. We have that $q^{D'}_{\mathcal{S}} = \{(\mathsf{Julie})\}$ while $cert^{D'}_{q_{\mathcal{O}},J} = \{(\mathsf{Julie}), (\mathsf{Jim}), (\mathsf{Sapienza})\}$ because D' is inconsistent with J. It follows that, according to the semantics proposed in [15] (which ranges over all $\mathcal{S}$-databases), $q_{\mathcal{O}}$ is not a *realization* of $q_{\mathcal{S}}$ in J, whereas, according to Definition 3.1, since $q^D_{\mathcal{S}} = cert^D_{q_{\mathcal{O}},J}$ holds for every $\mathcal{S}$-database D consistent with J, by definition $q_{\mathcal{O}}$ is a CQ-perfect J-abstraction of $q_{\mathcal{S}}$. △

Notice, however, that in the above example, the algorithm proposed in [15] for computing the realization of $q_{\mathcal{S}}$ in J returns exactly $q_{\mathcal{O}}$, and is therefore incorrect. More generally, it can be shown that such algorithm provides an incorrect result whenever the underline OBDM specification may give rise to an inconsistent OBDM system. Indeed, we observe that, contrarily to the results reported in the present paper, the results in [15] apply only to DLs where inconsistencies cannot arise.

As implicitly noted in [12,15,1] and illustrated more formally in the next example, perfect abstractions may not exist, even in trivial cases.

Example 3.2. Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \{\ \exists\mathsf{WorksFor} \sqsubseteq \mathsf{Worker}, \mathsf{MathStudent} \sqsubseteq \mathsf{Student}\ \}$
- $\mathcal{S} = \{\ s_1, s_2, s_3, s_4, s_5\ \}$
- $\mathcal{M} = \{\ m_1, m_2, m_3, m_4, m_5, m_6\ \}$, where:

$$\begin{array}{lrcl}
m_1: & s_1(x) & \rightarrow & \mathsf{Worker}(x)\\
m_2: & s_1(x) & \rightarrow & \mathsf{Student}(x)\\
m_3: & s_2(x_1, x_2) & \rightarrow & \mathsf{WorksFor}(x_1, x_2)\\
m_4: & s_3(x) & \rightarrow & \mathsf{MathStudent}(x)\\
m_5: & s_1(x) \wedge s_4(x) & \rightarrow & \mathsf{Engineer}(x)\\
m_6: & s_1(x_1) \wedge s_5(x_1, x_2) & \rightarrow & \mathsf{PlaysSport}(x_1, x_2)
\end{array}$$

Consider query $q_{\mathcal{S}} = \{(x) \mid s_1(x)\}$ over the source schema $\mathcal{S}$. By inspecting the mapping $\mathcal{M}$ and the ontology $\mathcal{O}$ one can see that, since the certain answers of $q^1_{\mathcal{O}} = \{(x) \mid \mathsf{Worker}(x)\}$ include also the values stored in the projection on the first component of s_2, and since the certain answers of $q^2_{\mathcal{O}} = \{(x) \mid \mathsf{Student}(x)\}$ include also the values stored in s_3, both queries are too general for exactly characterizing $q_{\mathcal{S}}$. On the other hand, queries $q^3_{\mathcal{O}} = \{(x) \mid \mathsf{Engineer}(x)\}$ and $q^4_{\mathcal{O}} = \{(x) \mid \mathsf{PlaysSport}(x, y)\}$ are too specific, and therefore a perfect J-abstraction of $q_{\mathcal{S}}$ does not exist. To formally prove this latter statement, consider the $\mathcal{S}$-databases $D_1 = \{s_1(a), s_2(a, b), s_3(a)\}$ and $D_2 = \{s_2(a, b), s_3(a)\}$. One can verify that, while $q^{D_1}_{\mathcal{S}} = \{(a)\}$ and $q^{D_2}_{\mathcal{S}} = \emptyset$, we have that $Mod_{D_1}(J) = Mod_{D_2}(J)$ implying that each query $q_{\mathcal{O}}$ over $\mathcal{O}$ must be such that $cert^{D_1}_{q_{\mathcal{O}},J} = cert^{D_2}_{q_{\mathcal{O}},J}$, and so, since $Mod_{D_1}(J) = Mod_{D_2}(J) \neq \emptyset$, a perfect J-abstraction of $q_{\mathcal{S}}$ cannot exist. △

In order to cope with the situations illustrated in the above example, we introduce the notions of sound and complete abstractions, which, intuitively, provide sound and complete approximations of perfect abstractions, respectively.

Definition 3.2. $q_{\mathcal{O}}$ is a *sound* (respectively, *complete*) J-abstraction of $q_{\mathcal{S}}$ if for every $\mathcal{S}$-database D, $Mod_D(J) \neq \emptyset$ implies $cert^D_{q_{\mathcal{O}},J} \subseteq q^D_{\mathcal{S}}$ (respectively, $q^D_{\mathcal{S}} \subseteq cert^D_{q_{\mathcal{O}},J}$).

Example 3.3. Refer to Example 3.2. Note that $q^1_{\mathcal{O}}$ and $q^2_{\mathcal{O}}$ are complete J-abstractions of $q_{\mathcal{S}}$, whereas $q^3_{\mathcal{O}}$ and $q^4_{\mathcal{O}}$ are sound J-abstractions of $q_{\mathcal{S}}$. △

Obviously, $q_{\mathcal{O}}$ is a perfect J-abstraction of $q_{\mathcal{S}}$ if and only if $q_{\mathcal{O}}$ is both a sound, and a complete J-abstraction of $q_{\mathcal{S}}$. The following theorem discusses relevant relationships between the notions of J-abstractions introduced here and the notions of rewritings studied in OBDM.

Proposition 3.2. *The following holds:*

1. *$q_{\mathcal{O}}$ is a complete J-abstraction of $q_{\mathcal{S}}$ if and only if $q_{\mathcal{S}}$ is a sound J-rewriting of $q_{\mathcal{O}}$.*
2. *If $q_{\mathcal{S}}$ is a perfect J-rewriting of $q_{\mathcal{O}}$, then $q_{\mathcal{O}}$ is a perfect J-abstraction of $q_{\mathcal{S}}$. The converse does not necessarily hold.*

Proof. As for 1, by definition $q_{\mathcal{S}}$ is a sound J-rewriting of $q_{\mathcal{O}}$ if and only if $q^D_{\mathcal{S}} \subseteq cert^D_{q_{\mathcal{O}},J}$ for every $\mathcal{S}$-database D. Since in the case of D inconsistent with J the above inclusion trivially holds, this is equivalent to the condition $q^D_{\mathcal{S}} \subseteq cert^D_{q_{\mathcal{O}},J}$ for every $\mathcal{S}$-database D consistent with J, which is exactly the definition of $q_{\mathcal{O}}$ being a complete J-abstraction of $q_{\mathcal{S}}$.

As for the sufficient condition of 2, by definition $q_{\mathcal{S}}$ is a perfect J-rewriting of $q_{\mathcal{O}}$ if and only if $q^D_{\mathcal{S}} = cert^D_{q_{\mathcal{O}},J}$ for every $\mathcal{S}$-database D, which obviously implies that $q_{\mathcal{O}}$ is a perfect J-abstraction of $q_{\mathcal{S}}$. To show that the converse does not necessarily hold, consider J, $q_{\mathcal{S}}$, and $q_{\mathcal{O}}$ as described in Example 1.1, and notice that, while $q_{\mathcal{O}}$ is a perfect J-abstraction of $q_{\mathcal{S}}$, $q_{\mathcal{S}}$ is not a perfect J-rewriting of $q_{\mathcal{O}}$. □

Notice, however, that the converse statement of point 2 of the above proposition becomes in fact true in the case of OBDM specifications based on ontology languages not able to express inconsistencies, such as the DL *DL-Lite*$_{\mathsf{RDFS}}$ and the DLs $\mathcal{EL}$ and $\mathcal{ELHI}$ [28] considered in [15].

It is easy to see that different sound or complete abstractions of $q_{\mathcal{S}}$ may exist, and therefore it is reasonable to look for the "best" approximations of $q_{\mathcal{S}}$, at least relative to a certain class of queries.

Definition 3.3. $q_{\mathcal{O}} \in \mathcal{L}$ is an *$\mathcal{L}$-maximally sound* (respectively, *$\mathcal{L}$-minimally complete*) J-abstraction of $q_{\mathcal{S}}$ if $q_{\mathcal{O}}$ is a sound (respectively, complete) J-abstraction of $q_{\mathcal{S}}$ and there exists no $q' \in \mathcal{L}$ such that (*i*) q' is a sound (respectively, complete) J-

abstraction of $q_{\mathcal{S}}$, (*ii*) $cert_{q_{\mathcal{O}},J} \sqsubseteq cert_{q',J}$ (respectively, $cert_{q',J} \sqsubseteq cert_{q_{\mathcal{O}},J}$), and (*iii*) $cert^D_{q_{\mathcal{O}},J} \subset cert^D_{q',J}$ (respectively, $cert^D_{q',J} \subset cert^D_{q_{\mathcal{O}},J}$) for an $\mathcal{S}$-database D.

Example 3.4. We refer again to Example 3.2. Observe that neither $q^1_{\mathcal{O}}$ nor $q^2_{\mathcal{O}}$ are CQ-minimally complete J-abstractions of $q_{\mathcal{S}}$. Indeed, one can verify that the CQ $q^5_{\mathcal{O}} = \{(x) \mid \mathsf{Worker}(x) \wedge \mathsf{Student}(x)\}$ is a UCQ-minimally complete J-abstraction of $q_{\mathcal{S}}$. As for $q^3_{\mathcal{O}}$ and $q^4_{\mathcal{O}}$, it is easy to see that they are both CQ-maximally sound J-abstractions of $q_{\mathcal{S}}$, but neither of them is a UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$. Indeed, one can verify that the UCQ $q^6_{\mathcal{O}} = q^3_{\mathcal{O}} \cup q^4_{\mathcal{O}}$ is a UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$. △

As we will see in a next proposition, there are types of OBDM specifications and query languages $\mathcal{L}_{\mathcal{O}}$ for which it is always the case that if there exists an $\mathcal{L}_{\mathcal{O}}$-maximally sound (respectively, $\mathcal{L}_{\mathcal{O}}$-minimally complete) J-abstraction of a query $q_{\mathcal{S}}$, then it is unique, up to equivalence w.r.t. J. In these cases, it is reasonable to talk about *the* $\mathcal{L}_{\mathcal{O}}$-maximally sound (respectively, $\mathcal{L}_{\mathcal{O}}$-minimally complete) J-abstraction of $q_{\mathcal{S}}$.

Definition 3.4. $q_{\mathcal{O}} \in \mathcal{L}$ is the $\mathcal{L}$-maximally sound (respectively, $\mathcal{L}$-minimally complete) J-abstraction of $q_{\mathcal{S}}$ if (*i*) $q_{\mathcal{O}}$ is a sound (respectively, complete) J-abstraction of $q_{\mathcal{S}}$, and (*ii*) every $q' \in \mathcal{L}$ that is a sound (respectively, complete) J-abstraction of $q_{\mathcal{S}}$ is such that $cert_{q',J} \sqsubseteq cert_{q_{\mathcal{O}},J}$ (respectively, $cert_{q_{\mathcal{O}},J} \sqsubseteq cert_{q',J}$).

Example 3.5. Refer again to Example 3.2, and consider also the queries $q^5_{\mathcal{O}}$ and $q^6_{\mathcal{O}}$ defined in Example 3.4. It can be shown that $q^5_{\mathcal{O}}$ (respectively, $q^6_{\mathcal{O}}$) is the UCQ-minimally complete (respectively, UCQ-maximally sound) J-abstraction of $q_{\mathcal{S}}$. Furthermore, observe that: (*i*) since $q^5_{\mathcal{O}}$ is a CQ, it is also the CQ-minimally complete J-abstraction of $q_{\mathcal{S}}$, and (*ii*) since both $q^3_{\mathcal{O}}$ and $q^4_{\mathcal{O}}$ are CQ-maximally sound J-abstractions of $q_{\mathcal{S}}$ and they are not equivalent w.r.t. J, we conclude that the CQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ does not exist. △

Given the general framework presented so far, it is natural to consider the following two basic computational problems, for classes $\mathcal{L}_{\mathcal{S}}$ and $\mathcal{L}_{\mathcal{O}}$ of queries over the source schema $\mathcal{S}$ and over the ontology $\mathcal{O}$, respectively:

- *Verification*: given $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, $q_{\mathcal{S}} \in \mathcal{L}_{\mathcal{S}}$ over $\mathcal{S}$, and $q_{\mathcal{O}} \in \mathcal{L}_{\mathcal{O}}$ over $\mathcal{O}$ of the same arity of $q_{\mathcal{S}}$, verify whether $q_{\mathcal{O}}$ is a perfect (respectively, sound, complete) J-abstraction of $q_{\mathcal{S}}$.
- *Computation*: given $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, and $q_{\mathcal{S}} \in \mathcal{L}_{\mathcal{S}}$ over $\mathcal{S}$, compute any $\mathcal{L}_{\mathcal{O}}$-perfect (respectively, $\mathcal{L}_{\mathcal{O}}$-maximally sound, $\mathcal{L}_{\mathcal{O}}$-minimally complete) J-abstraction of $q_{\mathcal{S}}$, if it exists.

In what follows, if not otherwise stated, we silently assume to deal with the following scenario:

- $\mathcal{O}$ is expressed in *DL-Lite*$_{\mathcal{R}}$,
- $\mathcal{M}$ is a GLAV mapping,
- both $\mathcal{L}_{\mathcal{O}}$ and $\mathcal{L}_{\mathcal{S}}$ denote the class of UCQs.

Furthermore, to ease the presentation, from now on whenever we refer to a UCQ $q_{\mathcal{S}}$ of arity n over a schema $\mathcal{S}$, we implicitly assume that all the disjuncts of $q_{\mathcal{S}}$ are different from the special CQ $\texttt{false}/n$. Clearly, all the results can be generalized in a straightforward manner to include also $\texttt{false}/n$ as possible disjuncts occurring in input UCQs over source schemas.

Interestingly, in this scenario, we have the following result.

Proposition 3.3. *If q_1 and q_2 are UCQ-maximally sound (respectively, UCQ-minimally complete) J-abstractions of $q_{\mathcal{S}}$, then they are equivalent w.r.t. J.*

Proof. We first address the case of UCQ-maximally sound, and then the case of UCQ-minimally complete.

Assume that q_1 and q_2 are UCQ-maximally sound J-abstractions of $q_{\mathcal{S}}$ and suppose, for the sake of contradiction, that they are not equivalent w.r.t. J. This implies the existence of an $\mathcal{S}$-database D and a tuple of constants $\vec{c}$ such that either $\vec{c} \notin cert^D_{q_1,J}$ and $\vec{c} \in cert^D_{q_2,J}$, or $\vec{c} \in cert^D_{q_1,J}$ and $\vec{c} \notin cert^D_{q_2,J}$. Let us assume, w.l.o.g., that $\vec{c} \notin cert^D_{q_1,J}$ and $\vec{c} \in cert^D_{q_2,J}$. But then, it can be readily seen that the UCQ $Q = q_1 \cup q_2$ is such that (*i*) since both q_1 and q_2 are sound J-abstractions of $q_{\mathcal{S}}$, Q is a sound J-abstraction of $q_{\mathcal{S}}$ as well, (*ii*) $cert_{q_1,J} \sqsubseteq cert_{Q,J}$, and (*iii*) the $\mathcal{S}$-database D is such that $cert^D_{q_1,J} \subset cert^D_{Q,J}$. Obviously, this contradicts the fact that q_1 is a UCQ-maximally sound J-abstractions of $q_{\mathcal{S}}$.

Assume now that q_1 and q_2 are UCQ-minimally complete J-abstractions of $q_{\mathcal{S}}$ and suppose, for the sake of contradiction, that they are not equivalent w.r.t. J. Following the same line of reasoning as above, we can assume there exists an $\mathcal{S}$-database D and a tuple of constants $\vec{c}$ such that $\vec{c} \in cert^D_{q_1,J}$ and $\vec{c} \notin cert^D_{q_2,J}$. But then, consider the query Q such that $cert^{D'}_{Q,J} = cert^{D'}_{q_1,J} \bigcap cert^{D'}_{q_2,J}$ for every $\mathcal{S}$-database D' and OBDM specification J. Obviously, since q_1 and q_2 are UCQs, Q

always exists and can be expressed as the UCQ $\bigcup_{q' \in q_1, q'' \in q_2} q' \wedge q''$. It can be readily seen that (*i*) since q_1 and q_2 are complete J-abstractions of $q_{\mathcal{S}}$, Q is a complete J-abstraction of $q_{\mathcal{S}}$ as well, (*ii*) $cert_{Q,J} \sqsubseteq cert_{q_1,J}$, and (*iii*) the $\mathcal{S}$-database D is such that $cert^D_{Q,J} \subset cert^D_{q_1,J}$. Obviously, this contradicts the fact that q_1 is a UCQ-minimally complete J-abstractions of $q_{\mathcal{S}}$. □

4. Complete abstractions

In this section, we study both the verification and the computation problem for complete abstractions.

4.1. Verification

Suppose we want to check whether $q_{\mathcal{O}}$ is a complete J-abstraction of $q_{\mathcal{S}}$. Obviously, if $q_{\mathcal{S}}$ is contained in $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$, then for every $\mathcal{S}$-database D consistent with J, we have that $q^D_{\mathcal{S}} \subseteq cert^D_{q_{\mathcal{O}},J}$ and therefore the answer is positive. If $q_{\mathcal{S}}$ is not contained in $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$, however, it might be the case that $q_{\mathcal{O}}$ is still a complete J-abstraction of $q_{\mathcal{S}}$, in particular in the case where the non-emptiness of the answers of $q_{\mathcal{S}}$ over D reveals the presence of inconsistencies. From this observation, we derive the following characterization.

Lemma 4.1. *$q_{\mathcal{O}}$ is a complete J-abstraction of $q_{\mathcal{S}}$ if and only if $q_{\mathcal{S}} \sqsubseteq (\mathsf{PerfRef}_{q_{\mathcal{O}},J} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J})$, where $n = ar(q_{\mathcal{O}}) = ar(q_{\mathcal{S}})$.*

Proof. The thesis immediately follows from Proposition 2.1, which sanctions that $cert_{q_{\mathcal{O}},J} \equiv (\mathsf{PerfRef}_{q_{\mathcal{O}},J} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J})$, and from the definition of complete J-abstractions. □

The following theorem characterizes the computational complexity of the verification problem for complete abstractions.

Theorem 4.1. *The verification problem for complete abstractions is NP-complete.*

Proof. As for the upper bound, by virtue of Lemma 4.1, it is sufficient to show how to check the containment $q_{\mathcal{S}} \sqsubseteq (\mathsf{PerfRef}_{q_{\mathcal{O}},J} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J})$ in NP, where $n = ar(q_{\mathcal{O}})$. Given a disjunct q of $q_{\mathcal{S}}$, we guess (*i*) a disjunct $\overline{q_{\mathcal{O}}}$ in $q_{\mathcal{O}}$ or in $\mathcal{V}^n_{\mathcal{O}}$, (*ii*) a query q' over $\mathcal{O}$ with the same arity as $\overline{q_{\mathcal{O}}}$ and size at most the maximum between $\sigma(\overline{q_{\mathcal{O}}})$ and $\sigma(\mathcal{V}^n_{\mathcal{O}})$, (*iii*) a sequence $\rho_{\mathcal{O}}$ of ontology assertions in $\mathcal{O}$, (*iv*) a set $\rho_{\mathcal{M}}$ of $\sigma(q')$ pairs $\langle m_i, \chi_i \rangle$, where $m_i = \exists \vec{y}.\phi_{\mathcal{S}}(\vec{x}, \vec{y}) \rightarrow \exists \vec{z}.\psi_{\mathcal{O}}(\vec{x}, \vec{z})$ is a mapping assertion in $\mathcal{M}$, and χ_i is a function from the variables of $\phi_{\mathcal{S}}$ to the variables of q. Note in particular that by guessing $\rho_{\mathcal{O}}$, we guess the sequence of ontology assertions that are used to rewrite $q_{\mathcal{O}}$ by running $\mathsf{PerfectRef}$ on $q_{\mathcal{O}}$ and $\mathcal{O}$. Hence, since, by construction, at each rewriting step, $\mathsf{PerfectRef}$ uses one ontology assertion to generate a query whose size is less or equal to that of $q_{\mathcal{O}}$ and that comprises terms using only the variables and constants occurring in $q_{\mathcal{O}}$ plus the symbol _, and since $\mathsf{PerfectRef}$ terminates when no new query can be generated [22], $\rho_{\mathcal{O}}$ comprises at most $\sigma(\mathcal{O}) \times \sigma(\overline{q_{\mathcal{O}}})$ ontology assertions. Then, we check in polynomial time (*i*) whether q' is obtained by rewriting $\overline{q_{\mathcal{O}}}$ using the sequence $\rho_{\mathcal{O}}$, (*ii*) whether each χ_i is a homomorphism, and (*iii*) whether q' is also obtained by initializing γ to the empty set, and then adding to γ the atoms $\psi_{\mathcal{O}}(\vec{c_i}, \vec{z})$ for every pair $\langle m_i, \chi_i \rangle$, where $\vec{c_i}$ is the projection to $\vec{x}$ of the image of χ_i. It is not difficult to see that the containment holds if and only if for every disjunct q of $q_{\mathcal{S}}$, for the above mentioned guess, all such check returns true.

As for the lower bound, the proof of NP-hardness is by a LogSpace reduction from the 3-colourability problem, which is NP-complete [56]. 3-colourability is the problem of deciding, given an undirected graph $G = (V, E)$ with no self-loops, whether G is 3-colourable, i.e., whether there exists a function $f : V \rightarrow \{R, G, B\}$ such that $f(y_i) \neq f(y_j)$ for each $(y_i, y_j) \in E$. Given $G = (V, E)$ with $V = (y_1, y_2, \ldots, y_n)$, we define:

- the OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ as follows: the ontology $\mathcal{O}$ comprises the atomic role P and the atomic concepts R, G, B, and has no assertions, the schema $\mathcal{S}$ contains a binary predicate E and three unary predicates s_R, s_G, and s_B, and the mapping $\mathcal{M}$ is composed by the mapping assertions $E(x_1, x_2) \rightarrow P(x_1, x_2)$, $s_R(x) \rightarrow R(x)$, $s_G(x) \rightarrow G(x)$, and $s_B(x) \rightarrow B(x)$.
- the boolean CQ $q_{\mathcal{O}}$ over $\mathcal{O}$ as follows:

$$q_{\mathcal{O}} = \{(x_R, x_G, x_B) \mid \exists y_1, \ldots, y_n . R(x_R) \wedge G(x_G) \wedge B(x_B) \wedge \bigwedge_{(y_i, y_j) \in E} (P(y_i, y_j) \wedge P(y_j, y_i))\}.$$

- the boolean CQJFE $q_{\mathcal{S}}$ over $\mathcal{S}$ as follows: $q_{\mathcal{S}} = \{(x_R, x_G, x_B) \mid s_R(x_R) \wedge s_G(x_G) \wedge s_B(x_B) \wedge E(x_R, x_G) \wedge E(x_G, x_R) \wedge E(x_R, x_B) \wedge E(x_B, x_R) \wedge E(x_B, x_G) \wedge E(x_G, x_B)\}$.

Observe that, while $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and $q_{\mathcal{S}}$ do not depend on the input of the 3-colourability problem (i.e., $G = (V, E)$), $q_{\mathcal{O}}$ can be constructed in LogSpace from it. We now show that G is 3-colourable if and only if $q_{\mathcal{O}}$ is a complete J-abstraction of $q_{\mathcal{S}}$. To begin observe that $\mathcal{V}_{\mathcal{O}} = \{() \mid \exists y . \bot(y)\}$, and hence, for each $\mathcal{S}$-database D, we have that $cert^D_{q_{\mathcal{O}},J} = \mathsf{PerfRef}^D_{q_{\mathcal{O}},J}$, where $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ is the following CQ over $\mathcal{S}$:

Algorithm MinimallyComplete.

```
Input: OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$; UCQ $q_{\mathcal{S}} = q^1_{\mathcal{S}} \cup \ldots \cup q^n_{\mathcal{S}}$ over $\mathcal{S}$, where $q^i_{\mathcal{S}} = \{\vec{t_i} \mid \exists \vec{y}_i.\phi_i(\vec{x_i}, \vec{y_i})\}$ for each $i \in [1, n]$
Output: UCQ $q_{\mathcal{O}}$ over $\mathcal{O}$
1: $q_{\mathcal{O}} := \{\vec{t_1} \mid \exists \vec{\mathcal{Y}}_1.\mathcal{M}(q^1_{\mathcal{S}})) \wedge \top(\vec{x_1})\} \cup \ldots \cup \{\vec{t_n} \mid \exists \vec{\mathcal{Y}}_n.\mathcal{M}(q^n_{\mathcal{S}}) \wedge \top(\vec{x_n})\}$, where $\vec{\mathcal{Y}}_i$ includes the set of existential variables of $q^i_{\mathcal{S}}$ occurring in $\mathcal{M}(q_{\mathcal{S}})$ plus the fresh existential variables introduced by $\mathcal{M}(q_{\mathcal{S}})$, for each $i \in [1, n]$
2: return $q_{\mathcal{O}}$
```

$$\{(x_R, x_G, x_B) \mid \exists y_1, \ldots, y_n.s_R(x_R) \wedge s_G(x_G) \wedge s_B(x_B) \wedge \bigwedge_{(y_i, y_j) \in E} (E(y_i, y_j) \wedge E(y_j, y_i))\}.$$

"**Only-if part:**" Suppose G is 3-colourable, that is, there exists a function $f : V \rightarrow \{R, G, B\}$ such that $f(y_i) \neq f(y_j)$ for each $(y_i, y_j) \in E$. But then, consider the function h from the variables of $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ to the variables of $q_{\mathcal{S}}$ such that $h(x_R) = x_R$, $h(x_G) = x_G$, $h(x_B) = x_B$, and, for each $i = 1, \ldots, n$:

$$h(y_i) = \begin{cases} x_R, & \text{if } f(y_i) = R, \\ x_G, & \text{if } f(y_i) = G, \\ x_B, & \text{if } f(y_i) = B. \end{cases}$$

It can be readily seen that h is a homomorphism from $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ to $q_{\mathcal{S}}$. It follows that $q_{\mathcal{S}} \sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}},J}$ which, due to Lemma 4.1, implies that $q_{\mathcal{O}}$ is a complete J-abstraction of $q_{\mathcal{S}}$, as required.

"**If part:**" Suppose G is not 3-colourable, i.e., every function $f : V \rightarrow \{R, G, B\}$ is such that $f(y_i) = f(y_j)$ for some $(y_i, y_j) \in E$. This implies that every function from the variables of $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ to the variables of $q_{\mathcal{S}}$ is such that for at least one pair y_i, y_j such that $P(y_i, y_j)$ we have that $h(y_i) = h(y_j) = x_R$, or $h(y_i) = h(y_j) = x_G$, or $h(y_i) = h(y_j) = x_B$, and therefore is not a homomorphism from $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ to $q_{\mathcal{S}}$. It follows that $q_{\mathcal{S}} \not\sqsubseteq (\mathsf{PerfRef}_{q_{\mathcal{O}},J} \cup \mathsf{PerfRef}_{\mathcal{V}^3_{\mathcal{O}},J})$ which, due to Lemma 4.1, implies that $q_{\mathcal{O}}$ is not a complete J-abstraction of $q_{\mathcal{S}}$, as required. □

Note that the use of (x_R, x_G, x_B) as target list of $q_{\mathcal{S}}$ in the above proof aims at showing that the result holds even when $q_{\mathcal{S}}$ has no join on existential variables. More precisely, it follows from the proof that NP-hardness already holds when $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ is fixed (i.e., it does not depend on the input of the reduction) with $\mathcal{O}$ containing no assertions and $\mathcal{M}$ being both a pure GAV mapping and a LAV mapping, $q_{\mathcal{S}}$ is a fixed CQJFE, and $q_{\mathcal{O}}$ is a CQ. Moroever, from the above proof, we can derive the following corollary, which will be useful in the following.

Corollary 4.1. *If q_1 is a CQJFE and q_2 is a CQ, then the problem of checking whether $q_1 \sqsubseteq q_2$ is NP-complete.*

4.2. Computation

We now present the algorithm MinimallyComplete for computing UCQ-minimally complete abstractions. In the algorithm, for a tuple $\vec{x} = (x_1, \ldots, x_m)$ of variables, $\top(\vec{x})$ denotes a shortcut for $\top(x_1) \wedge \ldots \wedge \top(x_m)$.

Informally, for each disjunct $q^i_{\mathcal{S}}$ of $q_{\mathcal{S}}$, the algorithm obtains a CQ by simply chasing the set of atoms $q^i_{\mathcal{S}}$ with respect to $\mathcal{M}$, using $\top$ to bind those distinguished variables that do not occur in $\mathcal{M}(q^i_{\mathcal{S}})$. Then, it obtains the output query $q_{\mathcal{O}}$ as the union of all the CQs obtained in such a way.

Example 4.1. Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \emptyset$
- $\mathcal{S} = \{\ s_1, s_2, s_3\ \}$
- $\mathcal{M} = \{\ m_1, m_2, m_3, m_4\ \}$, where:

$$\begin{aligned} m_1: & \quad s_1(x) & \rightarrow & \quad \exists z.P_1(x, z) \wedge A_1(z), \\ m_2: & \quad \exists y.s_2(x_1, y) \wedge s_2(y, x_2) & \rightarrow & \quad P_2(x_1, x_2), \\ m_3: & \quad \exists y.s_1(c_1) \wedge s_3(x, y) & \rightarrow & \quad P_3(x, c_2), \\ m_4: & \quad \exists y.s_3(x_1, x_2) \wedge s_2(x_2, y) & \rightarrow & \quad P_4(x_1, x_2), \end{aligned}$$

and $q_{\mathcal{S}}$ be the UCQ $\{(x_1, x_2) \mid \exists y_1, y_2.s_1(x_1) \wedge s_2(x_1, y_1) \wedge s_2(y_2, x_2)\} \cup \{(x_1, c_3) \mid \exists y_1, y_2.s_1(c_1) \wedge s_3(x_1, y_1) \wedge s_2(y_1, y_2)\}$, where c_1, c_2, c_3 are constants.

One can verify that $\mathsf{MinimallyComplete}(J, q_{\mathcal{S}})$ returns the UCQ $q_{\mathcal{O}} = \{(x_1, x_2) \mid \exists y_3.P_1(x_1, y_3) \wedge A_1(y_3)\} \cup \{(x_1, c_3) \mid \exists y_1, y_3.P_1(c_1, y_3) \wedge A_1(y_3) \wedge P_3(x_1, c_2) \wedge P_4(x_1, y_1)\}$, which corresponds to the UCQ-minimally complete J-abstraction of $q_{\mathcal{S}}$. △

The following theorem establishes termination and correctness of the MinimallyComplete algorithm.

Theorem 4.2. *$\mathsf{MinimallyComplete}(J, q_{\mathcal{S}})$ terminates and returns the UCQ-minimally complete J-abstraction of $q_{\mathcal{S}}$.*

Proof. Termination of the algorithm easily follows from the termination of the chase of a set of atoms with respect to a GLAV mapping, or, equivalently, with respect to a set of source-to-target tuple-generating dependencies [53].

As for correctness of the algorithm, we first show that the computed $q_{\mathcal{O}} = q^1_{\mathcal{O}} \cup \ldots \cup q^n_{\mathcal{O}}$ is a complete J-abstraction of $q_{\mathcal{S}}$. Since for every $i \in [1, n]$, $q^i_{\mathcal{O}} = \{\vec{t_i} \mid \exists \vec{y}_i.\mathcal{M}(q^i_{\mathcal{S}}) \wedge \top(\vec{x}_i)\}$, by construction we have that the CQ $q^i_{\mathcal{S}}$ is contained in a disjunct of $\mathsf{REW}_{\mathcal{M}}(q^i_{\mathcal{O}})$. Thus, $q^i_{\mathcal{S}} \sqsubseteq \mathsf{PerfRef}_{q^i_{\mathcal{O}},J}$ holds for every $i \in [1, n]$. It follows that $q_{\mathcal{S}} \sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}},J}$ which, due to Lemma 4.1, implies that $q_{\mathcal{O}}$ is a complete J-abstraction of $q_{\mathcal{S}}$.

We now show that every UCQ $q'_{\mathcal{O}}$ that is a complete J-abstraction of $q_{\mathcal{S}}$ is such that $cert_{q_{\mathcal{O}},J} \sqsubseteq cert_{q'_{\mathcal{O}},J}$, thus showing that $q_{\mathcal{O}}$ is actually the UCQ-minimally complete J-abstraction of $q_{\mathcal{S}}$ (cf. Definition 3.4). We do so by contradiction, i.e., by proving that every UCQ $q'_{\mathcal{O}}$ such that $cert_{q_{\mathcal{O}},J} \not\sqsubseteq cert_{q'_{\mathcal{O}},J}$ is not a complete J-abstraction of $q_{\mathcal{S}}$.

Let $q'_{\mathcal{O}}$ be a UCQ such that $cert_{q_{\mathcal{O}},J} \not\sqsubseteq cert_{q'_{\mathcal{O}},J}$, that is, there exists an $\mathcal{S}$-database D consistent with J such that $cert^D_{q_{\mathcal{O}},J} \nsubseteq cert^D_{q'_{\mathcal{O}},J}$. It follows that there is a tuple of constant $\vec{c} = (c_1, \ldots, c_m)$ such that $\vec{c} \notin cert^D_{q'_{\mathcal{O}},J}$ and $\vec{c} \in cert^D_{q_{\mathcal{O}},J}$, i.e., $\vec{c} \in cert^D_{q^i_{\mathcal{O}},J}$ for at least one $i \in [1, n]$. Consider now the freezing $D_{q^i_{\mathcal{S}}}$ of $q^i_{\mathcal{S}} = \{\vec{t_i} \mid \exists y_i.\phi_i(\vec{x}_i, \vec{y}_i)\}$, here called simply D_i, and let $\vec{c^i}$ be the freezed tuple of constants $\vec{c^i} = (c^i_1, \ldots, c^i_m)$ where, for each $j \in [1, m]$, $c^i_j = t_j$ if t_j is a constant, and $c^i_j = c_x$ if $t_j = x$. Obviously, $\vec{c^i} \in {q^i_{\mathcal{S}}}^{D_i}$ trivially holds. We now prove that $\vec{c^i} \notin cert^{D_i}_{q'_{\mathcal{O}},J}$, thus showing that $q'_{\mathcal{O}}$ is not a complete J-abstraction of $q_{\mathcal{S}}$.

Consider $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$, i.e., the canonical structure of $\mathcal{O}$ with respect to $\mathcal{M}$ and D. Since $\vec{c} \in cert^D_{q^i_{\mathcal{O}},J}$ and D is consistent with J, we have that $\vec{c} \in {q^i_{\mathcal{O}}}^{\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}}$, implying that there exists a homomorphism h from $q^i_{\mathcal{O}}$ to $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ for which $h(\vec{t_i}) = \vec{c^i}$. Furthermore, due to the facts that $\mathcal{M}$ is a GLAV mapping and $\mathcal{O}$ is a *DL-Lite*$_{\mathcal{R}}$ ontology, and considering the construction of $q^i_{\mathcal{O}}$ and D_i, it is easy to see that there exists a function f from $\mathcal{C}^{\mathcal{M}(D_i)}_{\mathcal{O}}$ to $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ for which (i) $f(c) = h(c) = c$ for each constant c occurring in $q^i_{\mathcal{O}}$, (ii) $f(c_v) = h(v)$ for each variable $v \in \vec{x}_i \cup \vec{y}_i$ of $q^i_{\mathcal{S}}$ occurring in $\mathcal{M}(q^i_{\mathcal{S}})$, and (iii) $f(\mathcal{C}^{\mathcal{M}(D_i)}_{\mathcal{O}}) \subseteq \mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$, where $f(\mathcal{C}^{\mathcal{M}(D_i)}_{\mathcal{O}})$ is the image of $\mathcal{C}^{\mathcal{M}(D_i)}_{\mathcal{O}}$ under f. Observe that $f(\vec{c^i}) = \vec{c}$, and, since D is consistent with J, D_i is consistent with J as well. Due to the existence of this function f and the assumption that $\vec{c} \notin cert^D_{q'_{\mathcal{O}},J}$, we derive that there is no disjunct $q' = \{\vec{t'} \mid \exists y'.\phi'(\vec{x'}, \vec{y'})\}$ of $q'_{\mathcal{O}}$ for which there is a homomorphism h' from q' to $\mathcal{C}^{\mathcal{M}(D_i)}_{\mathcal{O}}$ such that $h'(\vec{t'}) = \vec{c^i}$, otherwise the function $f \circ h'$ would result in a homomorphism from q' to $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ such that $f(h'(\vec{t'})) = \vec{c}$, and therefore the assumption $\vec{c} \notin cert^D_{q'_{\mathcal{O}},J}$ would be contradicted. Thus, $\vec{c^i} \notin cert^{D_i}_{q'_{\mathcal{O}},J}$ as well. To conclude the proof, observe that D_i is an $\mathcal{S}$-database consistent with J for which $\vec{c^i} \in {q^i_{\mathcal{S}}}^{D_i}$ (and so $\vec{c^i} \in q^{D_i}_{\mathcal{S}}$) and $\vec{c^i} \notin cert^{D_i}_{q'_{\mathcal{O}},J}$, thus implying that $q'_{\mathcal{O}}$ is not a complete J-abstraction of $q_{\mathcal{S}}$. □

The following result is an immediate consequence of the above theorem.

Corollary 4.2. *The UCQ-minimally complete J-abstraction of $q_{\mathcal{S}}$ always exists. Furthermore, if $q_{\mathcal{S}}$ is a CQ, then it can be expressed as a CQ as well.*

Regarding the cost of the MinimallyComplete algorithm, we observe that, essentially, it applies the chase to each disjunct of $q_{\mathcal{S}}$ with respect to $\mathcal{M}$. This result into a running time that does not depend on $\mathcal{O}$ and $\mathcal{S}$, is exponential in $\sigma(\mathcal{M})$, and polynomial in $\sigma(q_{\mathcal{S}})$. Notice, however, that if $\mathcal{M}$ is a LAV mapping, then the application of the chase can be done in polynomial time in $\sigma(\mathcal{M})$ (indeed, in this case there is no conjunction of atoms to evaluate when applying the chase), and therefore the running time of the algorithm becomes polynomial in the size of the whole input.

Conversely, even in the case of pure GAV mappings, we next show that a polynomial time algorithm for computing UCQ-minimally complete abstractions already of CQJFEs would imply a polynomial time algorithm for checking whether $q_1 \sqsubseteq q_2$, where q_1 is a CQJFE and q_2 is a CQ. Since we also show that this latter problem is NP-hard, it turns out that, unless PTime = NP, the computation problem for complete abstractions cannot be solved in polynomial time, even in the case of pure GAV mappings $\mathcal{M}$ and CQJFEs $q_{\mathcal{S}}$.

Proposition 4.1. *There exists an OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ with $\mathcal{O}$ not comprising any assertion and $\mathcal{M}$ being a pure GAV mapping, and a CQJFE $q_{\mathcal{S}}$ such that, assuming PTime ⊂ NP, the UCQ-minimally complete J-abstraction of $q_{\mathcal{S}}$ cannot be computed in polynomial time.*

Proof. From a boolean CQJFE $q_1 = \{() \mid \exists \vec{y_1}.\psi(\vec{y_1})\}$ and a CQ $q_2 = \{() \mid \exists \vec{y_2}.\phi(\vec{y_2})\}$, we define the OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ as follows: the ontology $\mathcal{O}$ contains the atomic concept A and no assertions, the schema $\mathcal{S}$ is constituted by all predicates involved in $\psi(\vec{y_1})$ and in $\phi(\vec{y_2})$, plus an additional fresh unary predicate s, and the mapping $\mathcal{M}$ comprises the following pure GAV mapping assertion:

$\exists\vec{y}_2.s(x) \wedge \phi(\vec{y}_2) \rightarrow A(x)$.

We also define the boolean CQJFE over $\mathcal{S}$: $q_{\mathcal{S}} = \{() \mid \exists\vec{y}_1.\exists y.s(y) \wedge \psi(\vec{y}_1)\}$, where y denotes a fresh existential variable occurring neither in $\vec{y}_1$ nor in $\vec{y}_2$. It is easy to see that the UCQ-minimally complete J-abstraction of $q_{\mathcal{S}}$ is either the query $\{() \mid \exists y.A(y)\}$ (in particular, if it is a complete J-abstraction of $q_{\mathcal{S}}$), or the query $\{() \mid \exists y.\top(y)\}$.

Specifically, we now prove that $q_{\mathcal{O}} = \{() \mid \exists y.A(y)\}$ is the UCQ-minimally complete J-abstraction if and only if $q_1 \sqsubseteq q_2$. Due to Lemma 4.1, $q_{\mathcal{O}}$ is a complete J-abstraction of $q_{\mathcal{S}}$ if and only if $q_{\mathcal{S}} \sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}},J} \cup \mathsf{PerfRef}_{\mathcal{V}_{\mathcal{O}},J}$. Since $\mathcal{V}_{\mathcal{O}} = \{() \mid \exists y.\bot(y)\}$, in this case we have that $q_{\mathcal{O}}$ is a complete J-abstraction of $q_{\mathcal{S}}$ if and only if $q_{\mathcal{S}} \sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}},J}$. Notice, however, that $\mathsf{PerfRef}_{q_{\mathcal{O}},J} = \{() \mid \exists\vec{y}_2.\exists y.s(y) \wedge \phi(\vec{y}_2)\}$, and this implies that $q_{\mathcal{S}} \sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}},J}$ if and only if $q_1 \sqsubseteq q_2$.

We reduced the problem of checking whether $q_1 \sqsubseteq q_2$ for a boolean CQJFE q_1 and a boolean CQ q_2 to the problem of computing the UCQ-minimally complete J-abstraction of a CQJFE $q_{\mathcal{S}}$, where both $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and $q_{\mathcal{S}}$ can be constructed in LogSpace from q_1 and q_2.

So, a polynomial time algorithm for computing UCQ-minimally complete abstractions of CQJFEs $q_{\mathcal{S}}$ would imply a polynomial time algorithm for checking $q_1 \sqsubseteq q_2$, where q_1 is a CQJFE and q_2 is a CQ. Since by Corollary 4.1 we know that this latter containment problem is NP-hard, it follows that, unless PTime = NP, the computation problem for complete abstractions cannot be solved in polynomial time. □

5. Sound abstractions

We now turn our attention to study both the verification, and the computation problem for sound abstractions.

5.1. Verification

We recall that, for an $\mathcal{S}$-database D consistent with J, $\mathsf{PerfRef}^D_{q_{\mathcal{O}},J}$ computes exactly $cert^D_{q_{\mathcal{O}},J}$. So, checking whether $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$ means checking whether for all $\mathcal{S}$-databases D, either $Mod_D(J) = \emptyset$ or $\mathsf{PerfRef}^D_{q_{\mathcal{O}},J} \subseteq q^D_{\mathcal{S}}$. From this observation, we derive the following characterization.

Lemma 5.1. *$q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$ if and only if $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J})$, where $n = ar(q_{\mathcal{O}}) = ar(q_{\mathcal{S}})$.*

Proof. **"Only-if part:"** Suppose $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$. By definition, we have that for every $\mathcal{S}$-database D either D is not consistent with J, or $cert^D_{q_{\mathcal{O}},J} \subseteq q^D_{\mathcal{S}}$. In the former case, we have $\mathsf{PerfRef}^D_{\mathcal{V}_{\mathcal{O}},J} = \{()\}$, which obviously implies that $\mathsf{PerfRef}^D_{q_{\mathcal{O}},J} \subseteq \mathsf{PerfRef}^D_{\mathcal{V}^n_{\mathcal{O}},J}$. In the latter case, since D is consistent with J, we have that $cert^D_{q_{\mathcal{O}},J} = \mathsf{PerfRef}^D_{q_{\mathcal{O}},J}$. Therefore, we have that $\mathsf{PerfRef}^D_{q_{\mathcal{O}},J} \subseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J})^D$ for every $\mathcal{S}$-database D, as required.

"If part:" Suppose $q_{\mathcal{O}}$ is not a sound J-abstraction of $q_{\mathcal{S}}$, i.e., there is an $\mathcal{S}$-database D consistent with J such that $cert^D_{q_{\mathcal{O}},J} \not\subseteq q^D_{\mathcal{S}}$. Since D is consistent with J, we have (i) $\mathsf{PerfRef}^D_{\mathcal{V}_{\mathcal{O}},J} = \emptyset$, which implies ($i$) $\mathsf{PerfRef}^D_{\mathcal{V}^n_{\mathcal{O}},J} = \emptyset$ and (ii) $cert^D_{q_{\mathcal{O}},J} = \mathsf{PerfRef}^D_{q_{\mathcal{O}},J}$. So, for the $\mathcal{S}$-database D, we have that $\mathsf{PerfRef}^D_{q_{\mathcal{O}},J} \not\subseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J})^D$. Thus, $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \not\sqsubseteq q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J}$, as required. □

The following theorem characterizes the computational complexity of the verification problem for sound abstractions.

Theorem 5.1. *The verification problem for sound abstractions is Π^p_2-complete.*

Proof. As for the upper bound, by virtue of Lemma 5.1, it is sufficient to show how to check the containment $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J})$ in Π^p_2, where $n = ar(q_{\mathcal{O}})$. In particular, checking whether $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \not\sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J})$ can be done in Σ^p_2 as follows: (i) we guess a CQ q_1 over $\mathcal{S}$ with the same arity as $q_{\mathcal{O}}$ and size at most $\sigma(\mathcal{M}) \cdot \sigma(q_{\mathcal{O}})$, and ($ii$) with an NP-oracle, similarly to what described in Theorem 4.1, we first check whether $q_1 \sqsubseteq \mathsf{PerfRef}_{q_{\mathcal{O}},J}$, and then whether $q_1 \not\sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J})$, again using the method mentioned in Theorem 4.1.

As for the lower bound, the proof of Π^p_2-hardness is by a LogSpace reduction from the ∀∃3-CNF problem, which is Π^p_2-complete [57]. Let $F = c_1 \wedge \ldots \wedge c_p$ be a ∀∃3-CNF formula on two disjoint sets of variables X and Y, where $Y = \{y_1, \ldots, y_m\}$ (respectively, $X = \{x_1, \ldots, x_n\}$) are universally (respectively, existentially) quantified. ∀∃-CNF is the problem of deciding, given F, whether F is satisfiable, i.e., whether for each truth assignment α_Y to the variables in Y, there exists a truth assignment α_X to the variables in X such that $\alpha_Y \cup \alpha_X$ satisfies F. Moreover, each clause c_i is a disjunction of three literals, where each literal is either a variable in $Y \cup X$ or its negated.

We next show how to construct from F an OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and two queries $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$, such that F is satisfiable if and only if $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$, or equivalently, due to Lemma 5.1, $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J})$. First of all, we define $\mathcal{O}$ as empty, so that $\mathcal{V}_{\mathcal{O}} = \{() \mid \exists y.\bot(y)\}$. Thus, we will have to show that F is satisfiable if and only if $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \sqsubseteq q_{\mathcal{S}}$. Intuitively, $q_{\mathcal{S}}$, J, and $q_{\mathcal{O}}$ are defined so as to enforce that each homomorphism from $q_{\mathcal{S}}$ to a CQ in $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ defines a satisfying truth assignment for the variables of F, for a specific assignment to the universally quantified variables of F. In particular, this is achieved (i) by encoding through $q_{\mathcal{S}}$ which variables occur, positive or negated, in which position in every clause c_i and (ii) by defining $q_{\mathcal{O}}$ and $\mathcal{O}$ so that $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ is a UCQ such that for each possible truth assignment to the universally quantified variables of F, $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ comprises a disjunct encoding all possible satisfying truth assignments to the variables occurring in c_i, for every clause c_i.

Based on the intuition above, from a ∀∃3-CNF formula F, the OBDM specification J and the queries $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$ are defined as follows.

- J is the tuple $\langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, where:
  - the alphabet of $\mathcal{O}$ consists of the roles $R_{i,1}$, $R_{i,2}$, and $R_{i,3}$, for every $i \in [1, p]$, and the roles H_j and the concepts W_j, for every $j \in [1, m]$, and $\mathcal{O}$ does not contain any assertion;
  - $\mathcal{S}$ consists of three binary predicates, $s_{i,1}$, $s_{i,2}$, and $s_{i,3}$, for every $i \in [1, p]$, and of the unary predicates *zero*, *one*, and e_j, for every $j \in [1, m]$;
  - $\mathcal{M}$ contains the following mapping assertions $s_{i,1}(x_1, x_2) \to R_{i,1}(x_1, x_2)$, $s_{i,2}(x_1, x_2) \to R_{i,2}(x_1, x_2)$, $s_{i,3}(x_1, x_2) \to R_{i,3}(x_1, x_2)$, $e_j(x) \to W_j(x)$, $zero(x) \to H_j(x, 0)$ and $one(x) \to H_j(x, 1)$, for every $i \in [1, p]$ and $j \in [1, m]$;
- $q_{\mathcal{S}}$ is a boolean CQ, whose body contains (i) for each clause c_i, the atoms $s_{i,1}(z_i, \omega_{i,1})$, $s_{i,2}(z_i, \omega_{i,2})$, $s_{i,3}(z_i, \omega_{i,3})$ where z_i is a fresh existential variable, $\omega_{i,k}$ is the variable occurring (either positive or negated) as k-th literal in c_i, for $k \in [1, 2, 3]$, and (ii) for each universally quantified variable y_j in Y, the atom $e_j(y_j)$;
- $q_{\mathcal{O}}$ is a boolean CQ, whose body is the conjunction of atoms γ_1 and γ_2, such that (i) γ_1 contains, for every $h \in [1, \dots, 7]$, the atoms $R_{i,1}(A_{i,h}, v_{h,1})$, $R_{i,2}(A_{i,h}, v_{h,2})$, and $R_{i,3}(A_{i,h}, v_{h,3})$, where $A_{i,h}$ is a constant, and for every $k \in [1, 2, 3]$, $v_{h,k}$ is either the constant 1 or the constant 0 such that the assignment $(v_{h,1}, v_{h,2}, v_{h,3})$ to $(\omega_{i,1}, \omega_{i,2}, \omega_{i,3})$ satisfies c_i, where $\omega_{i,k}$ denotes the variable occurring (either positive or negated) as k-th literal in c_i; and (ii) γ_2 contains, for every universally quantified variable y_j in Y, the atoms $W_j(y_j)$ and $H_j(u_j, y_j)$, where u_j is a fresh variable.

To provide a better intuition of the encoding, consider the following example. Let F be the formula $(x_1 \vee x_2 \vee y_1) \bigwedge (\neg x_1 \vee \neg x_2 \vee \neg y_2)$. In this case, the reduction would produce the mapping $\mathcal{M}$ composed of the following mapping assertions:

$$\begin{array}{rclcrcl}
s_{1,1}(x_1, x_2) & \to & R_{1,1}(x_1, x_2), & \qquad & e_1(x) & \to & W_1(x), \\
s_{1,2}(x_1, x_2) & \to & R_{1,2}(x_1, x_2), & & e_2(x) & \to & W_2(x), \\
s_{1,3}(x_1, x_2) & \to & R_{1,3}(x_1, x_2), & & zero(x) & \to & H_1(x, 0), \\
s_{2,1}(x_1, x_2) & \to & R_{2,1}(x_1, x_2) & & zero(x) & \to & H_2(x, 0), \\
s_{2,2}(x_1, x_2) & \to & R_{2,2}(x_1, x_2) & & one(x) & \to & H_1(x, 1), \\
s_{2,3}(x_1, x_2) & \to & R_{2,3}(x_1, x_2) & & one(x) & \to & H_2(x, 1),
\end{array}$$

and the CQs $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$, such that:

- $q_{\mathcal{S}} = \{() \mid \exists z_1, z_2, x_1, x_2, y_1, y_2 . s_{1,1}(z_1, x_1) \wedge s_{1,2}(z_1, x_2) \wedge s_{1,3}(z_1, y_1) \wedge s_{2,1}(z_2, x_1) \wedge s_{2,2}(z_2, x_2) \wedge s_{2,3}(z_2, y_2) \wedge e_1(y_1) \wedge e_2(y_2)\ \}$;
- $q_{\mathcal{O}} = \{() \mid \exists u_1, u_2, y_1, y_2 . \gamma_1 \wedge \gamma_2\}$, where $\gamma_2 = W_1(y_1) \wedge H_1(u_1, y_1) \wedge W_2(y_2) \wedge H_2(u_2, y_2)$ and γ_1 is the following conjunction of atoms:

$$\begin{array}{l}
R_{1,1}(A_{1,1}, 0) \wedge R_{1,2}(A_{1,1}, 0) \wedge R_{1,3}(A_{1,1}, 1) \wedge R_{1,1}(A_{1,2}, 0) \wedge R_{1,2}(A_{1,2}, 1) \wedge R_{1,3}(A_{1,2}, 0) \wedge \\
R_{1,1}(A_{1,3}, 0) \wedge R_{1,2}(A_{1,3}, 1) \wedge R_{1,3}(A_{1,3}, 1) \wedge R_{1,1}(A_{1,4}, 1) \wedge R_{1,2}(A_{1,4}, 0) \wedge R_{1,3}(A_{1,4}, 0) \wedge \\
R_{1,1}(A_{1,5}, 1) \wedge R_{1,2}(A_{1,5}, 0) \wedge R_{1,3}(A_{1,5}, 1) \wedge R_{1,1}(A_{1,6}, 1) \wedge R_{1,2}(A_{1,6}, 1) \wedge R_{1,3}(A_{1,6}, 0) \wedge \\
R_{1,1}(A_{1,7}, 1) \wedge R_{1,2}(A_{1,7}, 1) \wedge R_{1,3}(A_{1,7}, 1) \wedge R_{2,1}(A_{2,1}, 0) \wedge R_{2,2}(A_{2,1}, 0) \wedge R_{2,3}(A_{2,1}, 0) \wedge \\
R_{2,1}(A_{2,2}, 0) \wedge R_{2,2}(A_{2,2}, 0) \wedge R_{2,3}(A_{2,2}, 1) \wedge R_{2,1}(A_{2,3}, 0) \wedge R_{2,2}(A_{2,3}, 1) \wedge R_{2,3}(A_{2,3}, 0) \wedge \\
R_{2,1}(A_{2,4}, 0) \wedge R_{2,2}(A_{2,4}, 1) \wedge R_{2,3}(A_{2,4}, 1) \wedge R_{2,1}(A_{2,5}, 1) \wedge R_{2,2}(A_{2,5}, 0) \wedge R_{2,3}(A_{2,5}, 0) \wedge \\
R_{2,1}(A_{2,6}, 1) \wedge R_{2,2}(A_{2,6}, 0) \wedge R_{2,3}(A_{2,6}, 1) \wedge R_{2,1}(A_{2,7}, 1) \wedge R_{2,2}(A_{2,7}, 1) \wedge R_{2,3}(A_{2,7}, 0)
\end{array}$$

Then, $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ is obtained by unfolding (i) the atoms of γ_1 into atoms encoding all possible satisfying assignments for the variables occurring in every clause, and (ii) the atoms of γ_2 either into the atoms $e_j(0) \wedge zero(u_j)$ by setting $y_j = 0$, or into the atoms $e_j(1) \wedge one(u_j)$ by setting $y_j = 1$, for every $j \in [1, 2]$. For example, the disjunct q' of $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ obtained by unfolding the atoms of γ_2 by setting $y_1 = 0$ and $y_2 = 1$, is the following:

$$\begin{array}{rl}
q' = \{() \mid \exists u_1, u_2 . & s_{1,1}(A_{1,1}, 0) \wedge s_{1,2}(A_{1,1}, 0) \wedge s_{1,3}(A_{1,1}, 1) \wedge s_{1,1}(A_{1,2}, 0) \wedge s_{1,2}(A_{1,2}, 1) \wedge s_{1,3}(A_{1,2}, 0) \wedge \\
& s_{1,1}(A_{1,3}, 0) \wedge s_{1,2}(A_{1,3}, 1) \wedge s_{1,3}(A_{1,3}, 1) \wedge s_{1,1}(A_{1,4}, 1) \wedge s_{1,2}(A_{1,4}, 0) \wedge s_{1,3}(A_{1,4}, 0) \wedge \\
& s_{1,1}(A_{1,5}, 1) \wedge s_{1,2}(A_{1,5}, 0) \wedge s_{1,3}(A_{1,5}, 1) \wedge s_{1,1}(A_{1,6}, 1) \wedge s_{1,2}(A_{1,6}, 1) \wedge s_{1,3}(A_{1,6}, 0) \wedge
\end{array}$$

$$s_{1,1}(A_{1,7},1) \wedge s_{1,2}(A_{1,7},1) \wedge s_{1,3}(A_{1,7},1) \wedge s_{2,1}(A_{2,1},0) \wedge s_{2,2}(A_{2,1},0) \wedge s_{2,3}(A_{2,1},0) \wedge$$
$$s_{2,1}(A_{2,2},0) \wedge s_{2,2}(A_{2,2},0) \wedge s_{2,3}(A_{2,2},1) \wedge s_{2,1}(A_{2,3},0) \wedge s_{2,2}(A_{2,3},1) \wedge s_{2,3}(A_{2,3},0) \wedge$$
$$s_{2,1}(A_{2,4},0) \wedge s_{2,2}(A_{2,4},1) \wedge s_{2,3}(A_{2,4},1) \wedge s_{2,1}(A_{2,5},1) \wedge s_{2,2}(A_{2,5},0) \wedge s_{2,3}(A_{2,5},0) \wedge$$
$$s_{2,1}(A_{2,6},1) \wedge s_{2,2}(A_{2,6},0) \wedge s_{2,3}(A_{2,6},1) \wedge s_{2,1}(A_{2,7},1) \wedge s_{2,2}(A_{2,7},1) \wedge s_{2,3}(A_{2,7},0) \wedge$$
$$e_1(0) \wedge \mathit{zero}(u_1) \wedge e_2(1) \wedge \mathit{one}(u_2)\ \}$$

One can easily verify that the function h such that: $h(y_1) = 0$, $h(y_2) = 1$, $h(z_1) = A_{1,2}$, $h(z_2) = A_{2,4}$, $h(x_1) = 0$, $h(x_2) = 1$ is a homomorphism from $q_{\mathcal{S}}$ to q', and that the restriction of h to $Y \cup X$ defines an assignment α_X that makes F true for the assignment α_Y such that $\alpha_Y(y_1) = 0$ and $\alpha_Y(y_2) = 1$.

In fact, this example illustrates a crucial property of our construction. Specifically, one can verify that, by construction, there is a one-to-one correspondence between disjuncts of $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ (totally, 2^m) and truth assignments to the variables in Y. Indeed, the choice done for unfolding the atoms $H_1(u_1, y_1), \ldots, H_m(u_m, y_m)$, forces each y_j to be equal to 0 or to 1. More precisely, for every $j \in [1, m]$, if $H_j(u_j, y_j)$ is unfolded with the atom $\mathit{zero}(u_j)$, then $y_j = 0$, otherwise, if it is unfolded with the atom $\mathit{one}(u_j)$, then $y_j = 1$. Note in particular that this implies that in each disjunct of $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ appears either the atom $e_j(0)$ (if $y_j = 0$) or the atom $e_j(1)$ (if $y_j = 1$), for every $j \in [1, m]$.

We are now ready to prove that F is satisfiable if and only if $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \sqsubseteq q_{\mathcal{S}}$, i.e., there exists a homomorphism from $q_{\mathcal{S}}$ to every disjunct of $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$.

"**Only-if part:**" Suppose that F is true, that is, for every truth assignment α_Y to the variables in Y, there exists a truth assignment α_X to the variables in X that satisfies F. It is easy to see that the function h obtained by combining a given α_Y and a given α_X can be extended to every z_i, for $i \in [1, \ldots, p]$, so as to obtain a homomorphism from $q_{\mathcal{S}}$ to the disjunct in $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ corresponding to α_Y. Based on the observation above, this implies that if F is satisfiable, then there exists a homomorphism from $q_{\mathcal{S}}$ to every disjunct of $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$.

"**If part:**" Suppose that there exists a homomorphism between $q_{\mathcal{S}}$ to every disjunct in $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$. Let q' be any disjunct in $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ and h be a homomorphism from $q_{\mathcal{S}}$ to q'. It is easy to see that the restriction of h to X defines a truth assignment α_X that makes F true for the assignment α_Y corresponding to q'. Hence, by the observation above, we can conclude that for every α_Y, there exists α_X that satisfies F. □

Note that the above result already holds when the mapping is both GAV and LAV. Furthermore, with a slight modification of the above reduction, it can be shown that the Π_2^p-hardness holds also when the mapping is pure GAV (but not LAV). It remains an interesting open problem the computational complexity of the verification problem for sound abstractions when the mapping is both pure GAV and LAV.

5.2. Computation

We now address the problem of computing UCQ-maximally sound abstractions. Our main result is that there are many cases where UCQ-maximally sound abstractions are not guaranteed to exist. In order to illustrate the result, starting from the general scenario described at the end of Section 3, we introduce a *restricted scenario*, where

- the setting for OBDM specifications is obtained from the general one by both limiting the DL ontology language to *DL-Lite*$_{\mathsf{RDFS}}$ rather than *DL-Lite*$_{\mathcal{R}}$, and limiting the mapping language to follow the pure GAV approach rather than the GLAV approach;
- the source query language $\mathcal{L}_{\mathcal{S}}$ is UCQJFEs.

We now show that, surprisingly, as soon as we try to depart from the restricted scenario, either by extending the query language $\mathcal{L}_{\mathcal{S}}$ to CQs, or by extending the setting for OBDM specifications, we lose the guarantee of the existence of UCQ-maximally sound abstractions of queries over $\mathcal{S}$.

Theorem 5.2. *UCQ-maximally sound abstractions of a query $q_{\mathcal{S}}$ may not exist if we extend the restricted scenario with one of the following features:*

1. *$q_{\mathcal{S}}$ expressed in a fragment of CQs allowing joining existential variables, thus going beyond UCQJFEs.*
2. *disjointness assertions in the ontology;*
3. *inclusion assertions of the form $B \sqsubseteq \exists R$ in the ontology, where B is a basic concept and R is a basic role;*
4. *LAV mapping assertions in the mapping, even without joins involving existential variables in the right-hand side;*
5. *non-pure GAV mapping assertions in the mapping.*

Proof. We next consider the cases 1 and 2, and then refer to Appendix A for cases 3 to 5, since they rely on a similar line of reasoning.

Case 1. Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \emptyset$
- $\mathcal{S} = \{\ s_1, s_2, s_3, s_4, s_5\ \}$
- $\mathcal{M} = \{\ m_1, m_2, m_3, m_4\ \}$, where:

$$\begin{array}{lrcl} m_1: & s_1(x) & \rightarrow & A_1(x), \\ m_2: & s_2(x_1) \wedge s_3(x_1, x_2) & \rightarrow & P(x_1, x_2), \\ m_3: & s_1(x_2) \wedge s_5(x_1, x_2) & \rightarrow & P(x_1, x_2), \\ m_4: & s_2(x) \wedge s_4(x) & \rightarrow & A_2(x). \end{array}$$

Moreover, let $q_{\mathcal{S}}$ be the following boolean CQ over $\mathcal{S}$: $q_{\mathcal{S}} = \{() \mid \exists y.s_1(y) \wedge s_2(y)\}$, which has a join existential variable, and therefore goes beyond UCQJFEs.

First we show that there exists an infinite number of CQs over $\mathcal{O}$ that are sound J-abstractions of $q_{\mathcal{S}}$. Then, based on this, we show that no UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ exists.

Specifically, for every $n \geq 1$, let $q^n_{\mathcal{O}}$ be defined as follows:

- if $n = 1$, then $q^1_{\mathcal{O}} = \{() \mid \exists y_1.A_1(y_1) \wedge A_2(y_1)\}$;
- if $n > 1$, then $q^n_{\mathcal{O}} = \{() \mid \exists y_1, \ldots, y_n.A_1(y_1) \wedge \left(\bigwedge_{j=1}^{n-1} P(y_j, y_{j+1})\right) \wedge A_2(y_n)\}$.

We show, by induction on n, that $q^n_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$, for every $n \geq 1$. As for the base step ($n = 1$), one can easily see that $q^1_{\mathcal{O}} = \{() \mid \exists y_1.A_1(y_1) \wedge A_2(y_1)\}$ is a sound J-abstraction of $q_{\mathcal{S}}$, since $\mathsf{PerfRef}_{q^1_{\mathcal{O}},J} = \{() \mid \exists y_1.s_1(y_1) \wedge s_2(y_1) \wedge s_4(y_1)\} \sqsubseteq q_{\mathcal{S}}$.

As for the inductive step ($n > 1$), suppose that $q^n_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$. Now, let us substitute the atom $A_2(y_n)$ in $q^n_{\mathcal{O}}$ with the conjunction $P(y_n, y_{n+1}) \wedge A_2(y_{n+1})$, thus obtaining the query $q^{n+1}_{\mathcal{O}}$. Note that every disjunct of $\mathsf{PerfRef}_{q^{n+1}_{\mathcal{O}},J}$ can be obtained from some disjunct of $\mathsf{PerfRef}_{q^n_{\mathcal{O}},J}$ by substituting the conjunction $s_2(y_n) \wedge s_4(y_n)$ either with the conjunction $s_2(y_n) \wedge s_3(y_n, y_{n+1}) \wedge s_2(y_{n+1}) \wedge s_4(y_{n+1})$ or with $s_1(y_{n+1}) \wedge s_5(y_n, y_{n+1}) \wedge s_2(y_{n+1}) \wedge s_4(y_{n+1})$. But then, since $q^n_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$, we know that every disjunct in $\mathsf{PerfRef}_{q^n_{\mathcal{O}},J}$ contains the conjunction of atoms $s_1(y_i) \wedge s_2(y_i)$ for some $i \in [1, n]$. Thus, let us denote by q_i any disjunct in $\mathsf{PerfRef}_{q^n_{\mathcal{O}},J}$ that contains the conjunction $s_1(y_i) \wedge s_2(y_i)$. Either $i \in [1, n-1]$, in which case, for sure, also the disjuncts of $\mathsf{PerfRef}_{q^{n+1}_{\mathcal{O}},J}$ that are obtained from q_i contain $s_1(y_i) \wedge s_2(y_i)$ and are therefore contained in $q_{\mathcal{S}}$, or $i = n$, in which case the two disjuncts obtained from q_i contain either the conjunction $s_1(y_n) \wedge s_2(y_n)$ or the conjunction $s_1(y_{n+1}) \wedge s_2(y_{n+1})$, and hence, are also contained in $q_{\mathcal{S}}$. This proves that $\mathsf{PerfRef}_{q^{n+1}_{\mathcal{O}},J} \sqsubseteq q_{\mathcal{S}}$, and hence that $q^{n+1}_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$.

Now, let $\bar{Q}_{\mathcal{O}}$ be a UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ and let k be the maximum number of atoms occurring in the body of the disjuncts of $\bar{Q}_{\mathcal{O}}$. Also, consider the query $q^k_{\mathcal{O}}$ (which contains $k+1$ atoms) and consider the disjunct q of $\mathsf{PerfRef}_{q^k_{\mathcal{O}},J}$ obtained by rewriting (*i*) $A_1(y_1)$ with m_1, (*ii*) $P(y_i, y_{i+1})$ with m_2, for all $i \in [1, n-1]$, and (*iii*) $A_2(y_n)$ with m_4. Obviously, $() \in cert^{D_q}_{q^k_{\mathcal{O}},J}$. Hence, since $q^k_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$, we have that $() \in q^{D_q}_{\mathcal{S}}$. Also, $() \in cert^{D_q}_{\bar{Q}_{\mathcal{O}},J}$ (otherwise $Q_{\mathcal{O}} = \bar{Q}_{\mathcal{O}} \cup q^k_{\mathcal{O}}$ would be a sound J-abstraction of $q_{\mathcal{S}}$ such that $cert_{\bar{Q}_{\mathcal{O}},J} \sqsubset cert_{Q_{\mathcal{O}},J}$, thus contradicting the fact that $\bar{Q}_{\mathcal{O}}$ is a UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$). Now, consider $\mathcal{M}(D_q)$. Since the ontology is empty, we have that $() \in \bar{Q}_{\mathcal{O}}^{\mathcal{M}(D_q)}$, where, by construction, $\mathcal{M}(D_q)$ comprises $k+1$ facts, i.e., the facts that can be obtained by substituting with fresh constants the variables occurring in the atoms of the body of $q^k_{\mathcal{O}}$. But then, since the body of the disjuncts of $\bar{Q}_{\mathcal{O}}$ comprises at most k atoms, this implies that a subset M' of $\mathcal{M}(D_q)$, consisting of k atoms, is sufficient to make $() \in \bar{Q}_{\mathcal{O}}^{M'}$, or, equivalently, $\bar{Q}_{\mathcal{O}}$ true in M'. In particular, let us consider one such subset M': it is obtained by removing from $\mathcal{M}(D_q)$ either $A_1(c_{y_1})$, $A_2(c_{y_k})$, or $P(c_{y_i}, c_{y_{i+1}})$, for some $i \in [1, k-1]$. Suppose it is obtained by removing $A_1(c_{y_1})$. Also, let us consider the database $D' = D_{q'}$, such that q' is obtained from q by unfolding every atom $P(y_i, y_{i+1})$, for $i \in [1, k-1]$, with $s_2(y_i) \wedge s_3(y_i, y_{i+1})$ (i.e., with m_2), and the remaining atom $A_2(y_k)$ with $s_2(y_k) \wedge s_4(y_k)$ (i.e., with m_4). One can easily verify that D' is such that $\mathcal{M}(D') = M'$, and hence $() \in cert^{D'}_{\bar{Q}_{\mathcal{O}},J}$. On the other hand, $() \notin q^{D'}_{\mathcal{S}}$ since for every $s_2(c_{y_i})$, $i \in [1, k-1]$, $s_1(c_{y_i}) \notin D'$. Hence, we get a contradiction to the fact that $\bar{Q}_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$. Similarly, we get a contradiction if we suppose that M' is obtained by removing either $A_2(c_{y_k})$, or $P(c_{y_i}, c_{y_{i+1}})$, for some $i \in [1, k-1]$. This proves that no UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ can exist, and completes the proof.

Case 2. Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \{\ A \sqsubseteq \neg A\ \}$
- $\mathcal{S} = \ \{\ s_1, s_2, s_3, s_4\ \}$

- $\mathcal{M} = \{\ m_1, m_2, m_3, m_4, m_5, m_6\ \}$, where:

$$\begin{array}{lrcl} m_1: & s_1(x_1, x_2) & \rightarrow & P_1(x_1, x_2), \\ m_2: & s_2(x_1, x_2) & \rightarrow & P_1(x_1, x_2), \\ m_3: & s_2(x_1, x_2) & \rightarrow & P_2(x_1, x_2), \\ m_4: & s_3(x_1, x_2) & \rightarrow & P_2(x_1, x_2), \\ m_5: & s_3(x_1, x_2) \wedge s_4(x_2) & \rightarrow & P_3(x_1, x_2), \\ m_6: & \exists y_1, y_2 . s_2(y_1, x) \wedge s_3(x, y_2) & \rightarrow & A(x). \end{array}$$

Moreover, let $q_{\mathcal{S}}$ be the following full CQ over $\mathcal{S}$: $q_{\mathcal{S}} = \{(x_1, x_2) \mid s_1(x_1, x_2)\}$.

Since $\mathcal{O}$ contains the disjointness assertion $A \sqsubseteq \neg A$, the violation query for $\mathcal{O}$ is $\mathcal{V}_{\mathcal{O}} = \{() \mid \exists y . A(y)\}$, and therefore $\mathcal{V}^2_{\mathcal{O}} = \{(x_1, x_2) \mid \exists y . A(y) \wedge \top(x_1) \wedge \top(x_2)\}$. The proof is similar to that of case 1. In particular, we show that there exists an infinite number of CQs $q^i_{\mathcal{O}}$ over $\mathcal{O}$ that are sound J-abstractions of $q_{\mathcal{S}}$ and show that, based on this, assuming that a UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ exists, leads to a contradiction. For every $n \geq 1$, let $q^n_{\mathcal{O}}$ be defined as follows:

- if $n = 1$, then $q^1_{\mathcal{O}} = \{(x_1, x_2) \mid \exists y_1 . P_1(x_1, x_2) \wedge P_3(x_2, y_1)\}$;
- if $n > 1$, then $q^n_{\mathcal{O}} = \{(x_1, x_2) \mid \exists y_1, \ldots, y_n . P_1(x_1, x_2) \wedge P_2(x_2, y_1) \wedge \big(\bigwedge_{j=1}^{j=n-2} P_2(y_j, y_{j+1})\big) \wedge P_3(y_{n-1}, y_n)\}$.

In fact, we show, by induction on n, that $q^n_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$. To this aim we recall that, by Lemma 5.1, $q^n_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$ if $\mathsf{PerfRef}_{q^n_{\mathcal{O}}, J} \sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}}, J})$.

For the base step, notice that

$$\mathsf{PerfRef}_{q^1_{\mathcal{O}}, J} = \{(x_1, x_2) \mid \exists y_1 . s_1(x_1, x_2) \wedge s_3(x_2, y_1) \wedge s_4(y_1)\} \ \cup \ \{(x_1, x_2) \mid \exists y_1 . s_2(x_1, x_2) \wedge s_3(x_2, y_1) \wedge s_4(y_1)\}$$

Clearly, the first disjunct of $\mathsf{PerfRef}_{q^1_{\mathcal{O}}, J}$ is contained in $q_{\mathcal{S}}$, whereas the second is contained in $\mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}}, J}$. Hence, $q^1_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$.

As for the inductive step, suppose that $q^n_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$. Now, let us substitute the atom $P_3(y_{n-1}, y_n)$ in $q^n_{\mathcal{O}}$ with the conjunction $P_2(y_{n-1}, y_n) \wedge P_3(y_n, y_{n+1})$. It is easy to see that we get the query $q^{n+1}_{\mathcal{O}}$. Moreover, every disjunct of $\mathsf{PerfRef}_{q^{n+1}_{\mathcal{O}}, J}$ can be obtained from some disjunct of $\mathsf{PerfRef}_{q^n_{\mathcal{O}}, J}$ by substituting the conjunction $s_3(z, y_n) \wedge s_4(y_n)$, where z is either x_2 or y_{n-1}, either with the conjunction $s_2(z, y_n) \wedge s_3(y_n, y_{n+1}) \wedge s_4(y_{n+1})$ if $P_2(y_{n-1}, y_n)$ is unfolded with m_3, or with the conjunction $s_3(z, y_n) \wedge s_3(y_n, y_{n+1}) \wedge s_4(y_{n+1})$ if $P_2(y_{n-1}, y_n)$ is unfolded with m_4. But then, since $q^n_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$, we know that every disjunct in $\mathsf{PerfRef}_{q^n_{\mathcal{O}}, J}$ contains either an atom $s_1(x_1, x_2)$ or a conjunction of the form $s_2(z_1, z_2) \wedge s_3(z_2, z_3)$ for some variables $z_i, i \in [1, 2, 3]$. Hence, for every disjunct q' of $\mathsf{PerfRef}_{q^{n+1}_{\mathcal{O}}, J}$, we have three possible cases:

- q' is obtained from a disjunct of $\mathsf{PerfRef}_{q^n_{\mathcal{O}}, J}$ that contains $s_1(x_1, x_2)$; it thus contains $s_1(x_1, x_2)$, too, and, hence, $q' \sqsubseteq q_{\mathcal{S}}$;
- q' is obtained from a disjunct of $\mathsf{PerfRef}_{q^n_{\mathcal{O}}, J}$ that contains a conjunction of the form $s_2(z_1, z_2) \wedge s_3(z_2, y_i)$ for some variables z_1, z_2 and y_i, for $i \in [1, n-2]$; clearly, such a conjunction is also contained in q' and hence $q' \sqsubseteq \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}}, J}$; or,
- q' is obtained from a disjunct of $\mathsf{PerfRef}_{q^n_{\mathcal{O}}, J}$ that contains a conjunction of the form $s_2(z_1, z_2) \wedge s_3(z_2, y_n)$ for some variables z_1, z_2; in this case, $s_3(z_2, y_n)$ has been replaced (together with $s_4(y_n)$) either with the conjunction $s_2(z_2, y_n) \wedge s_3(y_n, y_{n+1}) \wedge s_4(y_{n+1})$ or with the conjunction $s_3(z_2, y_n) \wedge s_3(y_n, y_{n+1}) \wedge s_4(y_{n+1})$; then, in both cases, $q' \sqsubseteq \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}}, J}$ (in the former case because of the presence of the conjunction $s_2(z_2, y_n) \wedge s_3(y_n, y_{n+1})$, whereas in the latter because of the presence of the conjunction $s_2(z_1, z_2) \wedge s_3(z_2, y_n)$).

Hence, for every such disjunct q', we have that $q' \sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^2_{\mathcal{O}}, J})$, which proves that $q^{n+1}_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$.

Now, let $\bar{Q}_{\mathcal{O}}$ be a UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ and let k be the maximum number of atoms occurring in the body of the disjuncts of $\bar{Q}_{\mathcal{O}}$. Also, consider the query $q^k_{\mathcal{O}}$ (which contains $k+1$ atoms) and consider the disjunct q of $\mathsf{PerfRef}_{q^k_{\mathcal{O}}, J}$ obtained from $q^k_{\mathcal{O}}$ by rewriting: (*i*) $P(x_1, x_2)$ into $s_1(x_1, x_2)$ (i.e., with m_1), (*ii*) $P_2(z, y_i)$ into $s_3(z, y_i)$ (i.e., with m_4), for every $i \in [1, n-1]$, and (*iii*) $P_3(y_{n-1}, y_n)$ into $s_3(y_{n-1}, y_n), s_4(y_n)$ (i.e., with m_5). Then, $D_q = \{s_1(c_{x_1}, c_{x_2}), s_3(c_z, c_{y_1}), \ldots, s_3(c_z, c_{y_{n-1}}), s_3(c_{y_{n-1}}, c_{y_n}), s_4(c_{y_n})\}$. D_q is consistent with J since s_2 is empty. Also, obviously, $(c_{x_1}, c_{x_2}) \in cert^{D_q}_{q^k_{\mathcal{O}}, J}$. But then, since $q^k_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$, $(c_{x_1}, c_{x_2}) \in q^{D_q}_{\mathcal{S}}$, and hence $(c_{x_1}, c_{x_2}) \in cert^{D_q}_{\bar{Q}_{\mathcal{O}}, J}$ (otherwise $Q_{\mathcal{O}} = \bar{Q}_{\mathcal{O}} \cup q^k_{\mathcal{O}}$ would be a sound J-abstraction of $q_{\mathcal{S}}$ such that $cert_{\bar{Q}_{\mathcal{O}}, J} \sqsubset cert_{Q_{\mathcal{O}}, J}$, thus contradicting the fact that $\bar{Q}_{\mathcal{O}}$ is a UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$). Now, consider $\mathcal{M}(D_q)$. Since the ontology contains only a disjointness assertion, we have that $(c_{x_1}, c_{x_2}) \in \bar{Q}^{\mathcal{M}(D_q)}_{\mathcal{O}}$, where $\mathcal{M}(D_q)$ comprises at least the $k+1$ facts that can be obtained by substituting with fresh constants the variables occurring in the atoms of the body of $q^k_{\mathcal{O}}$. But then, since the body of the disjuncts of $\bar{Q}_{\mathcal{O}}$ comprise at most k atoms, this means that there exists a subset M' of $\mathcal{M}(D_q)$, consisting of k atoms, that is sufficient to make (c_{x_1}, c_{x_2}) belong to $\bar{Q}^{M'}_{\mathcal{O}}$. In particular, let us consider one such subset M'. We have the following:

- M' must contain the fact $P_1(c_{x_1}, c_{x_2})$; indeed, suppose it does not contain it, and consider the database D' obtained from D_q by removing $s_1(c_{x_1}, c_{x_2})$. Clearly, D' is consistent with J since it does not contain any atom s_2. Moreover, D' is such that $\mathcal{M}(D') = M'$, and hence $(c_{x_1}, c_{x_2}) \in cert^{D'}_{\bar{Q}_{\mathcal{O}},J}$, while $(c_{x_1}, c_{x_2}) \notin q^{D'}_{\mathcal{S}}$. Clearly, this would contradict that $\bar{Q}_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$.
- M' must contain the atom $P_2(c_{x_2}, c_{y_1})$; indeed, suppose it does not contain it and consider the database D' obtained from D_q by removing $s_3(c_{x_2}, c_{y_1})$ and substituting $s_1(c_{x_1}, c_{x_2})$ with $s_2(c_{x_1}, c_{x_2})$. It is easy to see that D' is consistent with J and that it is such that $\mathcal{M}(D') = M'$, and hence $(c_{x_1}, c_{x_2}) \in cert^{D'}_{\bar{Q}_{\mathcal{O}},J}$, while $(c_{x_1}, c_{x_2}) \notin q^{D'}_{\mathcal{S}}$. Again, this would contradict that $\bar{Q}_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$.
- for every $i \in [1, k-2]$, M' must contain the fact $P_2(c_{y_i}, c_{y_{i+1}})$; indeed, suppose it does not contain it and consider the database D' obtained from D_q by removing $s_3(c_{y_i}, c_{y_{i+1}})$ and substituting $s_1(c_{x_1}, c_{x_2})$ with $s_2(c_{x_1}, c_{x_2})$, and $s_3(z, c_{y_j})$ with $s_2(z, c_{y_j})$ for every z and every $j \in [1, i]$. Similarly to the previous cases, this would lead to a contradiction.
- M' must contain the atom $P_3(c_{y_{n-1}}, c_{y_n})$; indeed, similarly to the case above, if we assume that it does not contain it, then we get a contradiction.

Hence, M' must contain at least $k+1$ facts, which contradicts the fact that M' comprises only k atoms, and shows that no UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ can exist. □

We end this section by pointing out that it can be shown that, if we disallow constants in UCQs, then UCQ-maximally sound abstractions are still not guaranteed to exist if we extend the restricted scenario with one of the features 1, 2, 3 or 4 mentioned in Theorem 5.2. On the contrary, we conjecture that non-pure GAV mappings would preserve the existence of UCQ-maximally sound abstractions if constants were not allowed in UCQs.

6. Perfect abstractions

With the results of previous chapters at hand, we now study both the verification and the computation problem for perfect abstractions.

6.1. Verification

We remind the reader that, by definition, $q_{\mathcal{O}}$ is a perfect J-abstraction of $q_{\mathcal{S}}$ if and only if it is both a sound and a complete J-abstraction of $q_{\mathcal{S}}$. Thus, by combining Lemma 5.1 and Lemma 4.1, we immediately get the following.

Corollary 6.1. *$q_{\mathcal{O}}$ is a perfect J-abstraction of $q_{\mathcal{S}}$ if and only if both $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J})$ and $q_{\mathcal{S}} \sqsubseteq (\mathsf{PerfRef}_{q_{\mathcal{O}},J} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J})$ hold, where $n = ar(q_{\mathcal{O}}) = ar(q_{\mathcal{S}})$.*

The following theorem characterizes the computational complexity of the verification problem for perfect abstractions.

Theorem 6.1. *The verification problem for perfect abstractions is Π^p_2-complete.*

Proof. Let us discuss the lower bound first. As already observed in Section 3, for OBDM specifications where inconsistencies cannot arise, the notion of perfect abstraction coincides with the notion of realization considered in [15]. Since the problem of checking whether $q_{\mathcal{O}}$ is a realization of $q_{\mathcal{S}}$ in J is in general Π^p_2-hard even when $\mathcal{O}$ contains no assertions (i.e., $\mathcal{O} = \emptyset$), $\mathcal{M}$ is a pure GAV mapping, and both $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$ are boolean CQs [15, Theorem 11], and since in those cases the two notions are equivalent, such lower bound applies also to our notion.

As for the upper bound, by virtue of Corollary 6.1, it is sufficient to show how to check the following two containments in Π^p_2, where $n = ar(q_{\mathcal{O}}) = ar(q_{\mathcal{S}})$: (i) $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J})$, which holds if and only if $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$; and (ii) $q_{\mathcal{S}} \sqsubseteq (\mathsf{PerfRef}_{q_{\mathcal{O}},J} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J})$, which holds if and only if $q_{\mathcal{O}}$ is a complete J-abstraction of $q_{\mathcal{S}}$. By Theorem 5.1, the former can be verified in Π^p_2, and, by Theorem 4.1, the latter can be verified even in NP. □

We leave as an interesting open problem the question of whether the computational complexity of the verification problem for perfect abstractions decreases or not when $\mathcal{M}$ is a LAV mapping.

6.2. Computation

As for computation, consider any OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and UCQ $q_{\mathcal{S}}$ over $\mathcal{S}$. We have that (i) the UCQ-minimally complete J-abstraction of $q_{\mathcal{S}}$ always exists and can be computed by means of the MinimallyComplete algorithm

Algorithm Perfect.

```
Input: OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$; UCQ $q_{\mathcal{S}}$ over $\mathcal{S}$ of arity $n$
Output: either a UCQ $q_{\mathcal{O}}$ over $\mathcal{O}$, or report that "no UCQ-perfect $J$-abstraction of $q_{\mathcal{S}}$ exists"
$q_{\mathcal{O}} :=$ MinimallyComplete$(J, q_{\mathcal{S}})$
if $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \sqsubseteq (q_{\mathcal{S}} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J})$ then
    return $q_{\mathcal{O}}$
else
    return "no UCQ-perfect $J$-abstraction of $q_{\mathcal{S}}$ exists"
end if
```

(cf. Theorem 4.2); and (*ii*) by construction (see Definition 3.4), either this latter is also a sound, and therefore a perfect, J-abstraction of $q_{\mathcal{S}}$, or no UCQ-perfect J-abstraction of $q_{\mathcal{S}}$ exists. With these observations at hand, we can easily derive the algorithm Perfect together with its termination and correctness.

Essentially, the algorithm computes the UCQ-minimally complete J-abstraction of $q_{\mathcal{S}}$ using the MinimallyComplete algorithm (cf. Section 4), and then checks whether this latter is also a sound (see Lemma 5.1), and therefore a perfect, J-abstraction of $q_{\mathcal{S}}$. Notice that this last step is always deterministically feasible in exponential time (cf. Theorem 5.1). Finally, observe that the overall running time of the algorithm is exponential in the size of the input.

Example 6.1. Recall the OBDM specification J and the UCQ $q_{\mathcal{S}}$ illustrated in Example 1.1. One can verify that $\mathsf{Perfect}(J, q_{\mathcal{S}})$ returns the UCQ $q_{\mathcal{O}} = \{(x) \mid \mathsf{Professor}(x)\} \cup \{(x) \mid \exists y.\mathsf{TeachesTo}(x, y)\}$ (that is J-equivalent to $\{(x) \mid \mathsf{Professor}(x)\}$), which corresponds to the UCQ-perfect J-abstraction of $q_{\mathcal{S}}$. △

Example 6.2. Recall the OBDM specification J and the UCQ $q_{\mathcal{S}}$ illustrated in Example 3.2 (resp., Example 4.1). One can verify that $\mathsf{Perfect}(J, q_{\mathcal{S}})$ reports that "no UCQ-perfect J-abstraction of $q_{\mathcal{S}}$ exists". △

Theorem 6.2. *$\mathsf{Perfect}(J, q_{\mathcal{S}})$ terminates and returns the perfect J-abstraction of $q_{\mathcal{S}}$ if it exists and can be expressed as a UCQ, otherwise it reports that no UCQ-perfect J-abstraction of $q_{\mathcal{S}}$ exists.*

Furthermore, as a straightforward consequence of Corollary 4.2, we also get the following interesting result.

Corollary 6.2. *The UCQ-perfect J-abstraction of a CQ $q_{\mathcal{S}}$ either does not exists, or it can be expressed as a CQ as well.*

Finally, we briefly discuss the case of perfect abstractions under the semantics used in [15], that imposes the condition $q^D_{\mathcal{S}} = cert^D_{q_{\mathcal{O}},J}$ for *all* $\mathcal{S}$-databases D. From the results presented in the previous sections, we easily get the following.

Proposition 6.1. *$q_{\mathcal{O}}$ is a perfect J-abstraction of $q_{\mathcal{S}}$ under the semantics of [15] (i.e., $q_{\mathcal{O}}$ is a realization of $q_{\mathcal{S}}$ in J) if and only if $q_{\mathcal{S}} \equiv (\mathsf{PerfRef}_{q_{\mathcal{O}},J} \cup \mathsf{PerfRef}_{\mathcal{V}^n_{\mathcal{O}},J})$, where $n = ar(q_{\mathcal{O}}) = ar(q_{\mathcal{S}})$.*

Note that the above proposition allows us to easily derive algorithms for both verification and computation as well as an upper-bound for verification, under the semantics of [15], too.

7. Sound abstractions in the restricted scenario

We now deal with the restricted scenario mentioned in Section 5. In particular, we start by providing results on containment for UCQJFEs, which will be useful to tackle verification and computation in the following subsections. Then, we conclude the section by briefly analyzing a special case of the restricted scenario, namely when source queries are CQJFEs.

Before proceeding, we observe that, despite its limitations, the expressive power of the restricted scenario is enough for various meaningful applications. Indeed, several popular ontologies are expressible in the DL *DL-Lite*$_{\mathsf{RDFS}}$, e.g., `SKOS`[8] [58,59] and `Dublin Core`[9] [60], and the form of pure GAV mapping is exactly the one originally defined in the literature of data integration [23]. Furthermore, the class of (U)CQJFEs captures data services expressible in the famous (U)SPJ (Union, Select, Project, Join) fragment of Relational Algebra [42], with the only limitation that joining variables must appear in the final projection of the USPJ Relational Algebra query, i.e., they appear in the target list of the equivalent UCQ. Notice, moreover, that such fragment is precisely the one needed for all tasks related to source profiling [17,18]. Finally, as an example, we point out that the OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and the query $q_{\mathcal{S}}$ illustrated in Example 3.2 fall in the restricted scenario.

[8] `Simple Knowledge Organization System`: https://www.w3.org/2004/02/skos/.

[9] http://dublincore.org/.

7.1. Containment of UCQJFEs

We now study containment for UCQJFEs. To this aim, we start by introducing some crucial notions and results related to UCQJFEs.

Definition 7.1. Let $q = \{\vec{t} \mid \exists\vec{y}.\phi(\vec{x},\vec{y})\}$ be a CQ over a schema $\mathcal{S}$, and let $\alpha_1 = s(t_{1,1},\ldots,t_{1,n})$ and $\alpha_2 = s(t_{2,1},\ldots,t_{2,n})$ be two atoms over $\mathcal{S}$, where $\alpha_2 \in \phi(\vec{x},\vec{y})$. We say that α_1 *instantiates* α_2, if the following holds for each $i \in [1,n]$: if $t_{2,i}$ is a distinguished variable or a constant, then $t_{2,i} = t_{1,i}$.

Example 7.1. Consider the following CQs $q_1 = \{(x) \mid s_1(c,x,x) \wedge s_2(c,x)\}$ and $q_2 = \{(x) \mid \exists y.s_1(c,x,y) \wedge s_2(x,x)\}$ over a schema $\mathcal{S}$. Let $\alpha_1 = s_1(c,x,x)$, $\alpha_2 = s_2(c,x)$, $\alpha_3 = s_1(c,x,y)$, and $\alpha_4 = s_2(x,x)$. We have that α_1 instantiates α_3, whereas α_2 does not instantiate α_4. △

Clearly, given atoms α_1 and α_2 occurring in the bodies of CQs q_1 and q_2, respectively, checking whether α_1 instantiates α_2 can be done in polynomial time. Based on this observation, the following lemmata show that checking whether a UCQ q_1 is contained in a UCQJFE q_2 can be done in polynomial time as well.

Lemma 7.1. *Let q_1 and q_2 be a CQ and a CQJFE, respectively, over a schema $\mathcal{S}$ with the same target list. We have that $q_1 \sqsubseteq q_2$ if and only if for each atom α_2 of q_2 there exists an atom α_1 of q_1 such that α_1 instantiates α_2.*

Proof. "**Only-if part:**" Suppose that $q_1 \sqsubseteq q_2$, that is, there exists a homomorphism from q_2 to q_1. But then, since q_1 and q_2 have the same target list, and since q_2 is a CQJFE, by construction it can be readily seen that for each atom α_2 of q_2 there is at least an atom α_1 of q_1 such that α_1 instantiates α_2.

"**If part:**" Suppose that for each atom α_2 of q_2 there exists an atom α_1 of q_1 such that α_1 instantiates α_2. Let h be the function from the terms of q_2 to the terms of q_1 such that (*i*) $h(c) = c$, for each constant c appearing in q_2, (*ii*) $h(x) = x$, for every distinguished variable x, and finally (*iii*) $h(y) = t$ for every existential variable y occurring in q_2, where if y occurs as k-th argument of atom α_2 (since q_2 is a CQJFE, only one occurrence of y exists), then t is the k-th argument of the atom α_1 that instantiates α_2 (which exists by assumption). Since q_2 is a CQJFE, and since q_2 and q_1 have the same target list, we derive that h consists in a homomorphism from q_2 to q_1. It follows that $q_1 \sqsubseteq q_2$. □

We are now ready to tackle the containment problem for UCQJFEs, which is: given a UCQ q' and a UCQJFE q over the same schema $\mathcal{S}$, check whether $q' \sqsubseteq q$.

Lemma 7.2. *The containment problem for UCQJFEs is in* PTime.

Proof. To begin, observe that for each pair of UCQs q_1, q_2 we have $q_1 \sqsubseteq q_2$ if and only if for each disjunct q' of q_1 there exists a disjunct q of q_2 such that $q' \sqsubseteq q$ [45]. It is therefore sufficient to show that, given a CQ $q' = \{\vec{t'} \mid \exists\vec{y'}.\phi'(\vec{x'},\vec{y'})\}$ and a CQJFE $q = \{\vec{t} \mid \exists\vec{y}.\phi(\vec{x},\vec{y})\}$ not necessarily with the same target lists, checking whether $q' \sqsubseteq q$ can be done in polynomial time.

If q' and q do not have the same target list, i.e., $\vec{t'} = (t'_1,\ldots,t'_n) \neq \vec{t} = (t_1,\ldots,t_n)$, then consider the function f from the set of terms in the target list of q to the set of terms in the target list of q' with $f(t_i) = t'_i$, for each $i \in [1,n]$. Formally, since repetitions of terms in target lists is allowed, f might give rise to a *multivalued function*.[10] In this case, as well as in the case that $f(a) = b$ with $a \neq b$ for two constants $a \in \vec{t}$ and $b \in \vec{t'}$, it is straightforward to verify that $q' \not\sqsubseteq q$ trivially holds. Indeed, in those cases there can be no homomorphism from q to q' by construction.

Consider the query q'' obtained in polynomial time from q by replacing every occurrence of term t_i in q (even in the target list) with term $f(t_i) = t'_i$, for each $i \in [1,n]$. Observe that now q'' is a CQJFE with target list $\vec{t'}$, i.e., the same target list of q'. By virtue of Lemma 7.1, we can now check in polynomial time whether $q' \sqsubseteq q''$, where, if the answer is yes, then $q' \sqsubseteq q$; otherwise, it can be readily seen that $D_{q'}$ (i.e., the freezing of q') is a database witnessing that $q' \not\sqsubseteq q$.

From the above considerations, it is immediate to derive a polynomial time algorithm for checking whether a UCQ q_1 is contained in a UCQJFE q_2. □

In what follows, unless otherwise stated, we assume that OBDM specifications are expressed in the restricted setting for OBDM specifications mentioned in Section 5, i.e., the DL ontology language is *DL-Lite*$_{\mathsf{RDFS}}$ and the mapping language follows the pure GAV approach.

[10] In mathematics, a *multivalued function* (also known as *multiple-valued function* [61]) $f : A \to B$ is similar to a function, but it may associate more than one possible element $y \in B$ to each element $x \in A$.

7.2. Verification

We now address the verification problem for sound abstractions in the restricted scenario. The following theorem characterizes the computational complexity of the verification problem for sound abstractions in such scenario.

Theorem 7.1. *In the restricted scenario, the verification problem for sound abstractions is coNP-complete.*

Proof. As for the upper bound, since *DL-Lite*$_{\mathsf{RDFS}}$ does not allow for disjointness assertions, it is sufficient to show how to check the containment $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \sqsubseteq q_{\mathcal{S}}$ in coNP. In particular, checking $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \not\sqsubseteq q_{\mathcal{S}}$ can be done in NP in the following way: we guess (i) a disjunct $\overline{q_{\mathcal{O}}}$ in $q_{\mathcal{O}}$, (ii) a query q' over $\mathcal{O}$ with the same arity as $\overline{q_{\mathcal{O}}}$ and size at most $\sigma(\overline{q_{\mathcal{O}}})$, (iii) a sequence $\rho_{\mathcal{O}}$ of at most $\sigma(\mathcal{O})\cdot\sigma(\overline{q_{\mathcal{O}}})$ ontology assertions in $\mathcal{O}$, (iv) a query q'' over $\mathcal{S}$ with the same arity as $\overline{q_{\mathcal{O}}}$ and size at most $\sigma(\mathcal{M})\cdot\sigma(q')$ (v) a sequence $\rho_{\mathcal{M}}$ of at most $\sigma(q')$ pure GAV mapping assertions in $\mathcal{M}$. Then, we check in polynomial time (i) whether q' is obtained by rewriting $\overline{q_{\mathcal{O}}}$ using the sequence $\rho_{\mathcal{O}}$ (i.e., whether $q' \in \mathsf{PerfRef}_{\overline{q_{\mathcal{O}}},\mathcal{O}}$), (ii) whether q'' is obtained by rewriting q' using the sequence $\rho_{\mathcal{M}}$ (i.e., whether $q'' \in \mathsf{REW}_{\mathcal{M}}(q')$ and hence $q'' \in \mathsf{PerfRef}_{\overline{q_{\mathcal{O}}},J}$), and finally (iii) whether $q'' \not\sqsubseteq q_{\mathcal{S}}$, which, since $q_{\mathcal{S}}$ is a UCQJFE, by virtue of Lemma 7.2, it can be done in polynomial time.

As for the lower bound, the proof of coNP-hardness is by a LogSpace reduction from the validity problem, which is coNP-complete (see, e.g., [62]). validity is the problem of deciding, given a 3-DNF formula $F = c_1 \vee \ldots \vee c_m$ on a set of variables $X = \{x_1, \ldots, x_n\}$, whether F is *valid*, i.e., whether F is satisfied by every possible truth assignment to the variables in X. Each clause c_i is a conjunction of three literals, where each literal is either a variable $x_i \in X$ or its negated.

We define an OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ with $\mathcal{O}$ empty, and $\mathcal{S}$ and $\mathcal{M}$ as follows: for each variable $x_i \in X$, schema $\mathcal{S}$ comprises two unary relations s_{i_T} and s_{i_F}, and a further unary relation s'_i. Finally, for each variable $x_i \in X$, the mapping $\mathcal{M}$ includes the following three mapping assertions:

- $s_{i_T}(x) \rightarrow A_i(x)$,
- $s_{i_F}(x) \rightarrow A_i(x)$,
- $s'_i(x) \rightarrow B_i(x)$,

where each A_i and B_i are fresh atomic concepts, for each $i \in [1, n]$.

Intuitively, while each s'_i is simply mirrored to B_i, the possible unfoldings of an atom $A_i(x_i)$ (which are $s_{i_T}(x_i)$ and $s_{i_F}(x_i)$, respectively) correspond to the possible truth values (true and false, respectively) for the variable x_i.

We define the UCQJFE over $\mathcal{S}$ as $q_{\mathcal{S}} = q_1 \cup \ldots \cup q_m$, where, for each $i \in [1, m]$, the target list of q_i is $\vec{x} = (x_1, \ldots, x_n)$ and the body of q_i has the conjunction of atoms $s'_1(x_1) \wedge \ldots \wedge s'_n(x_n)$ in conjunction to the conjunction of atoms associated to the clause c_i of F, where a positive literal x_i is replaced with the atom $s_{i_T}(x_i)$, whereas a negative literal $\neg x_i$ is replaced with the atom $s_{i_F}(x_i)$.

Finally, we define the CQJFE over $\mathcal{O}$ as $q_{\mathcal{O}} = \{\vec{x} \mid B_1(x_1) \wedge \ldots \wedge B_n(x_n) \wedge A_1(x_1) \wedge \ldots \wedge A_n(x_n)\}$.

Observe that $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, $q_{\mathcal{S}}$, and $q_{\mathcal{O}}$ can be constructed in LogSpace from F, where $\mathcal{O} = \emptyset$ and $\mathcal{M}$ is both a pure GAV and a LAV mapping.

We now prove that formula F is valid if and only if $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$.

"**Only-if part:**" Suppose that formula F is valid, that is, F is satisfied by every possible truth assignment to the variables in X. It follows that, for any possible choice for unfolding the atoms $A_i(x_i)$ for $i = 1, \ldots, n$ (which can be equivalently seen as an assignment $V = (v_1, \ldots, v_n)$ to the variables $X = (x_1, \ldots, x_n)$), the query over $\mathcal{S}$ obtained is such that all the atoms also appear in a disjunct q_j of $q_{\mathcal{S}}$ for some $j \in [1, m]$ (equivalently, at least one clause c_j for some $j \in [1, m]$ is satisfied under the truth assignment V). It follows that $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \sqsubseteq q_{\mathcal{S}}$ which, since *DL-Lite*$_{\mathsf{RDFS}}$ does not allow for disjointness assertions, implies that $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$.

"**If part:**" Suppose that formula F is not valid, that is, there exists a truth assignment $V = (v_1, \ldots, v_n)$ to the variables in $X = (x_1, \ldots, x_n)$ that does not satisfy F. Consider now the disjunct q of $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ obtained by unfolding atom $A_i(x_i)$ of $q_{\mathcal{O}}$ with atom $s_{i_T}(x_i)$ if $v_i = 1$, and with atom $s_{i_F}(x_i)$ otherwise (i.e., $v_i = 0$), for each $i \in [1, n]$. As a result, for each disjunct q' of $q_{\mathcal{S}}$, there is at least one atom of q' not occurring in q. In proof, if there exists some disjunct q_j of $q_{\mathcal{S}}$ such that every atom of q_j appears also in q, then the clause c_j corresponding to disjunct q_j is satisfied under the truth assignment V, which would contradict the fact that F is not satisfied under such truth assignment. This implies that, for each disjunct q_j of $q_{\mathcal{S}}$, there is no homomorphism from q_j to q. It follows that $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \not\sqsubseteq q_{\mathcal{S}}$ which implies that $q_{\mathcal{O}}$ is not a sound J-abstraction of $q_{\mathcal{S}}$. □

Note that (i) the coNP upper bound holds even when ontologies $\mathcal{O}$ are expressed in a fragment of *DL-Lite*$_{\mathcal{R}}$ that does not admit disjointness assertions (thus, a more expressive language of *DL-Lite*$_{\mathsf{RDFS}}$) and mappings $\mathcal{M}$ are GLAV mappings (rather than pure GAV mappings), and (ii) the coNP lower bound already holds when $q_{\mathcal{O}}$ is a CQJFE, both $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$ do not have existential variables, and $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ is such that $\mathcal{O}$ is empty, and $\mathcal{M}$ is both a pure GAV mapping and a LAV mapping. In the following section, we will see that the computational complexity of the verification problem further decreases when $q_{\mathcal{S}}$ is restricted to be a CQJFE.

7.3. Computation

We now address the computation problem by providing an algorithm to compute UCQ-maximally sound abstractions, thus proving that UCQ-maximally sound abstractions are guaranteed to exist in the restricted scenario.

We start by introducing some useful notation. Given an OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ of the restricted setting and an atom β over $\mathcal{O}$, we denote by $\rho(\beta, J)$ the disjunction of conjunctions obtained by first unfolding β with respect to $\mathcal{O}$, and then by unfolding the resulting formula with respect to $\mathcal{M}$. The unfolding of an atom β with respect to a *DL-Lite*$_{\mathsf{RDFS}}$ ontology $\mathcal{O}$, denoted $\lambda(\beta, \mathcal{O})$, is the disjunction of atoms defined as in [63] (there called AtomRewr):

$$\lambda(A(t), \mathcal{O}) = \bigvee_{A' \colon \mathcal{O} \models A' \sqsubseteq A} A'(t) \;\vee \bigvee_{P \colon \mathcal{O} \models \exists P \sqsubseteq A} (\exists y. P(t, y)) \;\vee \bigvee_{P \colon \mathcal{O} \models \exists P^- \sqsubseteq A} (\exists y. P(y, t)),$$

$$\lambda(P(t_1, t_2), \mathcal{O}) = \bigvee_{E \colon \mathcal{O} \models E \sqsubseteq P} E(t_1, t_2) \;\vee \bigvee_{E \colon \mathcal{O} \models E^- \sqsubseteq P} E(t_2, t_1),$$

where y denotes a fresh existential variable, A and A' denote atomic concepts, and P and E denote atomic roles. Then, as for the unfolding of $\lambda(\beta, \mathcal{O})$ with respect to $\mathcal{M}$, it is obtained by replacing each atom β' occurring in $\lambda(\beta, \mathcal{O})$ with the logical disjunction of all the conjunctions of atoms over $\mathcal{S}$ corresponding to the left-hand sides of mapping assertions in $\mathcal{M}$ having the predicate name β' in the right-hand side (being careful to use unique variables in place of those variables that appear in the left-hand side of the mapping assertions but not in the right-hand side of those).

Example 7.2. Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \{\ \exists P_2 \sqsubseteq A\ \}$
- $\mathcal{S} = \{\ s_1, s_2, s_3\ \}$
- $\mathcal{M} = \{\ m_1, m_2, m_3, m_4\ \}$, where:

$$\begin{array}{lrcl}
m_1\colon & s_1(x_1, x_2) & \rightarrow & P_1(x_1, x_2), \\
m_2\colon & \exists y. s_1(x_1, y) \wedge s_2(y, x_2) & \rightarrow & P_1(x_1, x_2), \\
m_3\colon & \exists y. s_2(c, x) \wedge s_3(x, y) & \rightarrow & A(x), \\
m_4\colon & s_3(x_1, x_2) & \rightarrow & P_2(x_1, x_2).
\end{array}$$

Consider the atoms $\beta_1 = P_1(y, x)$ and $\beta_2 = A(x)$ over $\mathcal{O}$. We have $\lambda(\beta_1, \mathcal{O}) = \beta_1$ and $\lambda(\beta_2, \mathcal{O}) = \beta_2 \vee (\exists y_2. P_2(x, y_2))$. Thus, $\rho(\beta_1, J) = (s_1(y, x)) \vee (\exists y_1. s_1(y, y_1) \wedge s_2(y_1, x))$, whereas $\rho(\beta_2, J) = (\exists y_3. s_2(c, x) \wedge s_3(x, y_3)) \vee (\exists y_2. s_3(x, y_2))$. △

Finally, since *DL-Lite*$_{\mathsf{RDFS}}$ ontologies $\mathcal{O}$ contain no assertions with $\exists R$ occurring in the right-hand side for a basic role R, and since pure GAV mappings do not allow for repetitions of variables or constants in the right-hand side of mapping assertions, given an OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ of the restricted setting and a CQ $q_{\mathcal{O}} = \{\vec{t} \mid \exists \vec{y}. \phi(\vec{x}, \vec{y})\}$ over $\mathcal{O}$, it can be readily seen that $\mathsf{PerfRef}_{q_{\mathcal{O}}, J}$ is equivalent to turning the following logical query into an equivalent UCQ over $\mathcal{S}$:

$$\{\vec{t} \mid \exists \vec{y}. \bigwedge_{\beta \in \phi(\vec{x}, \vec{y})} \rho(\beta, J)\}$$

Example 7.3. Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the OBDM specification illustrated in Example 7.2. Consider the CQ $q_{\mathcal{O}} = \{(x) \mid \exists y. P_1(y, x) \wedge A(x)\}$ over $\mathcal{O}$, and let $\beta_1 = P_1(y, x)$ and $\beta_2 = A(x)$. Then, $\mathsf{PerfRef}_{q_{\mathcal{O}}, J}$ can be obtained by turning the logical query $\{(x) \mid \exists y. \rho(\beta_1, J) \wedge \rho(\beta_2, J)\} = \{(x) \mid \exists y. ((s_1(y, x)) \vee (\exists y_1. s_1(y, y_1) \wedge s_2(y_1, x))) \wedge ((\exists y_3. s_2(c, x) \wedge s_3(x, y_3)) \vee (\exists y_2. s_3(x, y_2)))\}$ into an equivalent UCQ over schema $\mathcal{S}$, thus obtaining $\mathsf{PerfRef}_{q_{\mathcal{O}}, J} = q^1_{\mathcal{S}} \cup q^2_{\mathcal{S}} \cup q^3_{\mathcal{S}} \cup q^4_{\mathcal{S}}$, where:

- $q^1_{\mathcal{S}} = \{(x) \mid \exists y, y_3. s_1(y, x) \wedge s_2(c, x) \wedge s_3(x, y_3)\}$;
- $q^2_{\mathcal{S}} = \{(x) \mid \exists y, y_2. s_1(y, x) \wedge s_3(x, y_2)\}$;
- $q^3_{\mathcal{S}} = \{(x) \mid \exists y, y_1, y_3. s_1(y, y_1) \wedge s_2(y_1, x) \wedge s_2(c, x) \wedge s_3(x, y_3)\}$;
- $q^4_{\mathcal{S}} = \{(x) \mid \exists y, y_2. s_1(y, y_1) \wedge s_2(y_1, x) \wedge s_3(x, y_2)\}$. △

Now, for a mapping $\mathcal{M}$, we denote by $\gamma(\mathcal{M})$ the number of mapping assertions occurring in $\mathcal{M}$. For a UCQ $q_{\mathcal{S}}$, we denote by $\eta(q_{\mathcal{S}})$ the sum of the number of atoms occurring in the body of the various disjuncts of $q_{\mathcal{S}}$. Then, for a mapping $\mathcal{M}$ and a UCQ $q_{\mathcal{S}}$, we define $bound(\mathcal{M}, q_{\mathcal{S}})$ as:

$$bound(\mathcal{M}, q_{\mathcal{S}}) = \sum_{i=0}^{\eta(q_{\mathcal{S}})} \gamma(\mathcal{M})^i$$

The next crucial lemma shows that, given any OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and any UCQJFE $q_{\mathcal{S}}$ over $\mathcal{S}$, we can always limit our attention to CQs over $\mathcal{O}$ having at most $bound(\mathcal{M}, q_{\mathcal{S}})$ atoms occurring in their bodies when seeking for CQ-maximally sound J-abstractions of $q_{\mathcal{S}}$.

Lemma 7.3. *Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification, and let $q_{\mathcal{S}}$ be a UCQJFE over $\mathcal{S}$. If a CQ $q_{\mathcal{O}}$ over $\mathcal{O}$ is a sound J-abstraction of $q_{\mathcal{S}}$, then there exists a CQ $q'_{\mathcal{O}}$ over $\mathcal{O}$ with same target list of $q_{\mathcal{O}}$ such that (*i*) the body of $q'_{\mathcal{O}}$ is the conjunction of at most $bound(\mathcal{M}, q_{\mathcal{S}})$ atoms occurring in the body of $q_{\mathcal{O}}$ (and therefore, $cert_{q_{\mathcal{O}},J} \sqsubseteq cert_{q'_{\mathcal{O}},J}$), and (*ii*) $q'_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$ as well.*

Proof. Let n denote the arity of $q_{\mathcal{O}}$ and $q_{\mathcal{S}}$, and let the target list of $q_{\mathcal{O}}$ be $\vec{t} = (t_1, \ldots, t_n)$. Without loss of generality, we can assume that $\vec{t}$ does not contain any constant nor repeated variable. Also, since $\mathcal{M}$ is composed of pure GAV mapping assertions, the target list of each disjunct in $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ is the same as that of $q_{\mathcal{O}}$. On the other hand, since $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$ and that *DL-Lite*$_{\mathsf{RDFS}}$ does not allow for disjointness assertions, we have that $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \sqsubseteq q_{\mathcal{S}}$, that is, each disjunct of $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ is contained in some disjunct of $q_{\mathcal{S}}$. As a matter of fact, without loss of generality, we can assume that: (*i*) each disjunct q of $q_{\mathcal{S}}$ is such that there is some disjunct r of $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ for which $r \sqsubseteq q$ (indeed, the other disjuncts of $q_{\mathcal{S}}$ that do not satisfy the above condition can simply be discarded, and the resulting $q_{\mathcal{S}}$ will remain such that $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \sqsubseteq q_{\mathcal{S}}$); and (*ii*) the target list of each disjunct q of $q_{\mathcal{S}}$ is the same as that of $q_{\mathcal{O}}$ (indeed, it does not contain any constant or repeated variable neither, and thus, its variables can be safely renamed).

We now show by induction on $\eta(q_{\mathcal{S}})$ (i.e., the sum of the number of atoms occurring in the body of the various disjuncts of $q_{\mathcal{S}}$) that if $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$, then there exists $m \leq bound(\mathcal{M}, q_{\mathcal{S}})$ atoms $\beta_1, \ldots, \beta_m$ occurring in the body of $q_{\mathcal{O}}$ for which the CQ $q'_{\mathcal{O}} = \{\vec{t} \mid \exists \vec{y}.\beta_1 \wedge \ldots \wedge \beta_m\}$ is a sound J-abstraction of $q_{\mathcal{S}}$, thus proving the claim. We do this by exploiting Lemma 7.1. Specifically, consider each CQ r that is a disjunct of $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$. We know that there is a CQJFE q that is a disjunct of $q_{\mathcal{S}}$ for which $r \sqsubseteq q$, and, moreover, since r and q have the same target list, by Lemma 7.1 we know that for each atom α_2 of q there exists an atom α_1 of r such that α_1 instantiates α_2.

Base step ($\eta(q_{\mathcal{S}}) = 1$): In this case, $q_{\mathcal{S}}$ is a single CQJFE whose body consists of only one atom α. So, there must exist at least one atom β in the body of $q_{\mathcal{O}}$ for which every possible disjunct of $\rho(\beta, J)$ contains at least one atom that instantiates α. Indeed, if this is not the case, then the disjunct r of $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ obtained by unfolding each atom β' of $q_{\mathcal{O}}$ with a disjunct of $\rho(\beta', J)$ which contains no atom that instantiates α would be such that $r \not\sqsubseteq q_{\mathcal{S}}$, implying that $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \not\sqsubseteq q_{\mathcal{S}}$, which would contradict that $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$. Thus, such atom β in the body of $q_{\mathcal{O}}$ must exist. But then, by exploiting Lemma 7.1, it is trivial to see that the CQ $q'_{\mathcal{O}} = \{\vec{t} \mid \exists \vec{y}.\beta\}$ is a sound J-abstraction of $q_{\mathcal{S}}$, as required.

Inductive step: Following the same line of reasoning of the base step, one can show that there must be (at least) one atom β in the body of $q_{\mathcal{O}}$ such that in every disjunct of $\rho(\beta, J)$ there is at least one atom that instantiates some atom occurring in the various disjuncts of $q_{\mathcal{S}}$. In particular, consider $\rho(\beta, J)$ for such atom β. For each disjunct θ_i of $\rho(\beta, J)$, let $q_{\mathcal{S}}^{\theta_i}$ be the UCQJFE obtained from $q_{\mathcal{S}}$ by removing all the atoms α such that an atom of θ_i instantiates α. Notice that, since each disjunct θ_i of $\rho(\beta, J)$ instantiates some atom of $q_{\mathcal{S}}$, each UCQJFE $q_{\mathcal{S}}^{\theta_i}$ is such that (*i*) $\eta(q_{\mathcal{S}}^{\theta_i}) \leq \eta(q_{\mathcal{S}}) - 1$ (i.e., there is at least an atom of $q_{\mathcal{S}}$ not occurring anymore in $q_{\mathcal{S}}^{\theta_i}$), and (*ii*) $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}^{\theta_i}$, due to the facts that $q_{\mathcal{S}} \sqsubseteq q_{\mathcal{S}}^{\theta_i}$ clearly holds and the initial assumption that $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$. Hence, by inductive hypothesis, for each disjunct θ_i of $\rho(\beta, J)$, there are atoms $\beta_1^{\theta_i}, \beta_2^{\theta_i}, \ldots, \beta_{p_i}^{\theta_i}$ for which $q_{\mathcal{O}}^{\theta_i} = \{\vec{t} \mid \exists \vec{y}_{\theta_i}.\beta_1^{\theta_i} \wedge \beta_2^{\theta_i} \wedge \ldots \wedge \beta_{p_i}^{\theta_i}\}$ is a sound J-abstraction of $q_{\mathcal{S}}^{\theta_i}$, where $p_i \leq bound(\mathcal{M}, q_{\mathcal{S}}^{\theta_i})$, i.e., $p_i \leq 1 + \lambda(\mathcal{M})^1 + \lambda(\mathcal{M})^2 + \ldots + \lambda(\mathcal{M})^{\eta(q_{\mathcal{S}})-1}$. But then, consider the following CQ $q'_{\mathcal{O}}$:

$$\{\vec{t} \mid \exists \vec{y}.\beta \bigwedge \beta_1^{\theta_1} \wedge \beta_2^{\theta_1} \wedge \ldots \wedge \beta_{p_1}^{\theta_1} \bigwedge \beta_1^{\theta_2} \wedge \beta_2^{\theta_2} \wedge \ldots \wedge \beta_{p2}^{\theta_2} \bigwedge \ldots \bigwedge \beta_1^{\theta_k} \wedge \beta_2^{\theta_k} \wedge \ldots \wedge \beta_{p_k}^{\theta_k}\}$$

It is not hard to ascertain that $q'_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$, where k is the number of disjuncts in $\rho(\beta, J)$, and $p_i \leq 1 + \lambda(\mathcal{M})^1 + \lambda(\mathcal{M})^2 + \ldots + \lambda(\mathcal{M})^{\eta(q_{\mathcal{S}})-1}$ for each $i \in [1, k]$. In proof, consider each disjunct θ_i of $\rho(\beta, J)$ for $i \in [1, k]$. Since the CQ $q_{\mathcal{O}}^{\theta_i}$ is a sound J-abstraction of $q_{\mathcal{S}}^{\theta_i}$, by Lemma 7.1, we derive that for each possible disjunct r^{θ_i} obtained by turning in disjunctive normal form the formula $\rho(\beta_1^{\theta_i}, J) \wedge \rho(\beta_2^{\theta_i}, J) \wedge \ldots \wedge \rho(\beta_{p_i}^{\theta_i}, J)$ there is a disjunct q^{θ_i} of $q_{\mathcal{S}}^{\theta_i}$ for which for each atom α of q^{θ_i} there is an atom of r^{θ_i} that instantiates α. This, together with the fact that all the atoms α occurring in $q_{\mathcal{S}}$ and not occurring in $q_{\mathcal{S}}^{\theta_i}$ are such that there is an atom of θ_i that instantiates α, allows us to derive that each disjunct $r_{\wedge}^{\theta_i}$ in the formula $\theta_i \wedge \rho(\beta_1^{\theta_i}, J) \wedge \rho(\beta_2^{\theta_i}, J) \wedge \ldots \wedge \rho(\beta_{p_i}^{\theta_i}, J)$ turned in disjunctive normal form is such that there is a disjunct q of $q_{\mathcal{S}}$ for which for each atom α of q there is an atom of $r_{\wedge}^{\theta_i}$ that instantiates α. Since this is true for each disjunct θ_i of $\rho(\beta, J)$, and since for each $i \in [1, k]$ the conjunction of atoms $\beta_1^{\theta_i} \wedge \beta_2^{\theta_i} \wedge \ldots \wedge \beta_{p_i}^{\theta_i}$ occurs in the body of the CQ $q'_{\mathcal{O}}$, we easily derive that for each possible disjunct r' of $q'_{\mathcal{O}}$ there is a disjunct q of $q_{\mathcal{S}}$ for which for each atom α of q there is an atom of r' that instantiates α. Thus, by Lemma 7.1, it follows that $q'_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$, as required.

To conclude the proof, observe that the number of disjuncts in $\rho(\beta, J)$ is at most $k \leq \lambda(\mathcal{M})$, and therefore, since $p_i \leq 1 + \lambda(\mathcal{M})^1 + \lambda(\mathcal{M})^2 + \ldots + \lambda(\mathcal{M})^{\eta(q_{\mathcal{S}})-1}$ for each $i \in [1, k]$, we derive that the number of atoms occurring in the body of the CQ $q'_{\mathcal{O}}$ is at most $1 + \lambda(\mathcal{M})^1 + \lambda(\mathcal{M})^2 + \ldots + \lambda(\mathcal{M})^{\eta(q_{\mathcal{S}})}$, as required. □

Algorithm MaximallySoundForUCQJFEs.

Input: OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$; UCQJFE over $q_{\mathcal{S}}$ over $\mathcal{S}$ of arity n
Output: UCQ $q_{\mathcal{O}}$ over $\mathcal{O}$

```
$q_{\mathcal{O}} := \{(x_1, \ldots, x_n) \mid \bot(x_1) \wedge \ldots \wedge \bot(x_n)\}$
for each CQ $q$ over $\mathcal{O}$ having at most $bound(\mathcal{M}, q_{\mathcal{S}})$ atoms in its body and possibly involving constants occurring in $q_{\mathcal{S}}$ or in $\mathcal{M}$ as terms do
    if $\mathsf{PerfRef}_{q,J} \sqsubseteq q_{\mathcal{S}}$ then
        $q_{\mathcal{O}} := q_{\mathcal{O}} \cup q$
    end if
end for
return $q_{\mathcal{O}}$
```

By relying on the above lemma, we immediately derive the following enumerative algorithm MaximallySoundForUCQJFEs for computing UCQ-maximally sound abstractions in the restricted scenario.

Informally, the algorithm simply enumerates all the possible CQs over $\mathcal{O}$ with at most $bound(\mathcal{M}, q_{\mathcal{S}})$ atoms occurring in their bodies and possibly involving constants occurring in $q_{\mathcal{S}}$ or in $\mathcal{M}$ as terms. Then, it returns the union of all and only the ones that are sound J-abstractions of $q_{\mathcal{S}}$.

Example 7.4. Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \emptyset$
- $\mathcal{S} = \{\ s_1, s_2, s_3, s_4, s_5\ \}$
- $\mathcal{M} = \{\ m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8\ \}$, where:

$$\begin{array}{lrcl}
m_1: & s_1(x_1, x_2) & \to & P_1(x_1, x_2), \\
m_2: & s_3(x_1, x_2) & \to & P_1(x_1, x_2), \\
m_3: & \exists y. s_2(x, y) & \to & A_1(x), \\
m_4: & s_4(x, c_3) & \to & A_1(x), \\
m_5: & s_1(x_1, x_2) \wedge s_3(x_1, x_2) & \to & P_2(x_1, x_2), \\
m_6: & \exists y. s_5(x_1, x_2) \wedge s_2(x_2, y) \wedge s_4(x_2, y) & \to & P_2(x_1, x_2), \\
m_7: & s_1(x, x) \wedge s_2(x, c_2) & \to & A_2(x), \\
m_8: & s_3(x_1, c_1) \wedge s_4(c_1, x_2) & \to & P_3(x_1, x_2).
\end{array}$$

For the UCQJFE $q_{\mathcal{S}} = \{(x_1, x_2) \mid \exists y. s_1(x_1, x_2) \wedge s_2(x_2, y)\} \ \cup\ \{(x_1, x_2) \mid \exists y. s_3(x_1, x_2) \wedge s_4(x_2, y)\}$, the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ is the query $q_{\mathcal{O}} = \{(x_1, x_2) \mid P_1(x_1, x_2) \wedge A_1(x_2) \wedge P_2(x_1, x_2)\} \ \cup\ \{(x, x) \mid A_2(x)\} \ \cup\ \{(x, c_1) \mid \exists y. P_3(x, y)\}$. Indeed one can verify that, on the one hand, each disjunct of $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$, and, on the other hand, each possible CQ q' over $\mathcal{O}$ being a sound J-abstraction of $q_{\mathcal{S}}$ is such that $cert_{q',J} \sqsubseteq cert_{q,J}$ for some disjunct q of $q_{\mathcal{O}}$.

Furthermore, one can easily check that MaximallySoundForUCQJFEs$(J, q_{\mathcal{S}})$ returns a UCQ which is equivalent (w.r.t. J) to $q_{\mathcal{O}}$, in fact it contains all the disjuncts of $q_{\mathcal{O}}$. △

The following theorem establishes termination and correctness of the MaximallySoundForUCQJFEs algorithm.

Theorem 7.2. *In the restricted scenario, MaximallySoundForUCQJFEs $(J, q_{\mathcal{S}})$ terminates and returns the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$.*

Proof. Termination of the algorithm follows from the fact that it just enumerates all possible CQs over $\mathcal{O}$ with a certain bound on the number of atoms occurring in their bodies, and involving only constants occurring in $q_{\mathcal{S}}$ or in $\mathcal{M}$.

As for the correctness, we first point out that the computed UCQ $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$. Indeed $q_{\mathcal{O}}$ is a UCQ whose disjuncts are sound J-abstractions of $q_{\mathcal{S}}$ by construction. We now show that $q_{\mathcal{O}}$ is actually the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$, that is, each UCQ $q'_{\mathcal{O}}$ that is a sound J-abstraction of $q_{\mathcal{S}}$ is such that $cert_{q'_{\mathcal{O}},J} \sqsubseteq cert_{q_{\mathcal{O}},J}$ (cf. Definition 3.4). We do this by way of contradiction.

Let $q'_{\mathcal{O}}$ be a UCQ such that $cert_{q'_{\mathcal{O}},J} \not\sqsubseteq cert_{q_{\mathcal{O}},J}$, that is, there exists an $\mathcal{S}$-database D consistent with J such that $cert^D_{q'_{\mathcal{O}},J} \not\subseteq cert^D_{q_{\mathcal{O}},J}$. It follows that there is a tuple of constants $\vec{c} = (c_1, \ldots, c_n)$ such that $\vec{c} \in cert^D_{q'_{\mathcal{O}},J}$ but, at the same time, $\vec{c} \notin cert^D_{q_{\mathcal{O}},J}$. Consider now $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$, i.e., the canonical structure of $\mathcal{O}$ with respect to $\mathcal{M}$ and D. Notice that, since $\mathcal{M}$ is a GAV mapping and $\mathcal{O}$ is a *DL-Lite*$_{\mathsf{RDFS}}$ ontology, we have that: (i) $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ does not introduce variables, and therefore we can see it as a set of facts $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)} = \{\beta_1, \ldots, \beta_m\}$ over $\mathcal{O}$; and (ii) $cert^D_{q_{\mathcal{O}},J} = q_{\mathcal{O}}^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$ for each $\mathcal{S}$-database D. We now exhibit an $\mathcal{S}$-database D' for which (i) $\vec{c} \notin q_{\mathcal{S}}^{D'}$, and (ii) $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)} \subseteq \mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$. To this aim, we exploit the boolean query q_β over $\mathcal{O}$ associated to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$, i.e., the following boolean CQ:

$$q_\beta = \{() \mid \beta_1 \wedge \ldots \wedge \beta_m\},$$

and all its possible unfoldings r over $\mathcal{S}$. In particular, there are two possible cases: either every disjunct r of $\textsf{PerfRef}_{q_\beta,J}$ is such that $\vec{c} \in q_{\mathcal{S}}^{D_r}$, or not. We recall that, for a CQ r over $\mathcal{S}$, D_r denotes the $\mathcal{S}$-database associated to r, i.e., the set of facts over $\mathcal{S}$ occurring in the body of r in which each existential variable v is replaced by a different fresh constant c_v.

In the former case, let q be the CQ over $\mathcal{O}$ in which the target list is initially $\vec{c} = (c_1, \ldots, c_n)$ and the body is the same as q_β, i.e., $q = \{\vec{c} \mid \beta_1 \wedge \ldots \wedge \beta_n\}$. Then, consider the following changes to q: for each constant c occurring in q (either in the body or in the target list) but occurring neither in $q_{\mathcal{S}}$ nor in $\mathcal{M}$, replace c everywhere (even in the target list) by a distinguished variable x_c if $c_i = c$ for some $i \in [1, n]$ (i.e., if c occurs in the target list of q), and by an existential variable y_c otherwise.

Obviously, by construction, we have $\vec{c} \in q^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$. Furthermore, due to the fact that $\mathcal{M}$ is a pure GAV mapping and the fact that $\mathcal{O}$ contains no assertions with $\exists R$ in the right-hand side for a basic role R, each possible disjunct r_q of $\textsf{PerfRef}_{q,J}$ has a corresponding disjunct r of $\textsf{PerfRef}_{q_\beta,J}$ in which the body of r is obtained from the body of r_q by replacing the distinguished variables x_c (respectively, the existential variable y_c) occurring in q with constant c. Notice that, by assumption, each disjunct r of $\textsf{PerfRef}_{q_\beta,J}$ is such that $\vec{c} \in q_{\mathcal{S}}^{D_r}$, i.e., for each disjunct r of $\textsf{PerfRef}_{q_\beta,J}$ there is a disjunct $q'_{\mathcal{S}}$ of $q_{\mathcal{S}}$ for which $\vec{c} \in {q'_{\mathcal{S}}}^{D_r}$. Since $q_{\mathcal{S}}$ is a UCQJFE, by exploiting Lemma 7.1, it is not hard to ascertain that this implies that for each disjunct r of $\textsf{PerfRef}_{q_\beta,J}$ there is a disjunct $q'_{\mathcal{S}}$ of $q_{\mathcal{S}}$ for which for each atom α of $q'_{\mathcal{S}}$ there is an atom of r that instantiates α. By construction of q, however, the above property holds even if we replace q_β with q, i.e., for each disjunct r_q of $\textsf{PerfRef}_{q,J}$ there is a disjunct $q'_{\mathcal{S}}$ of $q_{\mathcal{S}}$ for which for each atom α of $q'_{\mathcal{S}}$ there is an atom of r_q that instantiates α. Thus, by Lemma 7.1, we derive that q is a sound J-abstraction of $q_{\mathcal{S}}$. But then, due to Lemma 7.3, from q it is possible to derive a CQ q' with same target list as q but whose body is the conjunction of at most $\mathit{bound}(\mathcal{M}, q_{\mathcal{S}})$ atoms occurring in q and such that q' is a sound J-abstraction of $q_{\mathcal{S}}$. By construction of the algorithm, however, it can be readily seen that such a CQ q' is a disjunct of $q_{\mathcal{O}}$. Two considerations are now in order: (i) since q' has the same target list as q, and since its body is constituted only by a subset of the atoms of q, the fact that $\vec{c} \in q^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$ implies $\vec{c} \in {q'}^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$ as well, and (ii) since $\vec{c} \in {q'}^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$ and since q' is a disjunct of $q_{\mathcal{O}}$, we have $\vec{c} \in {q_{\mathcal{O}}}^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$. Thus, since in this setting for OBDM specifications $\mathit{cert}_{q_{\mathcal{O}},J}^{D} = q_{\mathcal{O}}^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$ for each UCQ $q_{\mathcal{O}}$ and $\mathcal{S}$-database D, we derive that $\vec{c} \in \mathit{cert}_{q_{\mathcal{O}},J}^{D}$, which is a contradiction on the initial assumption that $\vec{c} \notin \mathit{cert}_{q_{\mathcal{O}},J}^{D}$. It follows that the former case just considered is not possible because it leads to a contradiction. Therefore, we consider only the latter case.

Consider the latter case, i.e., there is a disjunct r of $\textsf{PerfRef}_{q_\beta,J}$ for which $\vec{c} \notin q_{\mathcal{S}}^{D_r}$. Consider $D' = D_r$. Since $\mathcal{M}$ is a pure GAV mapping and $\mathcal{O}$ is a *DL-Lite*$_{\textsf{RDFS}}$ ontology, and thus contains no assertions with $\exists R$ in the right-hand side for a basic role R, and since r is a disjunct of $\textsf{PerfRef}_{q_\beta,J}$ (i.e., the body of r is a way for unfolding all the facts occurring in $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$), it is easy to verify that $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ is such that $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)} \subseteq \mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$. Notice that $\vec{c} \in \mathit{cert}_{q'_{\mathcal{O}},J}^{D}$ holds by assumption, and therefore $\vec{c} \in {q'_{\mathcal{O}}}^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$. Furthermore, since $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)} \subseteq \mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ and $q_{\mathcal{O}}$ is a UCQ, we trivially derive that $\vec{c} \in {q'_{\mathcal{O}}}^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}}$ as well, which, in turn, implies that $\vec{c} \in \mathit{cert}_{q'_{\mathcal{O}},J}^{D'}$.

To complete the proof, consider the $\mathcal{S}$-database D'. We have that, on the one hand, $\vec{c} \notin q_{\mathcal{S}}^{D'}$, and, on the other hand, $\vec{c} \in \mathit{cert}_{q'_{\mathcal{O}},J}^{D'}$. It follows that $q'_{\mathcal{O}}$ is not a sound J-abstraction of $q_{\mathcal{S}}$, as required. □

Regarding the cost of the algorithm, we observe that the overall running time is exponential in the size of the input. Notice, moreover, that CQs over $\mathcal{O}$ may have an exponential number of atoms with respect to $\eta(q_{\mathcal{S}})$. Next we prove that (i) unless PTIME = NP, the computation problem for sound abstractions cannot be solved in polynomial time, even in the restricted scenario; and (ii) there exist OBDM specifications J and UCQJFEs $q_{\mathcal{S}}$ for which the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ is a CQ whose number of atoms is necessarily exponential with respect to $\eta(q_{\mathcal{S}})$.

Proposition 7.1. *There exists an OBDM specification J in the restriced scenario and a CQJFE $q_{\mathcal{S}}$ such that, assuming* PTIME ⊂ *NP, the UCQ-minimally sound J-abstraction of $q_{\mathcal{S}}$ cannot be computed in polynomial time.*

Proof. Let F be a 3-DNF formula on a set of variables $X = \{x_1, \ldots, x_n\}$. Consider the OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, the UCQJFE $q_{\mathcal{S}}$, and the CQ $q_{\mathcal{O}}$ constructed from F as illustrated in the reduction of the lower bound proof of Theorem 7.1.

In this case, it is easy to verify that the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ is either the CQ $q_{\mathcal{O}}$ if it is a sound J-abstraction of $q_{\mathcal{S}}$, or the CQ $q'_{\mathcal{O}} = \{(x_1, \ldots, x_n) \mid \bot(x_1) \wedge \ldots \wedge \bot(x_n)\}$ otherwise. Specifically, as shown in that coNP-hardness proof of Theorem 7.1, $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$ if and only if formula F is valid. So, due to the above observation, $q_{\mathcal{O}}$ is the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ if and only if formula F is valid.

We have therefore reduced the problem of checking the validity of a 3-DNF formula F to the problem of computing the UCQ-maximally sound J-abstraction of a UCQJFE $q_{\mathcal{S}}$, where both $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and $q_{\mathcal{S}}$ can be constructed in LOGSPACE from F.

Thus, a polynomial time algorithm for computing UCQ-maximally sound abstractions in the restricted scenario would imply a polynomial time algorithm for checking whether a 3-DNF formula is valid. Since this latter problem is known to be in general coNP-hard, it follows that, unless PTIME = NP, the computation problem for sound abstractions cannot be solved in polynomial time in the restricted scenario. □

Proposition 7.2. *In the restricted scenario, there are OBDM specifications J and UCQJFEs $q_{\mathcal{S}}$ for which the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ is a CQ whose number of atoms in its body is necessarily exponential with respect to $\eta(q_{\mathcal{S}})$.*

Proof. We provide here a small example showing the main reason of why the number of atoms in the body of the UCQ-maximally sound J-abstraction of a UCQJFE $q_{\mathcal{S}}$ may be exponential with respect to $\eta(q_{\mathcal{S}})$.

Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \emptyset$
- $\mathcal{S} = \{\ s_1, s_2, s_3, s_4\ \}$
- $\mathcal{M} = \{\ m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}\ \}$, where:

$$\begin{aligned}
m_1 &: & s_1(x) &\rightarrow & A_1(x),\\
m_2 &: & s_2(x) &\rightarrow & A_1(x),\\
m_3 &: & s_1(x) &\rightarrow & A_2(x),\\
m_4 &: & s_3(x) &\rightarrow & A_2(x),\\
m_5 &: & s_1(x) &\rightarrow & A_3(x),\\
m_6 &: & s_4(x) &\rightarrow & A_3(x),\\
m_7 &: & s_2(x) &\rightarrow & A_4(x),\\
m_8 &: & s_3(x) &\rightarrow & A_4(x),\\
m_9 &: & s_2(x) &\rightarrow & A_5(x),\\
m_{10} &: & s_4(x) &\rightarrow & A_5(x),\\
m_{11} &: & s_3(x) &\rightarrow & A_6(x),\\
m_{12} &: & s_4(x) &\rightarrow & A_6(x).
\end{aligned}$$

Let $q_{\mathcal{S}}$ be the following UCQJFE over $\mathcal{S}$: $q_{\mathcal{S}} = \{(x) \mid s_1(x) \wedge s_2(x) \wedge s_3(x)\} \cup \{(x) \mid s_1(x) \wedge s_2(x) \wedge s_4(x)\} \cup \{(x) \mid s_1(x) \wedge s_3(x) \wedge s_4(x)\} \cup \{(x) \mid s_2(x) \wedge s_3(x) \wedge s_4(x)\}$. One can verify that the CQ $q_{\mathcal{O}} = \{(x) \mid A_1(x) \wedge A_2(x) \wedge A_3(x) \wedge A_4(x) \wedge A_5(x) \wedge A_6(x)\}$ is a sound J-abstraction of $q_{\mathcal{S}}$, and, moreover, every possible CQ $q'_{\mathcal{O}}$ whose body contains only a strict subset of the atoms occurring in the body of $q_{\mathcal{O}}$ is such that $q'_{\mathcal{O}}$ is not a sound J-abstraction of $q_{\mathcal{S}}$. Thus, it follows that $q_{\mathcal{O}}$ is the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ (in fact, one can verify that $q_{\mathcal{O}}$ is the output of MaximallySoundForUCQJFEs$(J, q_{\mathcal{S}})$). More precisely, one can verify that $q_{\mathcal{O}}$ is the perfect J-abstraction of $q_{\mathcal{S}}$.

Let now denote by $|\mathcal{S}|$ the number of source predicates occurring in the source schema $\mathcal{S}$, and by $\chi(\mathcal{M})$ the number of times that an atomic concept A_i in the alphabet of the ontology $\mathcal{O}$ appears in the right-hand side of mapping assertions in $\mathcal{M}$. By generalizing the above construction, one can see that it is always possible to compose OBDM specifications $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and UCQJFEs $q_{\mathcal{S}}$ for which (i) $\eta(q_{\mathcal{S}}) = |\mathcal{S}|^2 - |\mathcal{S}|$, and (ii) the number of atoms occurring in the body of the CQ corresponding to the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ is necessarily equal to $\frac{|\mathcal{S}|!}{(|\mathcal{S}|-\chi(\mathcal{M}))! \cdot \chi(\mathcal{M})!}$ (and therefore, an exponential number of atoms with respect to $\eta(q_{\mathcal{S}})$). □

7.4. The restricted scenario for CQJFEs

We next discuss results about the verification and the computation problems for sound abstractions in a scenario that is further restricted with respect to the one considered so far, i.e., by limiting the query language $\mathcal{L}_{\mathcal{S}}$ to CQJFEs. Interestingly, with such a further limitation, the complexity of the verification problem goes down from coNP-complete to PTIME, while an algorithm for computation can be devised that is "smarter", even though exponential in the worst-case. We next provide an intuition for both these results. The interested reader can refer to Appendix B for more details.

As for verification, the complexity lowering for CQJFEs is due to the fact that in order to check whether $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$, one needs to check whether each disjunct q' of $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$, and if the latter is a CQJFE, this check can be done by verifying that for each atom α of $q_{\mathcal{S}}$ there exists an atom β of q' that *J-covers* α, i.e., that is such that every disjunct in $\rho(\beta, J)$ contains at least an atom α' that instantiates α.

As for computing the UCQ-maximally sound J-abstraction, if $q_{\mathcal{S}}$ is a CQJFE rather than UCQJFE, instead of enumerating all possible CQs over $\mathcal{O}$ of a certain bound, the computation is "guided" by the atoms occurring in the input query $q_{\mathcal{S}}$, in a way that is very similar to the *bucket algorithm* [64] used for rewriting queries using views. In particular, by exploiting the aforementioned characterization of sound J-abstractions of $q_{\mathcal{S}}$, for each atom α of $q_{\mathcal{S}}$, the algorithm computes a set B_α containing all the atoms β over $\mathcal{O}$ such that β *J-cover* α. Then, the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ is computed as the union of all queries that can be obtained by conjoining one atom from B_α, for every α in $q_\mathcal{S}$. Note, in particular, that it can be shown that, differently from the case of UCQJFEs, all such queries contain at most only $\eta(q_\mathcal{S})$ atoms.

8. Conclusion

We have presented a formal framework for a new OBDM reasoning task, namely the task of automatically computing semantic characterizations of data services through ontologies, called Abstraction. In particular, we have carried out a systematic and comprehensive analysis for the most common OBDM setting, including a restricted setting, still useful in practice.

We believe that the notions introduced and the technical results presented in this paper are not only theoretically interesting in themselves, but also have many possible practical applications besides making data services automatically Findable, Accessible, Interoperable, and Reusable (FAIR), as for example in the fields mentioned in the introduction, namely open data, source profiling, updating, and explanation of classifiers.

We point out that this work left some interesting and challenging open problems. In the following we detail the main ones:

- a relevant problem not addressed in this paper is the *Existence* problem, that is: given an OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and a query $q_\mathcal{S} \in \mathcal{L}_\mathcal{S}$, verify whether there exists an $\mathcal{L}_\mathcal{O}$-perfect (respectively, $\mathcal{L}_\mathcal{O}$-minimally complete, $\mathcal{L}_\mathcal{O}$-maximally sound) J-abstraction of $q_\mathcal{S}$. Some results for this problem can be immediately derived from the results provided in this paper for the general scenario considered, namely: (i) the Existence problem is trivial for the minimally complete case because, as Corollary 4.2 states, the UCQ-minimally complete J-abstraction of a UCQ $q_\mathcal{S}$ always exist; (ii) from Theorem 6.2, we immediately get decidability of the Existence problem for the perfect case. However, for the general scenario considered in this paper, we point out that the exact computational complexity for the perfect case and even decidability for the maximally sound case are interesting open challenges;
- still for the general scenario, we aim at singling out the minimal class $\mathcal{L}_\mathcal{O}$ of queries that guarantees the existence of an $\mathcal{L}_\mathcal{O}$-maximally sound abstraction of a query $q_\mathcal{S}$. Furthermore, we will extend our analysis to OBDM settings going beyond the one based on *DL-Lite*$_\mathcal{R}$, for example by considering *DL-Lite*$_\mathcal{A}$, the $\mathcal{EL}$ family, and/or other DLs as ontology languages;
- for the case of LAV mappings, the exact computational complexity of the verification problem for perfect abstractions is still open;
- in this paper we assume that integrity constraints are not specified over the sources; it would be interesting to extend our investigation to the case in which they were specified.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

This work has been partially supported by MUR under the PNRR project FAIR (PE0000013) and under the PRIN 2017 project HOPE (prot. 2017MMJJRE), and by the EU under the H2020-EU.2.1.1 project TAILOR (grant id. 952215). Gianluca Cima has been fully supported by MUR under the PNRR project FAIR (PE0000013).

Appendix A. Rest of the proof of Theorem 5.2 (cases 3, 4, and 5)

Case 3. Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \{ \ A \sqsubseteq \exists P \ \}$
- $\mathcal{S} = \{ \ s_1, s_2 \ \}$
- $\mathcal{M} = \{ \ m_1, m_2 \ \}$, where:

$$\begin{aligned} m_1\colon \quad s_1(x_1, x_2) &\quad\rightarrow\quad P(x_1, x_2), \\ m_2\colon \quad s_2(x) &\quad\rightarrow\quad A(x), \end{aligned}$$

and $q_\mathcal{S}$ be the following CQJFE over $\mathcal{S}$: $q_\mathcal{S} = \{(x) \mid \exists y.s_1(x, y)\}$.

Now, suppose that there exists a Q that is a UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$. Also, let $a, b \in \mathsf{Const}$ be two constants not occurring in Q (note that since Q is finite and Const is infinite, one such pair of constants always exists). The query $Q' = Q \cup q'$, where $q' = \{(a) \mid P(a, b)\}$, is certainly a sound J-abstraction of $q_{\mathcal{S}}$ since Q is a sound J-abstraction of $q_{\mathcal{S}}$ and $\mathsf{PerfRef}_{q',J} = \{(a) \mid s_1(a, b)\} \sqsubseteq q_{\mathcal{S}}$. Now, we have two possibilities: either $cert_{q',J} \sqsubseteq cert_{Q,J}$ or $cert_{q',J} \not\sqsubseteq cert_{Q,J}$. We next show that in both cases we get a contradiction. Let us first assume that $cert_{q',J} \not\sqsubseteq cert_{Q,J}$. Then, there exists an $\mathcal{S}$-database D such that $cert^D_{q',J} \not\subseteq cert^D_{Q,J}$, and hence, $cert^D_{Q,J} \subset cert^D_{Q',J}$. But then, this contradicts that Q is a UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$. Let us now assume that $cert_{q',J} \sqsubseteq cert_{Q,J}$. Then, $\mathsf{PerfRef}_{q',J} \sqsubseteq \mathsf{PerfRef}_{Q,J}$, which implies that there exists a CQ q'' in $\mathsf{PerfRef}_{Q,J}$ such that there exists a homomorphism from q'' to $\mathsf{PerfRef}_{q',J} = \{(a) \mid s_1(a, b)\}$. But then, by definition of homomorphism and since Q, by hypothesis, does not contain a and b, this implies that q'' is of the form $\{(x) \mid \exists y.s_1(x, y) \wedge \gamma\}$, where γ is a conjunction of atoms having predicate name s_1 and not mentioning y as first argument. It has not hard to verify that, in this case, Q must contain at least a disjunct of the form $\{(x) \mid \exists y.P(x, y) \wedge \gamma'\}$, where γ' is a conjunction of atoms having predicate name P and not mentioning y as first argument. This immediately contradicts the fact that Q is a UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$.

Case 4. The proof is based on the idea that, in certain cases, an assertion of the form $A \sqsubseteq \exists P$ logically implied by a *DL-Lite*$_{\mathcal{R}}$ ontology $\mathcal{O}$ can be simulated using LAV mapping assertions. In our case, consider the OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ illustrated in the proof of case 3, and let $J' = \langle \mathcal{O}', \mathcal{S}, \mathcal{M}'\rangle$ be the OBDM specification in which $\mathcal{O}' = \emptyset$ ($\mathcal{O}'$ and $\mathcal{O}$ share the same alphabet) and $\mathcal{M}' = \mathcal{M} \cup \{\ m_3 : s_2(x) \rightarrow \exists y.P(x, y)\ \}$. It can be readily seen that J' is *query-preserving with respect to* J [65,23], that is, $cert^D_{q,J} = cert^D_{q,J'}$ for every query q over $\mathcal{O}$ (equivalently, over $\mathcal{O}'$) and for every $\mathcal{S}$-database D. Therefore, a formal proof of this case can be obtained from the proof of case 3 by replacing the OBDM specification J with the OBDM specification J'.

Case 5. Let $\Sigma = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ be the following OBDM specification:

- $\mathcal{O} = \emptyset$
- $\mathcal{S} = \{\ s_1, s_2\ \}$
- $\mathcal{M} = \{\ m_1, m_2\ \}$, where:

$$\begin{aligned} m_1\!: \quad s_1(x_1, x_2) \quad &\rightarrow \quad P(x_1, x_2),\\ m_2\!: \quad s_2(x) \quad &\rightarrow \quad P(x, x). \end{aligned}$$

Let $q_{\mathcal{S}}$ be the following boolean CQJFE over $\mathcal{S}$: $q_{\mathcal{S}} = \{(x_1, x_2) \mid s_1(x_1, x_2)\}$.

Now, suppose that there exists a UCQ Q that is a UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$. Also, let $a, b \in \mathsf{Const}$ be two constants not occurring in Q such that $a \neq b$ (note that since Q is finite and Const is infinite, one such pair of constants always exists). Similarly to the case of 3, the query $Q' = Q \cup q'$, where $q' = \{(a, b) \mid P(a, b)\}$, is a UCQ that is a sound J-abstraction of $q_{\mathcal{S}}$ and is such that assuming either that $cert_{q',J} \sqsubseteq cert_{Q,J}$ or that $cert_{q',J} \not\sqsubseteq cert_{Q,J}$, would both lead to a contradiction.

Appendix B. Sound abstractions in the restricted scenario for CQJFEs

We next provide more details on the results mentioned in Subsection 7.4 about the verification and the computation problem for sound abstractions in the restricted scenario for CQJFEs. We remind the reader that the setting for OBDM specifications considered is obtained from the general one by (i) limiting ontologies $\mathcal{O}$ to be expressed in *DL-Lite*$_{\mathsf{RDFS}}$, and (ii) limiting the mappings $\mathcal{M}$ to be expressed as a set of pure GAV mapping assertions. However, now the query language $\mathcal{L}_{\mathcal{S}}$ is the one of CQJFEs rather than UCQJFEs.

Before going into details, we introduce the notion of *covering*.

Definition B.1. Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ be an OBDM specification, β be an atom over $\mathcal{O}$, and α be an atom occurring in the body of a CQ $q_{\mathcal{S}} = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ over $\mathcal{S}$. We say that β *J-covers* α, if the following holds: in each disjunct of $\rho(\beta, J)$ there is at least an atom α' such that α' instantiates α.

Example B.1. Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ be the following OBDM specification:

- $\mathcal{O} = \{\ P_1 \sqsubseteq P_2\ \}$
- $\mathcal{S} = \{\ s_1, s_2\ \}$
- $\mathcal{M} = \{\ m_1, m_2\ \}$, where:

$$\begin{aligned} m_1\!: \quad s_1(x_1, x_2, x_1) \wedge s_2(x_1, x_2) \quad &\rightarrow \quad P_1(x_1, x_2),\\ m_2\!: \quad s_1(x_1, x_2, c_1) \wedge s_2(x_2, x_2) \quad &\rightarrow \quad P_2(x_1, x_2). \end{aligned}$$

Consider the query $q_{\mathcal{S}} = \{(x) \mid \exists y.s_1(c_2, x, y) \wedge s_2(x, x)\}$ over $\mathcal{S}$, and the atom $\beta = P_2(c_2, x)$ over $\mathcal{O}$. Let $\alpha_1 = s_1(c_2, x, y)$ and $\alpha_2 = s_2(x, x)$. Note that $\rho(\beta, J) = (s_1(c_2, x, c_1) \wedge s_2(x, x)) \vee (s_1(c_2, x, c_2) \wedge s_2(c_2, x))$. Thus, we have that β J-covers α_1, whereas β does not J-cover α_2. This latter is because in the disjunct $(s_1(c_2, x, c_2) \wedge s_2(c_2, x))$ of $\rho(\beta, J)$ there is no atom α' such that α' instantiates α_2. $\triangle$

Clearly, for an OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, an atom β over $\mathcal{O}$, and an atom α over $\mathcal{S}$, checking whether β J-covers α is feasible in polynomial time.

B.1. Verification

We start by proving the following lemma, which will be used to prove that the verification problem in this setting can be solved in polynomial time.

Lemma B.1. *Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification, and let $q_{\mathcal{S}}$ and $q_{\mathcal{O}}$ be a CQJFE over $\mathcal{S}$ and a CQ over $\mathcal{O}$, respectively, with the same target list. We have that $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$ if and only if it is the case that for each atom α of $q_{\mathcal{S}}$ there exists an atom β of $q_{\mathcal{O}}$ such that β J-covers α.*

Proof. "**Only-if part:**" Suppose, for the sake of contradiction, that there exists an atom α of $q_{\mathcal{S}}$ such that for no atom β of $q_{\mathcal{O}}$ it is the case that β J-covers α. Let q' be the disjunct of $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ obtained by unfolding each atom β of $q_{\mathcal{O}}$ with the disjunct of $\rho(\beta, J)$ in which there is no atom α' that instantiates α. For each atom β of $q_{\mathcal{O}}$, there is at least one disjunct among the ones of $\rho(\beta, J)$ that satisfies this condition, otherwise, following Definition B.1, we would trivially derive a contradiction on the fact that β does not J-cover α.

Thus, the disjunct q' of $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ contains no atom that instantiates the atom α of $q_{\mathcal{S}}$, and therefore, due to Lemma 7.1, this implies that $q' \not\sqsubseteq q_{\mathcal{S}}$. It follows that $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \not\sqsubseteq q_{\mathcal{S}}$, which, since $\mathcal{V}_{\mathcal{O}} = \{() \mid \exists y.\bot(y)\}$ and due to Lemma 5.1, in turn implies that $q_{\mathcal{O}}$ is not a sound J-abstraction of $q_{\mathcal{S}}$, as required.

"**If part:**" Since for each atom α of $q_{\mathcal{S}}$ there is an atom β such that β J-covers α, we derive that each possible disjunct q' of $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$ satisfies the following condition: for each atom α of $q_{\mathcal{S}}$, there is an atom α' of q' such that α' instantiates α. Due to Lemma 7.1, it follows that $q' \sqsubseteq q$. Since this is true for each possible disjunct q' of $\mathsf{PerfRef}_{q_{\mathcal{O}},J}$, we also derive that $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \sqsubseteq q_{\mathcal{S}}$, which, since *DL-Lite*$_{\mathsf{RDFS}}$ does not allow for disjointness assertions, implies that $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$. $\square$

Based on the above lemma, the following theorem proves that, in the restricted scenario for CQJFEs, the verification problem for sound abstractions can be solved in polynomial time.

Theorem B.1. *In the restricted scenario for CQJFEs, the verification problem is in* PTIME.

Proof. Since *DL-Lite*$_{\mathsf{RDFS}}$ does not allow for disjointness assertions, it is sufficient to show how to check the containment $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \sqsubseteq q_{\mathcal{S}}$ in polynomial time, where $q_{\mathcal{O}}$ and $q_{\mathcal{S}}$ are a UCQ over $\mathcal{O}$ and a CQJFE over $\mathcal{S}$, respectively. To start, note that by construction $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$ if and only if each disjunct q of $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$, i.e., $\mathsf{PerfRef}_{q_{\mathcal{O}},J} \sqsubseteq q_{\mathcal{S}}$ if and only if $\mathsf{PerfRef}_{q,J} \sqsubseteq q_{\mathcal{S}}$ for each disjunct q of $q_{\mathcal{O}}$. It is therefore enough to show that, given a CQ $q = \{\vec{t'} \mid \exists \vec{y'}.\phi'(\vec{x'}, \vec{y'})\}$ over $\mathcal{O}$ and a CQJFE $q_{\mathcal{S}} = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ over $\mathcal{S}$, checking whether $\mathsf{PerfRef}_{q,J} \sqsubseteq q_{\mathcal{S}}$ can be done in polynomial time.

We assume that every atom β of q appears in the right-hand side of some mapping assertion in $\mathcal{M}$, otherwise we trivially have that $\mathsf{PerfRef}_{q,J} \equiv \mathtt{false}/n$ (where n is the arity of q), and therefore q is a sound J-abstraction of $q_{\mathcal{S}}$. Furthermore, if q and $q_{\mathcal{S}}$ do not have the same target list, i.e., $\vec{t'} = (t'_1, \ldots, t'_n) \neq \vec{t} = (t_1, \ldots, t_n)$, then consider the function f from the set of terms in the target list of q to the set of terms in the target list of $q_{\mathcal{S}}$ with $f(t_i) = t'_i$, for each $i \in [1, n]$. Formally, since repetitions of terms in target lists is allowed, f might give rise to a multivalued function. In this case, as well as in the case that $f(a) = b$ with $a \neq b$ for two constants $a \in \vec{t}$ and $b \in \vec{t'}$, it is straightforward to verify that $\mathsf{PerfRef}_{q,J} \not\sqsubseteq q_{\mathcal{S}}$ trivially holds. Indeed, in those cases there can be no homomorphism from $q_{\mathcal{S}}$ to the disjuncts of $\mathsf{PerfRef}_{q,J}$ by construction.

Consider the query $q'_{\mathcal{S}}$ obtained in polynomial time from $q_{\mathcal{S}}$ by replacing every occurrence of the term t_i in $q_{\mathcal{S}}$ (even in the target list) with the term $f(t_i) = t'_i$, for each $i \in [1, n]$. Observe that now $q'_{\mathcal{S}}$ is a CQJFE having the same target list $\vec{t'}$ of q. By virtue of Lemma B.1, we can now check whether q is a sound J-abstraction of $q'_{\mathcal{S}}$ by checking whether it is the case that for each atom α of q' there exists an atom β of q such that β J-covers α. This can be clearly done in polynomial time.

Obviously, if q is a sound J-abstraction of $q'_{\mathcal{S}}$, then we trivially have that q is sound J-abstraction of $q_{\mathcal{S}}$ as well. Conversely, if q is not a sound J-abstraction of $q'_{\mathcal{S}}$, then there is a disjunct r of $\mathsf{PerfRef}_{q,J}$ such that $r \not\sqsubseteq q'$. But then, it can be readily seen that D_r (i.e., the freezing of r) is the database witnessing that $r \not\sqsubseteq q_{\mathcal{S}}$. It follows that, for the $\mathcal{S}$-database D_r, we have $cert_{q,J}^{D_r} \not\subseteq q_{\mathcal{S}}^{D_r}$, which implies that q is not a sound J-abstraction of $q_{\mathcal{S}}$.

From the above considerations, it is immediate to derive a polynomial time algorithm for checking whether a UCQ $q_{\mathcal{O}}$ over $\mathcal{O}$ is a sound J-abstraction of a CQJFE $q_{\mathcal{S}}$ over $\mathcal{S}$. $\square$

B.2. Computation

As for the computation problem, we now provide an algorithm for computing UCQ-maximally sound abstractions which avoids the enumeration of all possible CQs over $\mathcal{O}$ of a certain bound as algorithm MaximallySoundForUCQJFEs does. The computation of the returned UCQ over $\mathcal{O}$ is rather guided by the atoms occurring in the input query $q_{\mathcal{S}}$, in a very similar fashion to the *bucket algorithm* [64] used for rewriting queries using views.

Specifically, by exploiting Lemma B.1, the idea is as follows: for each atom α_i occurring in the body of $q_{\mathcal{S}}$, we compute a set B_i containing all the atoms β over $\mathcal{O}$ such that β J-cover α. Then, disjuncts of the final UCQ $q_{\mathcal{O}}$ over $\mathcal{O}$ are constructed by simply selecting atoms from each set B_i and conjoining them.

There is, however, a preliminary issue to solve in order to apply this simple idea: let $J = \langle\mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ be an OBDM specification, and let α be an atom of a CQ $q_{\mathcal{S}} = \{\vec{t} \mid \exists\vec{y}.\phi(\vec{x},\vec{y})\}$ over $\mathcal{S}$. It might happen that an atom β over $\mathcal{O}$ does not J-cover α, but β J-covers α' if some equalities are applied in the target list of $q_{\mathcal{S}}$, where α' denotes the atom obtained from α after applying such equalities. The next example shows this complication:

Example B.2. Let $J = \langle\mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ be the following OBDM specification:

- $\mathcal{O} = \emptyset$
- $\mathcal{S} = \{\ s\ \}$
- $\mathcal{M} = \{\ m_1, m_2\ \}$, where:

$$\begin{aligned} m_1\!: &\quad s(x,x) &\rightarrow&\quad A(x),\\ m_2\!: &\quad s(x,c_1) &\rightarrow&\quad A'(x). \end{aligned}$$

Consider the CQJFE $q_{\mathcal{S}} = \{(x_1,x_2) \mid s(x_1,x_2)\}$ over $\mathcal{S}$. Observe that there is no atom β over $\mathcal{O}$ such that β J-covers $s(x_1,x_2)$.

Notice, however, that if we consider more specific queries obtained by applying some equalities to the query $q_{\mathcal{S}}$, such as $q^1_{\mathcal{S}} = \{(x_1,x_1) \mid s(x_1,x_1)\}$ (obtained by applying $x_1 = x_2$) and $q^2_{\mathcal{S}} = \{(x_1,c_1) \mid s(x_1,c_1)\}$ (obtained by applying $x_2 = c_1$), then we get that atom $A(x_1)$ J-covers $s(x_1,x_1)$ and atom $A'(x_1)$ J-covers $s(x_1,c_1)$. As a result, due to Lemma B.1, queries $q^1_{\mathcal{O}} = \{(x_1,x_1) \mid A(x_1)\}$ and $q^2_{\mathcal{O}} = \{(x_1,c_1) \mid A'(x_1)\}$ are a sound J-abstraction of $q^1_{\mathcal{S}}$ and $q^2_{\mathcal{S}}$, respectively. It follows that, since by construction $q^i_{\mathcal{S}} \sqsubseteq q_{\mathcal{S}}$ for both $i = 1$ and $i = 2$, both $q^1_{\mathcal{O}}$ and $q^2_{\mathcal{O}}$ are sound J-abstractions of $q_{\mathcal{S}}$ as well.

Furthermore, when applying equalities, we have to take into account not only constants occurring in mapping assertions $\mathcal{M}$, but also constants occurring in the body of the input query. Consider indeed the CQJFE $q'_{\mathcal{S}} = \{(x) \mid s(x,c_2)\}$ over $\mathcal{S}$. Observe that there is no atom β over $\mathcal{O}$ such that β J-covers $s(x,c_2)$. Notice, however, that if we consider the query $q^3_{\mathcal{S}} = \{(c_2) \mid s(c_2,c_2)\}$ (obtained by applying $x = c_2$ to $q'_{\mathcal{S}}$), then we get that atom $A(c_2)$ J-covers $s(c_2,c_2)$. So, due to Lemma B.1, the query $q^3_{\mathcal{O}} = \{(c_2) \mid A(c_2)\}$ is a sound J-abstraction of $q^3_{\mathcal{S}}$, and therefore a sound J-abstraction of $q'_{\mathcal{S}}$ since $q^3_{\mathcal{S}} \sqsubseteq q'_{\mathcal{S}}$. △

Therefore, before of applying the idea described above for computing UCQ-maximally sound abstractions, we first have to compute the head completion of $q_{\mathcal{S}}$ with respect to $\mathsf{con}_{\mathcal{M}} \cup \mathsf{con}_{q_{\mathcal{S}}}$, where $\mathsf{con}_{\mathcal{M}}$ (respectively, $\mathsf{con}_{q_{\mathcal{S}}}$) denote the set of all constants occurring in $\mathcal{M}$ (respectively, $q_{\mathcal{S}}$).

Roughly speaking, the *head completion* of a CQ $q_{\mathcal{S}} = \{\vec{t} \mid \exists\vec{y}.\phi(\vec{x},\vec{y})\}$ with respect to a set of constants con is an equivalent UCQ in which each disjunct is computed by considering a possible unification between the terms in $\vec{t} \cup \mathsf{con}$.

We now present the algorithm HeadCompletion that, given a CQ q and a set of constants con, returns a logically equivalent UCQ representing the head completion of q with respect to con.

In the algorithm, two terms t_1 and t_2 are *compatible* if t_1 and t_2 denote distinct terms and at least one of them is a variable. Furthermore, for a query q, $q[t_1/t_2]$ denotes the query obtained from q by replacing every occurrence (even in the target list) of the term t_1 in q with the term t_2 (if one of the two terms is a constant, then we always assume that t_2 is the constant and t_1 is the variable).

For a CQ q and a set of constants con, HeadCompletion(q, con) computes the equivalent UCQ Q obtained by unifying compatible terms of the target list of q, and of the set of constants con, in all possible ways.

The next example illustrates an execution of the HeadCompletion algorithm.

Example B.3. Let $q_{\mathcal{S}}$ be the following CQ $q_{\mathcal{S}} = \{(x_1,x_2) \mid \exists y.s_1(x_1,c_2,y) \wedge s_2(x_1,x_2)\}$, and let con be the following set of constants $\mathsf{con} = \{c_1,c_2\}$. One can verify that HeadCompletion($q_{\mathcal{S}}$, con) returns the UCQ $Q = \bigcup_{1\leq i\leq 10} q^i_{\mathcal{S}}$, where:

- $q^1_{\mathcal{S}} = \{(x_1,x_2) \mid \exists y.s_1(x_1,c_2,y) \wedge s_2(x_1,x_2)\}$;
- $q^2_{\mathcal{S}} = \{(x_1,x_1) \mid \exists y.s_1(x_1,c_2,y) \wedge s_2(x_1,x_1)\}$;
- $q^3_{\mathcal{S}} = \{(x_1,c_1) \mid \exists y.s_1(x_1,c_2,y) \wedge s_2(x_1,c_1)\}$;

Algorithm HeadCompletion.

```
Input: CQ $q$; set of constants con
Output: UCQ $Q$
$Q := q$
repeat
    $Q' := Q$
    for each CQ $q' \in Q'$ do
        Let $q' = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$
        for each each pair of compatible terms $t_1, t_2$ in $\vec{t} \cup \mathsf{con}$ do
            $Q := Q \cup q'[t_1/t_2]$
        end for
    end for
until $Q' = Q$
return $q_{\mathcal{O}}$
```

Algorithm MaximallySoundForCQJFEs.

```
Input: OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$; CQJFE over $q_{\mathcal{S}}$ over $\mathcal{S}$ of arity $n$
Output: UCQ $q_{\mathcal{O}}$ over $\mathcal{O}$
$q_{\mathcal{O}} := \{(x_1, \ldots, x_n) \mid \bot(x_1) \wedge \ldots \wedge \bot(x_n)\}$
$\mathsf{con} := \mathsf{con}_{\mathcal{M}} \cup \mathsf{con}_{\mathsf{q}_{\mathcal{S}}}$
for each CQ $q \in \mathsf{HeadCompletion}(q_{\mathcal{S}}, \mathsf{con})$ do
    Let $q = \{\vec{t} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$, where $\phi(\vec{x}, \vec{y}) = \alpha_1 \wedge \ldots \wedge \alpha_{\eta(q_{\mathcal{S}})}$
    for each $i \leftarrow 1$ to $\eta(q_{\mathcal{S}})$ do
        $B_i := \emptyset$
        for each possible atoms $\beta$ over $\mathcal{O}$ having as arguments the terms occurring in $\alpha_i$ and possibly fresh existential variables do
            if $\beta$ $J$-covers $\alpha_i$ then
                $B_i := B_i \cup \beta$
            end if
        end for
    end for
    for each combinations of atoms $(\beta_1, \ldots, \beta_{\eta(q_{\mathcal{S}})}) \in B_1 \times \ldots \times B_{\eta(q_{\mathcal{S}})}$ do
        $q_{\mathcal{O}} := q_{\mathcal{O}} \cup \{\vec{t} \mid \exists \vec{y}'.\phi'(\vec{x}, \vec{y}')\}$, where $\phi'(\vec{x}, \vec{y}') = \beta_1 \wedge \ldots \wedge \beta_{\eta(q_{\mathcal{S}})}$
    end for
end for
return $q_{\mathcal{O}}$
```

- $q^4_{\mathcal{S}} = \{(x_1, c_2) \mid \exists y.s_1(x_1, c_2, y) \wedge s_2(x_1, c_2)\}$;
- $q^5_{\mathcal{S}} = \{(c_1, x_2) \mid \exists y.s_1(c_1, c_2, y) \wedge s_2(c_1, x_2)\}$;
- $q^6_{\mathcal{S}} = \{(c_2, x_2) \mid \exists y.s_1(c_2, c_2, y) \wedge s_2(c_2, x_2)\}$;
- $q^7_{\mathcal{S}} = \{(c_1, c_2) \mid \exists y.s_1(c_1, c_2, y) \wedge s_2(c_1, c_2)\}$;
- $q^8_{\mathcal{S}} = \{(c_1, c_1) \mid \exists y.s_1(c_1, c_2, y) \wedge s_2(c_1, c_1)\}$;
- $q^9_{\mathcal{S}} = \{(c_2, c_1) \mid \exists y.s_1(c_2, c_2, y) \wedge s_2(c_2, c_1)\}$;
- $q^{10}_{\mathcal{S}} = \{(c_2, c_2) \mid \exists y.s_1(c_2, c_2, y) \wedge s_2(c_2, c_2)\}$. △

We are now ready to focus on the problem of computing UCQ-maximally sound abstractions in the restricted scenario for CQJFEs, and present algorithm MaximallySoundForCQJFEs.

Informally, the algorithm first computes the head completion of $q_{\mathcal{S}}$ with respect to con, where $\mathsf{con} = \mathsf{con}_{\mathcal{M}} \cup \mathsf{con}_{\mathsf{q}_{\mathcal{S}}}$. Subsequently, for each possible CQ $q \in \mathsf{HeadCompletion}(q_{\mathcal{S}}, \mathsf{con})$, the algorithm proceeds in two main steps. In the first step, for each atom α_i occurring in the body of q, it is computed the set B_i of relevant atoms over $\mathcal{O}$, where β J-covers α_i for each $\beta \in B_i$.

In the second step, for each possible combination which includes a single atom from every set B_i (i.e., for each possible tuple of the Cartesian product $B_1 \times \ldots \times B_{\eta(q_{\mathcal{S}})}$), the CQ with the same target list of q and body the conjunction of those atoms is added as a disjunct of the final returned UCQ $q_{\mathcal{O}}$ over $\mathcal{O}$.

Example B.4. Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \emptyset$
- $\mathcal{S} = \{\ s_1, s_2, s_3, s_4, s_5, s_6\ \}$

- $\mathcal{M} = \{\ m_1, m_2, m_3, m_4, m_5, m_6\ \}$

$$\begin{array}{lrcl} m_1: & \exists y. s_1(x_1, x_2, y) & \rightarrow & P_1(x_1, x_2), \\ m_2: & s_2(x_1, x_2) & \rightarrow & P_2(x_1, x_2), \\ m_3: & s_3(x_1, x_2) & \rightarrow & P_2(x_1, x_2), \\ m_4: & \exists y. s_2(x, x) \wedge s_4(x, y) & \rightarrow & A_1(x), \\ m_5: & \exists y. s_2(x, x) \wedge s_4(x, y) & \rightarrow & A_2(x), \\ m_6: & s_2(x_1, c_1) \wedge s_6(x_1, x_2) & \rightarrow & P_3(x_1, x_2) \end{array}$$

Let $q_{\mathcal{S}}$ be the CQJFE illustrated in Example B.3. As a first step, the algorithm computes $\mathsf{HeadCompletion}(q_{\mathcal{S}}, \mathsf{con})$ which, since $\mathsf{con}_{\mathcal{M}} = \{c_1\}$ and $\mathsf{con}_{\mathsf{q}_{\mathcal{S}}} = \{c_2\}$, it turns out to be the UCQ $Q = \bigcup_{1 \le i \le 10} q^i_{\mathcal{S}}$ illustrated in Example B.3, where $\mathsf{con} = \mathsf{con}_{\mathcal{M}} \cup \mathsf{con}_{\mathsf{q}_{\mathcal{S}}} = \{c_1, c_2\}$.

Then, for each $i \in [1, 10]$, the algorithm processes query $q^i_{\mathcal{S}}$ to add possible CQs that are sound J-abstractions of $q^i_{\mathcal{S}}$ (and therefore of $q_{\mathcal{S}}$). We point out that, for every $j = [1, 4, 5, 6, 7]$, the resulting atom α^j_2 with predicate name s_2 in the body of $q^j_{\mathcal{S}}$ is such that $B^j_2 = \emptyset$, i.e., there is no atom β for which β J-covers α^j_2, and therefore no disjunct is added for those queries.

As for the query $q^2_{\mathcal{S}} = \{(x_1, x_1) \mid \exists y. s_1(x_1, c_2, y) \wedge s_2(x_1, x_1)\}$, we have $B^2_1 = \{P_1(x_1, c_2)\}$ and $B^2_2 = \{A_1(x_1), A_2(x_1)\}$. Thus, the CQs $q^1_{\mathcal{O}} = \{(x_1, x_1) \mid P_1(x_1, c_2) \wedge A_1(x_1)\}$ and $q^2_{\mathcal{O}} = \{(x_1, x_1) \mid P_1(x_1, c_2) \wedge A_1(x_1)\}$ are disjuncts of the final UCQ $q_{\mathcal{O}}$.

As for the query $q^3_{\mathcal{S}} = \{(x_1, c_1) \mid \exists y. s_1(x_1, c_2, y) \wedge s_2(x_1, c_1)\}$, we have $B^3_1 = \{P_1(x_1, c_2)\}$ and $B^3_2 = \{P_3(x, y')\}$. Thus, the CQ $q^3_{\mathcal{O}} = \{(x_1, c_1) \mid \exists y'. P_1(x_1, c_1) \wedge P_3(x_1, y')\}$ is a disjunct of the final UCQ $q_{\mathcal{O}}$.

For the queries $q^j_{\mathcal{S}}$ with $j = [8, 9, 10]$, we observe that all disjuncts over $\mathcal{O}$ generated by the algorithm are subsumed (w.r.t. J) by $q^i_{\mathcal{O}}$ for some $i = [1, 2, 3]$. As a conclusion, one can verify that $\mathsf{MaximallySoundForCQJFEs}(J, q_{\mathcal{S}})$ returns a UCQ that is equivalent (w.r.t. J) to $q_{\mathcal{O}} = q^1_{\mathcal{O}} \cup q^2_{\mathcal{O}} \cup q^3_{\mathcal{O}}$, which is the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$. △

The following theorem establishes termination and correctness of the MaximallySoundForCQJFEs algorithm.

Theorem B.2. *In the restricted scenario for CQJFEs, $\mathsf{MaximallySoundForCQJFEs}$ $(J, q_{\mathcal{S}})$ terminates and returns the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$.*

Proof. Termination of the algorithm easily follows from the termination of the HeadCompletion algorithm, and the fact that checking whether an atom β over $\mathcal{O}$ J-covers an atom α over $\mathcal{S}$ can be done in polynomial time.

As for the correctness, we first show that the computed $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$. By construction, each disjunct $\{\vec{t} \mid \exists \vec{y}'. \phi'(\vec{x}, \vec{y}')\}$ of $q_{\mathcal{O}}$ satisfies the following condition: there is a query $q \in \mathsf{HeadCompletion}(q_{\mathcal{S}}, \mathsf{con})$ with target list $\vec{t}$ such that for each atom α_i of q there is an atom β_i of $\phi(\vec{x}, \vec{y}')$ that J-covers α_i, where $\mathsf{con} = \mathsf{con}_{\mathcal{M}} \cup \mathsf{con}_{\mathsf{q}_{\mathcal{S}}}$. Due to Lemma B.1, this implies that $\{\vec{t} \mid \exists \vec{y}'. \phi'(\vec{x}, \vec{y}')\}$ is a sound J-abstraction of a query $q \in \mathsf{HeadCompletion}(q_{\mathcal{S}}, \mathsf{con})$. Since for each query $q \in \mathsf{HeadCompletion}(q_{\mathcal{S}}, \mathsf{con})$ we trivially have that $q \sqsubseteq q_{\mathcal{S}}$, we derive that $\{\vec{t} \mid \exists \vec{y}'. \phi'(\vec{x}, \vec{y}')\}$ is a sound J-abstraction of $q_{\mathcal{S}}$ as well. Furthermore, since the above condition is true for each disjunct $\{\vec{t} \mid \exists \vec{y}'. \phi'(\vec{x}, \vec{y}')\}$ of $q_{\mathcal{O}}$, it follows that the computed $q_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$. We now show that $q_{\mathcal{O}}$ is actually the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$, that is, each UCQ $q'_{\mathcal{O}}$ that is a sound J-abstraction of $q_{\mathcal{S}}$ is such that $cert_{q'_{\mathcal{O}}, J} \sqsubseteq cert_{q_{\mathcal{O}}, J}$ (cf. Definition 3.4). We do this by way of contradiction.

Let $q'_{\mathcal{O}}$ be a UCQ such that $cert_{q'_{\mathcal{O}}, J} \not\sqsubseteq cert_{q_{\mathcal{O}}, J}$, that is, there exists an $\mathcal{S}$-database D consistent with J such that $cert^D_{q'_{\mathcal{O}}, J} \not\subseteq cert^D_{q_{\mathcal{O}}, J}$. It follows that there is a tuple of constants $\vec{c} = (c_1, \ldots, c_n)$ such that $\vec{c} \in cert^D_{q'_{\mathcal{O}}, J}$ but, at the same time, $\vec{c} \notin cert^D_{q_{\mathcal{O}}, J}$. If $\vec{c} \notin q^D_{\mathcal{S}}$, then $q'_{\mathcal{O}}$ is trivially not a sound J-abstraction of $q_{\mathcal{S}}$, and we are done. Therefore, we assume that $\vec{c} \in q^D_{\mathcal{S}}$. Specifically, let H the set of all homomorphisms h from $q_{\mathcal{S}}$ to D with $h(\vec{t}') = \vec{c}$ (where $\vec{t}'$ is the target list of $q_{\mathcal{S}}$), and let Γ be the set of all facts in D that partecipate in some homomorphism $h \in H$, i.e.: $\Gamma = \bigcup_{h \in H} h(q_{\mathcal{S}})$

Consider now the $\mathcal{S}$-database $\Delta = D \setminus \Gamma$. Obviously, since $\Delta \subseteq D$, and since the left-hand side of mapping assertions are CQs, we have that $\mathcal{C}^{\mathcal{M}(\Delta)}_{\mathcal{O}} \subseteq \mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$. In particular, let $\Lambda = \{\beta_1, \ldots, \beta_k\}$ be the set composed of all the facts in $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ that are not in $\mathcal{C}^{\mathcal{M}(\Delta)}_{\mathcal{O}}$, i.e., $\Lambda = \mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}} \setminus \mathcal{C}^{\mathcal{M}(\Delta)}_{\mathcal{O}}$ (since $\mathcal{M}$ is a pure GAV mapping and $\mathcal{O}$ is a *DL-Lite*$_{\mathsf{RDFS}}$ ontology, clearly, both $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}}$ and $\mathcal{C}^{\mathcal{M}(\Delta)}_{\mathcal{O}}$ do not introduce variables). We now exhibit an $\mathcal{S}$-database D' for which (i) $\vec{c} \notin q^{D'}_{\mathcal{S}}$, and $\mathcal{C}^{\mathcal{M}(D)}_{\mathcal{O}} \subseteq \mathcal{C}^{\mathcal{M}(D')}_{\mathcal{O}}$. To this aim, we exploit the following things:

- Let $q = \{\vec{t} \mid \exists \vec{y}. \phi(\vec{x}, \vec{y})\} = \{\vec{t} \mid \exists \vec{y}. \alpha_1 \wedge \ldots \wedge \alpha_{\eta(q_{\mathcal{S}})}\} \in \mathsf{HeadCompletion}(q_{\mathcal{S}}, \mathsf{con})$ be the most specific query over $\mathcal{S}$ for which $\vec{c} = (c_1, \ldots, c_n) \in q^D$ still holds, i.e., the CQ whose target list $\vec{t} = (t_1, \ldots, t_n)$ is such that (i) for each $i \in [1, n]$, if c_i is a constant occurring either in $\mathsf{con}_{\mathsf{q}_{\mathcal{S}}}$ or in $\mathsf{con}_{\mathcal{M}}$, then $t_i = c_i$ (otherwise term t_i is a distinguished variable), and (ii) for each pair of numbers $i, j \in [1, n]$, $c_i = c_j$ if and only if $t_i = t_j$.

- Consider the target list $\vec{t} = (t_1, \ldots, t_n)$ of q and the tuple of constants $\vec{c} = (c_1, \ldots, c_n)$. Let $\Lambda' = \{\beta'_1, \ldots, \beta'_k\}$ be the set of atoms over $\mathcal{O}$ obtained from the set of facts $\Lambda = \{\beta_1, \ldots, \beta_k\}$ by (i) replacing everywhere the constant $c_i \in \vec{c}$ with the term $t_i \in \vec{t}$ (either a distinguished variable, or the constant c_i itself), for each $i \in [1, n]$, and (ii) replacing everywhere each constant c occurring neither in $q_{\mathcal{S}}$ nor in $\mathcal{M}$ with a fresh existential variable y_c.

In particular, there are two possible cases: either for each $i \in [1, \eta(q_{\mathcal{S}})]$ there is an atom $\beta'_i \in \Lambda'$ such that β'_i J-covers α_i, or not.

In the former case, since for each atom α_i of q there is an atom $\beta'_i \in \Lambda'$ such that β'_i J-covers α_i, by construction of the algorithm, it can be readily seen that the CQ $q' = \{\vec{t} \mid \exists \vec{y}'.\beta'_1 \wedge \ldots \wedge \beta'_{\eta(q_{\mathcal{S}})}\}$ over $\mathcal{O}$ is a disjunct of $q_{\mathcal{O}}$. Furthermore, it is clear that $\vec{c} \in q'^{\Lambda}$. Two considerations are now in order: (i) due to the facts that $\vec{c} \in q'^{\Lambda}$ and $\Lambda \subseteq \mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$, and since q' is a CQ, we derive that $\vec{c} \in q'^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$ as well, and (ii) since $\vec{c} \in q'^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$ and since q' is a disjunct of $q_{\mathcal{O}}$, we have $\vec{c} \in q_{\mathcal{O}}{}^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$. Thus, as already observed, since in this setting for OBDM specifications $cert^D_{q_{\mathcal{O}},J} = q_{\mathcal{O}}^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$ for each UCQ $q_{\mathcal{O}}$ and $\mathcal{S}$-database D, we derive that $\vec{c} \in cert^D_{q_{\mathcal{O}},J}$, which is a contradiction on the initial assumption that $\vec{c} \notin cert^D_{q_{\mathcal{O}},J}$. It follows that the former case just considered is not possible because it leads to a contradiction. Therefore, we consider only the latter case.

Consider the latter case, that is, there exists at least an atom α_i of q for which no atom $\beta' \in \Lambda'$ is such that β' J-covers α_i. It is not hard to ascertain that this implies that there is at least an atom α'_i of $q_{\mathcal{S}}(\vec{c})$ for which no atom $\beta \in \Lambda$ is such that β J-covers α'_i, where we recall that $q_{\mathcal{S}}(\vec{c}) = \{() \mid \exists \vec{y}.\alpha'_1 \wedge \ldots \wedge \alpha'_{\eta(q_{\mathcal{S}})}\}$ is the boolean CQ obtained from $q_{\mathcal{S}}$ by replacing each occurrence of term t'_i in the body of $q_{\mathcal{S}}$ with constant c_i, for each $i \in [1, n]$ (where $\vec{t'} = (t_1, \ldots, t_n)$ is the target list of $q_{\mathcal{S}}$). But then, consider the set of facts Ω obtained by unfolding each fact $\beta \in \Lambda$ with a disjunct of $\rho(\beta, J)$ such that there is no atom over $\mathcal{S}$ that instantiates α' (clearly, since by assumption β does not J-cover α', following Definition B.1, at least one of such disjunct must exists). As a result, we trivially have that $\Omega \not\models q_{\mathcal{S}}(\vec{c})$, which implies that $\vec{c} \notin q_{\mathcal{S}}^{\Omega}$.

We now prove that the $\mathcal{S}$-database we are seeking is $D' = \Delta \cup \Omega$. Observe that: (i) $q_{\mathcal{S}}$ is a CQJFE, and therefore it does not have existential variables in join occurring in its body, (ii) from (i) and by construction of Δ, we know that there are no facts that may partecipate in a possible homomorphism from $q_{\mathcal{S}}$ to D with $h(\vec{t'}) = \vec{c}$ (where $\vec{t'}$ is the target list of $q_{\mathcal{S}}$) in Δ, (iii) $\vec{c} \notin q_{\mathcal{S}}^{\Omega}$. Putting together the above three observations, one can easily verify that they imply that $\vec{c} \notin q_{\mathcal{S}}^{D'}$, where $D' = \Delta \cup \Omega$. Furthermore, since $\mathcal{M}$ is a pure GAV mapping and $\mathcal{O}$ is a *DL-Lite*$_{\mathsf{RDFS}}$ ontology, and thus contains no assertions with $\exists R$ in the right-hand side for a basic role R, and since Ω is obtained by unfolding each atom $\beta \in \Lambda$ with a disjunct of $\rho(\beta, J)$, it is easy to verify that $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(\Omega)}$ is such that $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)} \subseteq \mathcal{C}_{\mathcal{O}}^{\mathcal{M}(\Omega)}$, which obviously implies that $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)} \subseteq \mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ because $\Omega \subseteq D'$ and the left-hand side of mapping assertions are CQs. Notice that $\vec{c} \in cert^D_{q'_{\mathcal{O}},J}$ holds by assumption, and therefore $\vec{c} \in q'_{\mathcal{O}}{}^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}}$. Furthermore, since $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)} \subseteq \mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}$ and $q_{\mathcal{O}}$ is a UCQ, we trivially derive that $\vec{c} \in q'_{\mathcal{O}}{}^{\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D')}}$ as well, which, in turn, implies that $\vec{c} \in cert^{D'}_{q'_{\mathcal{O}},J}$.

To complete the proof, consider the $\mathcal{S}$-database D'. We have that, on the one hand, $\vec{c} \notin q_{\mathcal{S}}^{D'}$, and, on the other hand, $\vec{c} \in cert^{D'}_{q'_{\mathcal{O}},J}$. It follows that $q'_{\mathcal{O}}$ is not a sound J-abstraction of $q_{\mathcal{S}}$, as required. □

As a specialization of Lemma 7.3 in the restricted scenario for CQJFEs, we have the following result which straightforward follows from the above theorem.

Corollary B.1. *Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification, and let $q_{\mathcal{S}}$ be a CQJFE over $\mathcal{S}$. If a CQ $q_{\mathcal{O}}$ over $\mathcal{O}$ is a sound J-abstraction of $q_{\mathcal{S}}$, then there exists a CQ $q'_{\mathcal{O}}$ over $\mathcal{O}$ with same target list of $q_{\mathcal{O}}$ such that* (i) *the body of $q'_{\mathcal{O}}$ is the conjunction of at most $\eta(q_{\mathcal{S}})$ atoms occurring in the body of $q_{\mathcal{O}}$ (and therefore, $cert_{q_{\mathcal{O}},J} \sqsubseteq cert_{q'_{\mathcal{O}},J}$), and* (ii) *$q'_{\mathcal{O}}$ is a sound J-abstraction of $q_{\mathcal{S}}$ as well.*

Regarding the cost of the algorithm, we observe that the overall running time is exponential in the size of the input. Indeed, the computation of the head completion of $q_{\mathcal{S}}$ with respect to $\mathsf{con} = \mathsf{con}_{\mathcal{M}} \cup \mathsf{con}_{\mathsf{q}_{\mathcal{S}}}$ is, in general, exponential with respect to the size of the target list of $q_{\mathcal{S}}$, even when $\mathsf{con} = \emptyset$. Furthermore, for each possible $q \in \mathsf{HeadCompletion}(q_{\mathcal{S}}, \mathsf{con})$, the generated disjuncts added to the final UCQ $q_{\mathcal{O}}$ are, potentially, exponentially many with respect to $\eta(q_{\mathcal{S}})$.

The next proposition shows that there exists OBDM specifications J and CQJFEs $q_{\mathcal{S}}$ for which the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ necessarily consists of an exponential number of disjuncts with respect to $\eta(q_{\mathcal{S}})$ for exponentially many source queries with respect to the size of the target list of $q_{\mathcal{S}}$.

Proposition B.1. *In the restricted scenario for CQJFEs, there are OBDM specifications J and CQJFEs $q_{\mathcal{S}}$ for which the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ is a UCQ having necessarily an exponential number of disjuncts with respect to $\eta(q_{\mathcal{S}})$, for exponentially many source queries with respect to the size of the target list of $q_{\mathcal{S}}$.*

Proof. We provide here a small example showing the main reason of why the UCQ-maximally sound J-abstraction of a CQJFE $q_{\mathcal{S}}$ may contain an exponential number of disjuncts with respect to $\eta(q_{\mathcal{S}})$, for exponentially many source queries with respect to the size of the target list of $q_{\mathcal{S}}$.

Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the following OBDM specification:

- $\mathcal{O} = \emptyset$
- $\mathcal{S} = \{\ s_1, s_{1,1}, s_{1,2}, s'_{1,1}, s'_{1,2}, s_2, s_{2,1}, s_{2,2}, s'_{2,1}, s'_{2,2}, s_3, s_{3,1}, s_{3,2}, s'_{3,1}, s'_{3,2}\ \}$
- $\mathcal{M} = \{\ m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9, m_{10}, m_{11}, m_{12}\ \}$, where:

$$\begin{aligned}
m_1\colon&\quad s_{1,1}(x_1,x_2) \wedge s_1(x_1,x_2) &\rightarrow&\quad P_{1,1}(x_1,x_2),\\
m_2\colon&\quad s_{1,2}(x_1,x_2) \wedge s_1(x_1,x_2) &\rightarrow&\quad P_{1,2}(x_1,x_2),\\
m_3\colon&\quad s'_{1,1}(x) \wedge s_1(x,x) &\rightarrow&\quad A_{1,1}(x),\\
m_4\colon&\quad s'_{1,2}(x) \wedge s_1(x,x) &\rightarrow&\quad A_{1,2}(x),\\
m_5\colon&\quad s_{2,1}(x_1,x_2) \wedge s_2(x_1,x_2) &\rightarrow&\quad P_{2,1}(x_1,x_2),\\
m_6\colon&\quad s_{2,2}(x_1,x_2) \wedge s_2(x_1,x_2) &\rightarrow&\quad P_{2,2}(x_1,x_2),\\
m_7\colon&\quad s'_{2,1}(x) \wedge s_2(x,x) &\rightarrow&\quad A_{2,1}(x),\\
m_8\colon&\quad s'_{2,2}(x) \wedge s_2(x,x) &\rightarrow&\quad A_{2,2}(x),\\
m_9\colon&\quad s_{3,1}(x_1,x_2) \wedge s_3(x_1,x_2) &\rightarrow&\quad P_{3,1}(x_1,x_2),\\
m_{10}\colon&\quad s_{3,2}(x_1,x_2) \wedge s_3(x_1,x_2) &\rightarrow&\quad P_{3,2}(x_1,x_2),\\
m_{11}\colon&\quad s'_{3,1}(x) \wedge s_3(x,x) &\rightarrow&\quad A_{3,1}(x),\\
m_{12}\colon&\quad s'_{3,2}(x) \wedge s_3(x,x) &\rightarrow&\quad A_{3,2}(x).
\end{aligned}$$

Let $q_{\mathcal{S}}$ be the following CQJFE over $\mathcal{S}$: $q_{\mathcal{S}} = \{(x_1,x_2,x_3,x_4,x_5,x_6) \mid s_1(x_1,x_2) \wedge s_2(x_3,x_4) \wedge s_3(x_5,x_6)\}$. Consider the following CQs occurring in $\mathsf{HeadCompletion}(q_{\mathcal{S}},\{\})$:

1. $q^1_{\mathcal{S}} = q_{\mathcal{S}} = \{(x_1,x_2,x_3,x_4,x_5,x_6) \mid s_1(x_1,x_2) \wedge s_2(x_3,x_4) \wedge s_3(x_5,x_6)\}$;
2. $q^2_{\mathcal{S}} = \{(x_1,x_1,x_3,x_4,x_5,x_6) \mid s_1(x_1,x_1) \wedge s_2(x_3,x_4) \wedge s_3(x_5,x_6)\}$;
3. $q^3_{\mathcal{S}} = \{(x_1,x_2,x_3,x_3,x_5,x_6) \mid s_1(x_1,x_2) \wedge s_2(x_3,x_3) \wedge s_3(x_5,x_6)\}$;
4. $q^4_{\mathcal{S}} = \{(x_1,x_2,x_3,x_4,x_5,x_5) \mid s_1(x_1,x_2) \wedge s_2(x_3,x_4) \wedge s_3(x_5,x_5)\}$;
5. $q^5_{\mathcal{S}} = \{(x_1,x_1,x_3,x_3,x_5,x_6) \mid s_1(x_1,x_1) \wedge s_2(x_3,x_3) \wedge s_3(x_5,x_6)\}$;
6. $q^6_{\mathcal{S}} = \{(x_1,x_1,x_3,x_4,x_5,x_5) \mid s_1(x_1,x_1) \wedge s_2(x_3,x_4) \wedge s_3(x_5,x_5)\}$;
7. $q^7_{\mathcal{S}} = \{(x_1,x_2,x_3,x_3,x_5,x_5) \mid s_1(x_1,x_2) \wedge s_2(x_3,x_3) \wedge s_3(x_5,x_5)\}$;
8. $q^8_{\mathcal{S}} = \{(x_1,x_1,x_3,x_3,x_5,x_5) \mid s_1(x_1,x_1) \wedge s_2(x_3,x_3) \wedge s_3(x_5,x_5)\}$.

One can verify that for each of the CQs illustrated above there are at least eight CQs over $\mathcal{O}$ that must necessarily appear in the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$. For instance, consider the query $q^6_{\mathcal{S}}$ in case 6. We have that $B_1 = \{P_{1,1}(x_1,x_1), P_{1,2}(x_1,x_1), A_{1,1}(x_1), A_{1,2}(x_1)\}$, $B_2 = \{P_{2,1}(x_3,x_4), P_{2,2}(x_3,x_4)\}$, and $B_3 = \{P_{3,1}(x_5,x_5), P_{3,2}(x_5,x_5), A_{3,1}(x_5), A_{3,2}(x_5)\}$ are the set of all the atoms that J-cover $s_1(x_1,x_1)$, $s_2(x_1,x_2)$, and $s_3(x_5,x_5)$, respectively. In particular, if we consider the subsets $B'_1 = \{A_{1,1}(x_1), A_{1,2}(x_1)\}$ and $B'_3 = \{A_{3,1}(x_1), A_{3,2}(x_5)\}$ of B_1 and B_3, respectively, then it is easy to verify that for each possible combination of atoms $(\beta_1,\beta_2,\beta_3)$ occurring in the Cartesian Product $B'_1 \times B_2 \times B'_3$, we have that the CQ $\{(x_1,x_1,x_3,x_4,x_5,x_5) \mid \exists\vec{y}'.\beta_1 \wedge \beta_2 \wedge \beta_3\}$ is necessarily a disjunct of the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$, and, moreover, each of these CQs will not be produced when considering any other query in $\mathsf{HeadCompletion}(q_{\mathcal{S}},\{\})$.

By generalizing the above construction, one can see that it is always possible to compose OBDM specifications $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and CQJFEs $q_{\mathcal{S}}$ for which the number of source queries to consider when computing the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ is equal to $2^{\frac{\eta(q_{\mathcal{S}})}{2}}$ (and therefore, an exponential number of source queries with respect to the size of the target list of $q_{\mathcal{S}}$). Furthermore, for each of these source queries, the number of disjuncts occurring in the UCQ-maximally sound J-abstraction of $q_{\mathcal{S}}$ is at least $2^{\eta(q_{\mathcal{S}})}$ (and therefore, an exponential number of disjuncts with respect to $\eta(q_{\mathcal{S}})$). □

References

[1] G. Cima, M. Lenzerini, A. Poggi, Semantic characterization of data services through ontologies, in: Proc. of the 28th Int. Joint Conf. on Artificial Intelligence, IJCAI, 2019, pp. 1647–1653.

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

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

[4] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodríguez-Muro, R. Rosati, Ontologies and databases: the *DL-Lite* approach, in: Reasoning Web. Semantic Technologies for Informations Systems – 5th Int. Summer School Tutorial Lectures (RW), in: Lecture Notes in Computer Science, vol. 5689, Springer, 2009, pp. 255–356.

[5] M. Bienvenu, Ontology-mediated query answering: Harnessing knowledge to get more from data, in: Proc. of the 25th Int. Joint Conf. on Artificial Intelligence, IJCAI, 2016, pp. 4058–4061.

[6] G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati, M. Zakharyaschev, Ontology-based data access: a survey, in: Proc. of the 27th Int. Joint Conf. on Artificial Intelligence, IJCAI, 2018, pp. 5511–5519.

[7] M. Ortiz, Improving data management using domain knowledge, in: Proc. of the 27th Int. Joint Conf. on Artificial Intelligence, IJCAI, 2018, pp. 5709–5713.

[8] M.J. Carey, N. Onose, M. Petropoulos, Data services, Commun. ACM 55 (6) (2012) 86–97.

[9] D. Machan, DaaS: the new information goldmine, Wall St. J. (2009) 1–3.

[10] M.D. Wilkinson, M. Dumontier, I.J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L.B. da Silva Santos, P.E. Bourne, J. Bouwman, A.J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C.T. Evelo, R. Finkers, A. Gonzalez-Beltran, A.J.G. Gray, P. Groth, C. Goble, J.S. Grethe, J. Heringa, P.A.C. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S.J. Lusher, M.E. Martone, A. Mons, A.L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M.A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, B. Mons, The FAIR guiding principles for scientific data management and stewardship, Sci. Data 3 (2016).

[11] Z. Zheng, J. Zhu, M.R. Lyu, Service-generated big data and big data-as-a-service: an overview, in: Proc. of the IEEE Int. Congress on Big Data (BigData Congress), 2013, pp. 403–410.

[12] G. Cima, Preliminary results on ontology-based open data publishing, in: Proc. of the 30th Int. Workshop on Description Logic (DL), in: CEUR Electronic Workshop Proceedings, vol. 1879, 2017, http://ceur-ws.org/.

[13] G. Cima, M. Lenzerini, A. Poggi, Semantic technology for open data publishing, in: Proc. of the 7th Int. Conf. on Web Intelligence, Mining and Semantics, WIMS, 2017, p. 1.

[14] R.M. Aracri, A.M. Bianco, R. Radini, M. Scannapieco, L. Tosco, F. Croce, M. Lenzerini, D.F. Savo, On the experimental usage of ontology-based data access for the Italian integrated system of statistical registers: quality issues, in: European Conference on Quality in Official Statistics, 2018.

[15] C. Lutz, J. Marti, L. Sabellek, Query expressibility and verification in ontology-based data access, in: Proc. of the 16th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR), 2018, pp. 389–398.

[16] D. Lembo, R. Rosati, V. Santarelli, D.F. Savo, E. Thorstensen, Mapping repair in ontology-based data access evolving systems, in: Proc. of the 26th Int. Joint Conf. on Artificial Intelligence, IJCAI, 2017, pp. 1160–1166.

[17] Z. Abedjan, L. Golab, F. Naumann, Data profiling: a tutorial, in: Proc. of the 2017 ACM Int. Conf. on Management of Data, SIGMOD, 2017, pp. 1747–1751.

[18] Z. Abedjan, L. Golab, F. Naumann, T. Papenbrock, Data Profiling, Synthesis Lectures on Data Management, Morgan & Claypool Publishers, 2018.

[19] M. Ortiz, Ontology-mediated queries from examples: a glimpse at the dl-lite case, in: Proceedings of the Fifth Global Conference on Artificial Intelligence, GCAI 2019, in: EPiC Series in Computing, vol. 65, 2019, pp. 1–14.

[20] G. Cima, F. Croce, M. Lenzerini, Query definability and its approximations in ontology-based data management, in: Proc. of the 30th Int. Conf. on Information and Knowledge Management, CIKM 2021, ACM, 2021, pp. 271–280.

[21] I.L. Salvadori, A. Huf, F. Siqueira, Semantic data-driven microservices, in: Forty-Third IEEE Annual Computer Software and Applications Conference, COMPSAC 2019, IEEE, 2019, pp. 402–410.

[22] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati, Tractable reasoning and efficient query answering in description logics: the *DL-Lite* family, J. Autom. Reason. 39 (3) (2007) 385–429.

[23] M. Lenzerini, Data integration: a theoretical perspective, in: Proc. of the 21st ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems, PODS, 2002, pp. 233–246.

[24] A. Doan, A.Y. Halevy, Z.G. Ives, Principles of Data Integration, Morgan Kaufmann, 2012.

[25] B.C. Grau, A possible simplification of the semantic web architecture, in: Proc. of the 13th Int. World Wide Web Conf., WWW, 2004, pp. 704–713.

[26] G. Cima, M. Lenzerini, A. Poggi, Answering conjunctive queries with inequalities in DL-Lite$_{\mathcal{R}}$, in: Proc. of the 34th AAAI Conf. on Artificial Intelligence, AAAI 2020, 2020, pp. 2782–2789.

[27] G. Cima, Abstraction in Ontology-Based Data Management, Frontiers in Artificial Intelligence and Applications, vol. 348, IOS Press, 2022, http://ebooks.iospress.nl/bookseries/frontiers-in-artificial-intelligence-and-applications.

[28] F. Baader, S. Brandt, C. Lutz, Pushing the $\mathcal{EL}$ envelope, in: Proc. of the 19th Int. Joint Conf. on Artificial Intelligence, IJCAI, 2005, pp. 364–369.

[29] D. Calvanese, G. De Giacomo, M. Lenzerini, M.Y. Vardi, Query processing under GLAV mappings for relational and graph databases, Proc. VLDB Endow. 6 (2) (2012) 61–72.

[30] A.Y. Levy, A.O. Mendelzon, Y. Sagiv, D. Srivastava, Answering queries using views, in: Proc. of the 14th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems, PODS, 1995, pp. 95–104.

[31] A.Y. Halevy, Answering queries using views: a survey, VLDB J. 10 (4) (2001) 270–294.

[32] G. Cima, M. Console, M. Lenzerini, A. Poggi, Abstraction in data integration, in: Proc. of the 36th Annual ACM/IEEE Symp. on Logic in Computer Science, LICS 2021, IEEE, 2021, pp. 1–11.

[33] F.N. Afrati, M. Gergatsoulis, T. Kavalieros, Answering queries using materialized views with disjunction, in: Proc. of the 7th Int. Conf. on Database Theory, ICDT, in: Lecture Notes in Computer Science, vol. 1540, Springer, 1999, pp. 435–452.

[34] O.M. Duschka, M.R. Genesereth, Query planning with disjunctive sources, in: Proc. of the AAAI-98 Workshop on AI and Information Integration, AAAI Press, 1998.

[35] S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison Wesley Publ. Co., 1995.

[36] 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.

[37] F. Baader, I. Horrocks, C. Lutz, U. Sattler, An Introduction to Description Logic, Cambridge University Press, 2017.

[38] T. Imielinski, W. Lipski Jr., Incomplete information in relational databases, J. ACM 31 (4) (1984) 761–791.

[39] P. Beame, P. Koutris, D. Suciu, Skew in parallel query processing, in: Proc. of the 33rd ACM SIGACT SIGMOD SIGAI Symp. on Principles of Database Systems, PODS, 2014, pp. 212–223.

[40] P. Koutris, P. Beame, D. Suciu, Worst-case optimal algorithms for parallel query processing, in: Proc. of the 19th Int. Conf. on Database Theory, ICDT 2016, in: LIPIcs, vol. 48, 2016, 8.

[41] B. Ketsman, D. Suciu, A worst-case optimal multi-round algorithm for parallel computation of conjunctive queries, in: Proc. of the 36th ACM SIGMOD-SIGACT-SIGAI Symp. on Principles of Database Systems, PODS 2017, 2017, pp. 417–428.

[42] E.F. Codd, A relational model of data for large shared data banks, Commun. ACM 13 (6) (1970) 377–387.

[43] M. Benedikt, P. Bourhis, L. Jachiet, E. Tsamoura, Balancing expressiveness and inexpressiveness in view design, in: Proc. of the 17th Int. Conf. on Principles of Knowledge Representation and Reasoning, KR 2020, 2020, pp. 109–118.

[44] A.K. Chandra, P.M. Merlin, Optimal implementation of conjunctive queries in relational data bases, in: Proc. of the 9th ACM Symp. on Theory of Computing, STOC, 1977, pp. 77–90.

[45] Y. Sagiv, M. Yannakakis, Equivalences among relational expressions with the union and difference operators, J. ACM 27 (4) (1980) 633–655.

[46] B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, C. Lutz, OWL 2 web ontology language profiles, in: W3C Recommendation, World Wide Web Consortium, second edition, 2012, available at http://www.w3.org/TR/owl2-profiles/.

[47] D. Brickley, R.V. Guha, RDF schema 1.1, W3C recommendation, world wide web consortium, available at https://www.w3.org/TR/2014/REC-rdf-schema-20140225/, 2014.

[48] 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, Springer, 2018, pp. 187–202.

[49] D. Maier, A.O. Mendelzon, Y. Sagiv, Testing implications of data dependencies, ACM Trans. Database Syst. 4 (4) (1979) 455–469.

[50] C. Beeri, M.Y. Vardi, A proof procedure for data dependencies, J. ACM 31 (4) (1984) 718–741.

[51] A. Calì, G. Gottlob, M. Kifer, Taming the infinite chase: query answering under expressive relational constraints, J. Artif. Intell. Res. 48 (2013) 115–174.

[52] B. ten Cate, L. Chiticariu, P.G. Kolaitis, W.C. Tan, Laconic schema mappings: computing the core with SQL queries, Proc. VLDB Endow. 2 (1) (2009) 1006–1017.

[53] R. Fagin, P.G. Kolaitis, R.J. Miller, L. Popa, Data exchange: semantics and query answering, Theor. Comput. Sci. 336 (1) (2005) 89–124.

[54] D. Calvanese, G. De Giacomo, M. Lenzerini, M.Y. Vardi, View-based query processing: on the relationship between rewriting, answering and losslessness, in: Proc. of the 10th Int. Conf. on Database Theory, ICDT, in: Lecture Notes in Computer Science, vol. 3363, 2005, pp. 321–336.

[55] M. Friedman, A. Levy, T. Millstein, Navigational plans for data integration, in: Proc. of the 16th Nat. Conf. on Artificial Intelligence, AAAI, 1999, pp. 67–73.

[56] M.R. Garey, D.S. Johnson, L.J. Stockmeyer, Some simplified NP-complete graph problems, Theor. Comput. Sci. 1 (3) (1976) 237–267.

[57] L.J. Stockmeyer, The polynomial-time hierarchy, Theor. Comput. Sci. 3 (1) (1976) 1–22.

[58] A. Miles, J.R. Pérez-Agüera, SKOS: simple knowledge organisation for the web, Cat. Classif. Q. 43 (3–4) (2007) 69–83.

[59] A. Miles, S. Bechhofer, SKOS simple knowledge organization system, W3C recommendation, world wide web consortium, available at http://www.w3.org/TR/skos-reference, 2009.

[60] S. Weibel, J.A. Kunze, C. Lagoze, M. Wolf, Dublin core metadata for resource discovery, Req. Comments 2413 (1998) 1–8.

[61] K. Knopp, Theory of Functions, Parts I and II, Dover Publications, 1996.

[62] C.H. Papadimitriou, Computational Complexity, Addison Wesley Publ. Co., 1994.

[63] G. Cima, D. Lembo, R. Rosati, D.F. Savo, Controlled query evaluation in description logics through instance indistinguishability, in: Proc. of the 29th Int. Joint Conf. on Artificial Intelligence, IJCAI 2020, 2020, pp. 1791–1797.

[64] A.Y. Levy, A. Rajaraman, J.J. Ordille, Query answering algorithms for information agents, in: Proc. of the 13th Nat. Conf. on Artificial Intelligence, AAAI, 1996, pp. 40–47.

[65] A. Calì, D. Calvanese, G. De Giacomo, M. Lenzerini, Data integration under integrity constraints, in: Proc. of the 14th Int. Conf. on Advanced Information Systems Engineering, CAiSE, in: Lecture Notes in Computer Science, vol. 2348, Springer, 2002, pp. 262–279.