Semantic Web 0 (2022) 1
IOS Press

Separability and its Approximations in Ontology-based Data Management

Gianluca Cima [a], Federico Croce [b] and Maurizio Lenzerini [b]
[a] *CNRS and University of Bordeaux, France*
E-mail: gianluca.cima@u-bordeaux.fr
[b] *Sapienza University of Rome, Italy*
E-mails: croce@diag.uniroma1.it, lenzerini@diag.uniroma1.it

Abstract. Given two datasets, i.e., two sets of tuples of constants, representing positive and negative examples, logical separability is the reasoning task of finding a formula in a certain target query language that separates them. As already pointed out in previous works, this task turns out to be relevant in several application scenarios such as concept learning and generating referring expressions. Besides, if we think of the input datasets of positive and negative examples as composed of tuples of constants classified, respectively, positively and negatively by a black-box model, then the separating formula can be used to provide global post-hoc explanations of such a model. In this paper, we study the separability task in the context of Ontology-based Data Management (OBDM), in which a domain ontology provides a high-level, logic-based specification of a domain of interest, semantically linked through suitable mapping assertions to the data source layer of an information system. Since a formula that properly separates (proper separation) two input datasets does not always exist, our first contribution is to propose (best) approximations of the proper separation, called (minimally) complete and (maximally) sound separations. We do this by presenting a general framework for separability in OBDM. Then, in a scenario that uses by far the most popular languages for the OBDM paradigm, our second contribution is a comprehensive study of three natural computational problems associated with the framework, namely Verification (check whether a given formula is a proper, complete, or sound separation of two given datasets), Existence (check whether a proper, or best approximated separation of two given datasets exists at all), and Computation (compute any proper, or any best approximated separation of two given datasets).

Keywords: Ontology-based Data Management, Separability, Explainable Artificial Intelligence, Semantic Technologies

1. Introduction

The separability problem deals with finding an intensional representation of two datasets, i.e., sets of data items, interpreted as positive and negative examples. In this problem, one is given two sets of data items, one with positive and the other with negative examples, and is asked to provide a query so that the evaluation of such a query over the database contains all the data items in the set of positive examples, and none of the data items in the set of negative examples. We say that a solution to this problem is a query that separates the given datasets. A special case of this problem arises when only one set of positive examples is given as input, and one is interested in finding a query whose evaluation over the database coincides with the data items in such a set. In this paper, we refer to the latter special case with the term characterizability, and we say that a solution to this problem is a query that characterizes the given dataset.

The separability problem has initially been studied for relational databases and is known in the community as the query-by-example problem[1]. Over the years, researchers have found several interesting applications of the separa-

[1] On the other hand, the characterizability problem is known in the literature as query definability.

bility problem, spanning from simplifying query formulation by non-experts, to debugging facilities for data engineers. Indeed, the problem has been studied as a useful tool for data exploration, concept learning, data analysis, usability, data security and more [1, 2]. Moreover, as already observed in [3], the problem is studied in two special cases in which the input datasets are constituted by only one single tuple. In separability, this special case is studied for entity comparison in RDF graphs, where the goal is to find a meaningful description that separates one entity from another. Similarly, in characterizability, this special case is studied for generating referring expressions (GRE), where one is interested in describing a single data item by a logical expression that allows to separate it from all other data items. With the rise of Machine Learning (ML), we argue that this topic acquires primary importance for providing meaningful explanations to any typical supervised black-box model used for classification tasks. When applied to classification, the ultimate goal of supervised learning is to construct models that are able to predict the target output (i.e., the class) of the proposed inputs. To achieve this, the learning algorithm is provided with some training examples that demonstrate the intended relation of input and output values. Then, the learned model is supposed to be able to correctly classify instances that have not been shown during training. A crucial problem for wise and safe adoption of ML-based black-box models is that, especially in high-risk domains such as healthcare and finance, it is often very hard to understand the rationale behind a classification made by these models. This may lead to discriminatory biases in the classification that were not intended and, more surprisingly, of which the designers were unaware of.

In this paper, we assume that the classification task is performed in an organization that adopts an Ontology-based Data Management (OBDM) approach [4, 5]. OBDM is a paradigm for accessing data using a conceptual representation of the domain of interest expressed as an ontology. The OBDM paradigm relies on a three-level architecture, consisting of the data layer, the ontology, and the mapping between the two. Consequently, an OBDM specification is a triple $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ which, together with an $\mathcal{S}$-database D, form a so-called OBDM system $\Sigma = \langle J, D \rangle$. We are going to tackle the separability problem by leveraging the notion of evaluation of a query with respect to an OBDM system, in turn based on the notion of *certain answers* to a query over an OBDM system. Intuitively, given an OBDM system $\Sigma = \langle J, D \rangle$ and two D-datasets λ^+ and λ^-, our goal is to derive a query expression over $\mathcal{O}$ that *separates* λ^+ and λ^- in Σ (called here *proper separation*). An important contribution of our work is to provide approximated results for all the cases in which it is not possible to provide a separating query. We argue that, in these cases, reasonable and useful ontological characterizations can still be provided. We propose to resort to suitable approximations of the proper separating query, by introducing the notions of sound and complete separating queries. The former is a query whose certain answers have empty intersection with the D-dataset λ^-, whereas the certain answers of the latter form a superset of the D-dataset λ^+. Obviously, we are interested in computing the best approximated separating queries, which we call maximally sound and minimally complete separations, respectively. A maximally sound (resp., minimally complete) separation is a sound (resp., complete) separation such that no other sound (resp., complete) separation exists that better approximates the input datasets. Moreover, we cover the special cases in which the input datasets are consituted by only one single tuple for all our results and refer to them as the *single tuple* variants of the problems we deal with.

In this context, the training set used in the classification task, which is a collection of data items that are labeled as positive and negative examples, is seen as two sets of tuples in the database schema. The query derived by solving the separability problem results into an intensional definition of such training set, and are considered as an explanation of the intensional properties of the training set. The same principle can also be applied to a set of tuples that has not been seen during the training of the model. In this scenario, one can consider the black-box model as an oracle that assigns a class to all given tuples. Then, the query derived by solving the separability problem in this new context, is considered as an explanation of the intrinsic behaviour of the model. Traditionally, there are two different types of explanations: *global* and *local*. We refer to the former for explanations of the general behaviour of the model, and to the latter for explanations of the output of the model with respect to a specific object. The present work poses the foundational basis for providing both kind of explanations. Indeed, it deals with global explanations when the separating query is searched with respect to positive and negative examples containing and arbitrary number of tuples. It deals with local explanations when the sets of positive and negative examples for which one searches for a separating query contain only one single tuple.

Our procedure fits into the definition of *post-hoc* explanations of black-box models, i.e. a set of techniques aimed at approximating the behaviour of a black-box model with a surrogate interpretable model. We are now going to

describe how in our context the role of the surrogate model is played by the query resulting from the solution of the separability problem. Suppose an organization that adopts the OBDM paradigm wants to train a model for predicting which candidates in a selection process are the most likely to perform well in a certain job. For training the model, the organization is given the curricula of current employees with a feedback on their performances that makes it possible to divide the training set in two different classes: the good and bad performers. For the sake of simplicity, suppose John, Mary, and Jane, who studied Biology, Medicine, and Math respectively, perform well. Suppose also that Matt, Angeline, and Jess, who studied Music, Linguistics, and Fashion respectively, perform badly. The ML-based black-box model is trained with this dataset and it is optimised to reach the highest possible accuracy. Now suppose some candidates apply for the job and are evaluated by the model. The latter states that Lucy, Mara, and George, who studied Math, Chemistry, and Physics respectively, have high probability to perform well in doing the job. At the same time, the model states that Lucas, and Paul, who studied Art, and Classics are believed to perform badly. The organization now wonders why the black-box model divided the candidates in this way. By instantiating a separability problem with the two sets of positive and negative candidates, the organization finds out that, no matter how sophisticated the internal details of the black-box model are, the resulting classification is so that all positive candidates are answers to the query *"return all the candidates with a scientific background"*, and none of the negative candidates are. This separating query provides an intuitive explanation of the actual behaviour of the model. Of course, there are in general many valid separating queries for a given instance of a separability problem, as it is possible that in many cases there is no valid separating query at all.

Contributions of the paper. The contribution provided by this paper can be summarized as follows:

- We present a formal framework for separability in OBDM. In particular, we first cast the classical notion of separating query in the OBDM context (called here *proper separation*), and then we propose the relaxations mentioned above (*complete separation* and *sound separation*) as well as their optimal versions (*minimally complete separation* and *maximally sound separation*). We do exactly the same for the special case of characterization, which deals only with a dataset of positive examples rather than both positive and negative examples.
- We study the Verification problem for separability and characterizability in OBDM, i.e. check whether a given query is a proper, complete, or sound separation (resp., characterization) of two given datasets (resp., a given dataset). More specifically, we introduce three families of decision problems for both separability and characterizability, called X-*V*SEP($\mathcal{L}_\mathcal{O}$,$\mathcal{L}_\mathcal{M}$,$\mathcal{Q}$) (resp., X-*V*CHAR($\mathcal{L}_\mathcal{O}$,$\mathcal{L}_\mathcal{M}$,$\mathcal{Q}$)) for X={Proper, Complete, Sound}, which are parametric with respect to an ontology language $\mathcal{L}_\mathcal{O}$, a mapping language $\mathcal{L}_\mathcal{M}$, and a query language $\mathcal{Q}$. We provide tight computational complexity results for the most common languages used in OBDM, i.e., $\mathcal{L}_\mathcal{O}$ is *DL-Lite*$_\mathcal{R}$, $\mathcal{L}_\mathcal{M}$ is GLAV, and $\mathcal{Q}$ is UCQ. In particular, both Proper-*V*SEP(*DL-Lite*$_\mathcal{R}$, GLAV, UCQ) and Proper-*V*CHAR(*DL-Lite*$_\mathcal{R}$, GLAV, UCQ) are DP-complete, both Complete-*V*SEP(*DL-Lite*$_\mathcal{R}$, GLAV, UCQ) and Complete-*V*CHAR(*DL-Lite*$_\mathcal{R}$, GLAV, UCQ) are NP-complete, and both Complete-*V*SEP(*DL-Lite*$_\mathcal{R}$, GLAV, UCQ) and Complete-*V*CHAR(*DL-Lite*$_\mathcal{R}$, GLAV, UCQ) are coNP-complete. Interestingly, all the lower bounds already hold for the single-tuple version of the decision problems (i.e., when all the input datasets are singleton sets).
- We study the Computation problem. We provide two algorithms that, taking as input an OBDM system $\Sigma = \langle\langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle, D\rangle$ and two D-datasets λ^+ and λ^-, where $\mathcal{O}$ is a *DL-Lite*$_\mathcal{R}$ ontology and $\mathcal{M}$ is a GLAV mapping, return, respectively, a UCQ-minimally complete Σ-separation of λ^+ and λ^- and a UCQ-maximally sound Σ-separation of λ^+ and λ^-. As a consequence, this proves that in the scenario under consideration the two best approximated versions of proper separations always exist.
- We study the Existence problem for both separability and characterizability in OBDM. Since for the scenario under consideration their two best approximated versions always exist, we only focus on the existence of a proper separation (resp., characterization), i.e., check whether a proper separation (resp., characterization) in a target query language $\mathcal{Q}$ of two given datasets (resp., a given dataset) exists at all. More specifically, we introduce a family of decision problems for both separability and characterizability, called SEP($\mathcal{L}_\mathcal{O}$, $\mathcal{L}_\mathcal{M}$, $\mathcal{Q}$) (resp., CHAR($\mathcal{L}_\mathcal{O}$, $\mathcal{L}_\mathcal{M}$, $\mathcal{Q}$)), which are parametric with respect to an ontology language $\mathcal{L}_\mathcal{O}$, a mapping language $\mathcal{L}_\mathcal{M}$, and a query language $\mathcal{Q}$. In particular, we prove that both SEP(*DL-Lite*$_\mathcal{R}$, LAV, UCQ) and CHAR(*DL-Lite*$_\mathcal{R}$, LAV, UCQ) are coNP-complete, both SEP(*DL-Lite*$_\mathcal{R}$, GAV, UCQ) and CHAR(*DL-Lite*$_\mathcal{R}$, GAV, UCQ) are Θ_2^p-complete, and both SEP(*DL-Lite*$_\mathcal{R}$, GLAV, UCQ) and CHAR(*DL-Lite*$_\mathcal{R}$, GLAV, UCQ) are

coNEXPTIME-complete. Again, all the lower bounds already hold for the single-tuple version of the decision problems.

This paper is an extended version of our CIKM'21 conference paper [6]. We point out that the conference paper focused only on the characterizability reasoning task, whereas here we illustrate a formal framework and study the related computational problems both for separability and characterizability. Moreover, in this extended version we provide all the full proofs that were only sketched on the conference version of this paper.

To the best of our knowledge, the present work is the first to introduce separability and characterizability in the OBDM context. In particular, while separability and characterizability have been studied in various ontology-enriched query answering settings, this is the first to take into account the full three layers stack of the OBDM paradigm, thus including also the mapping layer. Furthermore, this is the first paper that propose and study also natural relaxations of the separability and characterizability notions.

Finally, we mention that, since *DL-Lite*$_{\mathcal{R}}$ is insensitive to the adoption of the *unique name assumption (UNA)* for UCQ answering [7], all our results hold both with and without the UNA.

Outline of the paper. The paper is organized as follows. After the discussion of related works in Section 2, Section 3 introduces the relevant background for our study and Section 4 illustrates a formal framework for separation in OBDM. Then, Sections 5, 6, and 7 present the results on the three computational problems Verification, Computation, and Existence, respectively. Finally, Section 8 concludes the paper with a perspective on future work.

2. Related Work

Query definability and query-by-example have long been studied for classical databases, starting from [8] up to many recent studies [9–15]. We have laid the foundations for analysing the complexity of our decision problems from the results in [10, 11]. We found the work in [9] inspirational since they dealt with the problem of deriving a metric for establishing how close is a non proper separating query to the optimal solution. We found the survey in [13] important for the whole research problem as they present a well described motivating example and they outline the main open problems of this topic. Even though we decided to frame our problem in a different way, we compared our work with the approach described in [14] that is based on the query identification model introduced by Angluin [16].

Our work has also been inspired by the notion of *query abstraction* [17–19]. Although the goal is still to derive a query expression over the ontology, in that work the input is a query over the data layer of the OBDM architecture, whereas in query definability (resp. in query-by-example) the input is a set of tuples (or two sets of tuples). It follows that the two tasks are completely different and require different technical solutions: in [17–19], the goal is to find a query over the ontology such that the certain answers of the query are equal to the evaluation of the given query over the database schema, for all possible databases of the OBDM system. In the framework section of this paper we better characterize the difference between the present work and the ones in [17–19].

The works that are closer to ours are the ones in [20–22]. In [20] the authors study the existence, and verification problems both for query definability and for query-by-example, and computation problem for the query definability case. In their work, the ontology is expressed as an RDF graph and they consider several fragments of SPARQL as the query language to be used for the separation of the examples. Differently from the present work, they do not aim at finding the best approximated separation. We share with [21] the expressive power of the language used for the ontology (DL-Lite), and for the separation query the input examples (UCQs). However, the work in [21] does not deal with the cases in which proper separations for the input examples do not exist, and it is based on a slightly different framework from ours, i.e. since in that work they study the problems in the context of ontology-mediated queries, they do not have the mapping layer that instead is part of our more general OBDM paradigm.

The work in [22], studies the query-by-example problem for expressive horn description logic ontologies, namely Horn-$\mathcal{ALCI}$. Apart from the clear difference in the expressive power of the ontologies, their work does not consider the case of approximated separations of the input examples, and it does not consider the mapping layer.

This work has also been inspired by [23–26]. All these papers' goal is to learn a concept expression that best captures a given set of examples (or two set of positive and negative examples for query-by-example). We differ from these works because our goal is to derive a full-blown query that separates the input examples.

The problem of checking whether there exists a formula separating positive and negative examples in the presence of an ontology has recently been studied in [3]. Other than verifying that the ontology entails the searched formula for all positive examples, the authors conducted an in-depth analysis of the so-called separability problem accounting for both weak separability, i.e. the one we study in this paper, and strong separability, i.e. checking whether the negation of the separating formula is entailed by the KB. They also consider the case of enriching the separating formula by adding helper symbols that are not originally present in the ontology, and study the complexity of the decision problems for a wide range of languages both for expressing the ontology and the separating formula. In this paper, we are interested in studying the weak separability problem in the context of the OBDM by also considering the cases in which the separating formula does not exist and one wants to search for sound and complete approximations of it.

Another important line of research related to the present work is the one regarding post-hoc explanations of opaque machine learning models. As also highlighted by other works [27–30], the query-by-example problem can easily be adapted for explaining the output of a black-box machine learning classification model. For example, consider the case of a binary classifier labelling a set of examples in two classes 1 and 0. In this scenario, the solution of the query-by-example problem is considered as a surrogate of the machine learning model, so that the examples labelled by class 1 are the answers of the reverse engineered query, while none of the examples labelled by class 0 are. Therefore the query acts as an explanation because it provides a more human understandable way, especially in our framework in which the query is based on the knowledge of the ontology, for classifying the given examples in the two different classes. Although relevant for our work, we differ from all the above cited papers. In [28], they map the inputs of the machine learning classifier to the ontology and then uses a concept learning tool to find a class expression over the ontology that best describes the positive example. In [29], for explaining the behaviour of a black-box classifier, they build another black-box classifier (a neural network) and then project the output of this latter model onto a so-called rule space, where each coordinate represents the activation of a rule that is described in First Order Logic. In [27] they present the TREPAN algorithm, i.e. a way for building a decision tree in which the nodes are linked to an ontology, that is used as a means for explaining the input positive and negative examples. We consider the work in [30] to be relevant for our work, even though is rather preliminary and does not specify many important details such as the expressive power of the language of the ontology, and the language of the query they search for. The biggest differences with the present work are the fact that they do not consider the mapping layer between the ontology and the data, and that they do not focus on the concepts of maximally-sound and minimally-complete solutions, in cases where a perfect solution does not exist. On the contrary, they define a best approximated query as the one minimising the jaccard distance between the answers of the query and the set of positive examples in input.

Inductive Logic Programming (ILP) [31] has long been considered related to the query definability and query-by-example tasks. We also considered it inspiring for our work, but we soon acknowledged that the expressive power of the languages used for representing the knowledge base are incomparable, and that in ILP they are interested in searching for explanations of a set of logical facts rather than a set of tuples.

Finally, the Active Learning task initially introduced by [16] has been studied as a possible framework for learning queries from examples in the presence of a *DL-Lite* ontology [32]. Differently from our approach, this framework involves the use of two actors: a learner that proposes new queries, and an oracle that guides the learning process by either validating the proposed query or by providing counterexamples showing that the proposed query was unable to separate the input examples.

3. Preliminaries

We recall some notations and languages about relational databases [33], Description Logics (DLs) [34], and the Ontology-based Data Management (OBDM) paradigm [35].

Databases, Datasets, and Queries. A *relational database schema* (or simply *schema*) $\mathcal{S}$ is a finite set of predicate symbols, each with a specific arity. Given a schema $\mathcal{S}$, an *$\mathcal{S}$-database* D is a finite set of *facts* whose form is $s(\vec{c})$, where s is an n-ary predicate symbol of $\mathcal{S}$, and $\vec{c} = (c_1, \ldots, c_n)$ is an n-tuple of constants, each taken from a countable infinite set of symbols denoted by Const. We denote by $dom(D)$ the finite set of constants occurring in D.

Given a schema $\mathcal{S}$ and an $\mathcal{S}$-database D, a *D-dataset* λ of arity n is simply a finite set of n-tuples $\vec{c}$ of constants occurring in D, i.e., $\lambda \subseteq dom(D)^n$.

A *query* $q_\mathcal{S}$ over a schema $\mathcal{S}$ is an expression in a certain query language $\mathcal{Q}$ using the predicate symbols of $\mathcal{S}$ and arguments of predicates are *variables*, i.e., we disallow constants to occur in queries. Each query has an associated arity. The *evaluation* of a query $q_\mathcal{S}$ of arity n over an $\mathcal{S}$-databases D is a set of *answers* $q_\mathcal{S}^D$, each answer being an n-tuple of constants occurring in $dom(D)$, i.e., $q_\mathcal{S}^D \subseteq dom(D)^n$. A query $q_\mathcal{S}$ of arity 0 over a schema $\mathcal{S}$ is called a *boolean* query, and we denote by $q_\mathcal{S}^D = \{\langle\rangle\}$ (resp., $q_\mathcal{S}^D = \emptyset$) the fact that $D \models q_\mathcal{S}$ (resp., $D \not\models q_\mathcal{S}$). We are particularly interested in conjunctive queries and unions thereof.

A *conjunctive query (CQ)* over a schema $\mathcal{S}$ is an expression of the form $q_\mathcal{S} = \{\vec{x} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ such that (*i*) $\vec{x} = (x_1, \ldots, x_n)$, called the *target list* of $q_\mathcal{S}$, is an n-tuple of *distinguished variables*, where n is the arity $q_\mathcal{S}$ (*ii*) $\vec{y} = (y_1, \ldots, y_m)$ is an m-tuple of *existential variables*; and (*iii*) $\phi(\vec{x}, \vec{y})$, called the *body* of $q_\mathcal{S}$, is a finite conjunction of atoms of the form $s(v_1, \ldots, v_p)$, where s is an p-ary predicate symbol of $\mathcal{S}$ and v_i is either a distinguished or an existential variable, i.e., $v_i \in \vec{x} \cup \vec{y}$, for each $i = [1, p]$. Variables belong to a countable infinite set of symbols denoted by $\mathcal{V}$, where $\mathsf{Const} \cap \mathcal{V} = \emptyset$. A *union of conjunctive queries (UCQ)* is a finite set of CQs with same arity, called its *disjuncts*.

For a conjunction of atoms $\phi(\vec{x}, \vec{y})$, we denote by $set(\phi)$ the set of all the atoms occurring in ϕ. For a set of atoms $\mathcal{C}$ and a tuple $\vec{c} = (c_1, \ldots, c_n)$ of constants, we denote by $query(\mathcal{C}, \vec{c})$ the CQ $\{\vec{x} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$, where (*i*) $\phi(\vec{x}, \vec{y})$ is the conjunction of all the atoms occurring in the set of atoms $\mathcal{C}'$, where $\mathcal{C}'$ is obtained from $\mathcal{C}$ by replacing everywhere each constant c_i occurring in $\vec{c}$ with a fresh variable x_{c_i} and each constant c not occurring in $\vec{c}$ with a fresh variable y_c, (*ii*) $\vec{x} = (x_{c_1}, \ldots, x_{c_n})$, and (*iii*) $\vec{y}$ is the tuple of all variables occurring in $\mathcal{C}'$ that do not occur in $\vec{x}$.

Following the terminology of [11, 36], we say that a query $q_\mathcal{S}$ over a schema $\mathcal{S}$ *explains two D-datasets λ^+ and λ^-* (resp., *defines a D-dataset λ^+*) *inside an $\mathcal{S}$-database D* if $q_\mathcal{S}^D \subseteq \lambda^+$ and $q_\mathcal{S}^D \cap \lambda^- = \emptyset$ (resp., $q_\mathcal{S}^D = \lambda^+$). We also say that λ^+ and λ^- are *$\mathcal{Q}$-explainable* (resp., λ^+ is *$\mathcal{Q}$-definable*) *inside D*, for a query language $\mathcal{Q}$, if there exists a query $q_\mathcal{S} \in \mathcal{Q}$ that explains λ^+ and λ^- (resp., defines λ^+) inside D.

Given a set of atoms $\mathcal{C}$, we denote by $dom(\mathcal{C})$ the set of all constants and variables occurring in a set of atoms $\mathcal{C}$. Observe that $dom(\mathcal{C}) \subseteq \mathsf{Const} \cup \mathcal{V}$. Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two sets of atoms. We say that a function $h : dom(\mathcal{C}_1) \to dom(\mathcal{C}_2)$ is a *homomorphism* from $\mathcal{C}_1$ to $\mathcal{C}_2$ if $h(\mathcal{C}_1) \subseteq h(\mathcal{C}_2)$, where $h(\mathcal{C}_1)$ is the image of $\mathcal{C}_1$ under h, i.e., $h(\mathcal{C}_1) = \{h(\alpha) \mid \alpha \in \mathcal{C}_1\}$ with $h(s(t_1, \ldots, t_n)) = s(h(t_1), \ldots, h(t_n))$ for each atom $\alpha = s(t_1, \ldots, t_n)$. For two sets of atoms $\mathcal{C}_1$ and $\mathcal{C}_2$ and two tuples of terms $\vec{t_1}$ and $\vec{t_2}$, we write $(\mathcal{C}_1, \vec{t_1}) \to (\mathcal{C}_2, \vec{t_2})$ if there is a function h from $dom(\mathcal{C}_1) \cup \vec{t_1}$ to $dom(\mathcal{C}_2) \cup \vec{t_2}$ such that (*i*) h is a homomorphism from $\mathcal{C}_1$ to $\mathcal{C}_2$, and (*ii*) $h(\vec{t_1}) = \vec{t_2}$ (where, for a tuple of terms $\vec{t} = (t_1, \ldots, t_n)$, $h(\vec{t}) = (h(t_1), \ldots, h(t_n))$), $(\mathcal{C}_1, \vec{t_1}) \not\to (\mathcal{C}_2, \vec{t_2})$ otherwise.

Observe that for an $\mathcal{S}$-database D and a CQ $q_\mathcal{S} = \{\vec{x} \mid \exists \vec{y}.\phi(\vec{x}, \vec{y})\}$ over $\mathcal{S}$ of arity n, the set of answers $q_\mathcal{S}^D$ corresponds to the set of n-tuples $\vec{c}$ of constants occurring in D for which $(set(\phi), \vec{x}) \to (D, \vec{c})$.

Syntax and Semantics of DL-Lite$_\mathcal{R}$. DLs are fragments of First-order logic languages using only unary and binary predicates, called *atomic concepts* and *atomic roles*, respectively. In this paper, a *DL ontology* (or simply *ontology*) $\mathcal{O}$ is a TBox ("Terminological Box") expressed in a specific DL, that is, a set of assertions stating general properties of concepts and roles built according to the syntax of the specific DL, which represents the intensional knowledge of a modeled domain. Sometimes we also need to view $\mathcal{O}$ as a schema, in which cases we implicitly refer to the finite set of atomic concepts and atomic roles which constitute the *alphabet* of $\mathcal{O}$. We assume that every ontology $\mathcal{O}$ comprises the *bottom concept* $\bot$ in its alphabet. Whenever we speak about queries $q_\mathcal{O}$ over ontologies $\mathcal{O}$, we mean queries in a certain language $\mathcal{Q}$ using the atomic concepts and roles in the alphabet of $\mathcal{O}$ as predicate symbols.

We are interested in DL ontologies expressed in *DL-Lite$_\mathcal{R}$*, the member of the *DL-Lite* family [37] that underpins `OWL 2 QL`, i.e., the `OWL 2` profile especially designed for efficient query answering [38]. A *DL-Lite$_\mathcal{R}$* ontology $\mathcal{O}$ is a finite set of *assertions* of the form:

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

where B_1, B_2 are basic concepts, i.e., expressions of the form A, $\exists P$, or $\exists P^-$, with A and P an atomic concept and an atomic role, respectively, and R_1 and R_2 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 the disjointness assertion $B \sqsubseteq \neg B$.

Given a *DL-Lite*$_\mathcal{R}$ ontology $\mathcal{O}$, we denote by $V_\mathcal{O}$ the *$\mathcal{O}$-violation query*, i.e., the boolean UCQ over $\mathcal{O}$ constituted by the disjunct $\{() \mid \exists y.\bot(y)\}$ and a 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 $R_1 \sqsubseteq \neg R_2$) occurring in $\mathcal{O}$, 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., $R = P^-$.

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 constants, and the *interpretation function* $\cdot^\mathcal{I}$ assigns to each atomic concept A a set of domain objects $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$, 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 other basic concepts and basic roles as follows: $\bot^\mathcal{I} = \emptyset$, $(\exists P)^\mathcal{I} = \{o \mid \exists o'.\ (o, o') \in P^\mathcal{I}\}$, and $(P^-)^\mathcal{I} = \{(o, o') \mid (o', o) \in P^\mathcal{I}\}$. We often treat interpretations $\mathcal{I}$ for ontologies $\mathcal{O}$ as a (possibly infinite) set of facts over $\mathcal{O}$.

We say that an interpretation $\mathcal{I}$ for an ontology $\mathcal{O}$ satisfies $\mathcal{O}$, denoted by $\mathcal{I} \models \mathcal{O}$, if $\mathcal{I}$ satisfies every assertion in $\mathcal{O}$. For the *DL-Lite*$_\mathcal{R}$ assertions, an interpretation $\mathcal{I}$ satisfies 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 it satisfies 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, $R_1^\mathcal{I} \cap R_2^\mathcal{I} = \emptyset$).

For a UCQ $q_\mathcal{O}$ over a *DL-Lite*$_\mathcal{R}$ ontology $\mathcal{O}$, we denote by $\mathsf{PerfectRef}(\mathcal{O}, q_\mathcal{O})$ the UCQ computed by executing the algorithm PerfectRef [37] on $\mathcal{O}$ and $q_\mathcal{O}$ (slightly extended to deal also with the bottom concept $\bot$).

Ontology-based Data Management. According to [35, 39], an *Ontology-based Data Management (OBDM)* specification 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*, and $\mathcal{M}$ is a *mapping*, i.e., a finite set of assertions over the signature $\mathcal{S} \cup \mathcal{O}$ relating the source schema $\mathcal{S}$ to the ontology $\mathcal{O}$. An OBDM system is a pair $\Sigma = \langle J, D \rangle$, where $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ is an OBDM specification and D is an $\mathcal{S}$-database.

The semantics of an OBDM system $\Sigma = \langle \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle, D \rangle$ is given in terms of interpretations $\mathcal{I} = \langle \Delta^\mathcal{I}, \cdot^\mathcal{I} \rangle$ for $\mathcal{O}$ in which the interpretation function $\cdot^\mathcal{I}$ further assigns to each constant $c \in dom(D)$ a domain object $c \in \Delta^\mathcal{I}$. Specifically, we say that an interpretation $\mathcal{I}$ for $\mathcal{O}$ is a *model* of an OBDM system $\Sigma = \langle \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle, D \rangle$ if (*i*) $\mathcal{I} \models \mathcal{O}$, and (*ii*) $\langle \mathcal{I}, D \rangle \models \mathcal{M}$, where $\langle \mathcal{I}, D \rangle$ denotes the set of facts over the signature $\mathcal{S} \cup \mathcal{O}$. We say that an OBDM system Σ is *consistent* if it has at least one model, *inconsistent* otherwise.

The set of *certain answers* of a query $q_\mathcal{O}$ over an ontology $\mathcal{O}$ w.r.t. an OBDM system $\Sigma = \langle J, D \rangle$ with $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$, denoted by $cert^D_{q_\mathcal{O}, J}$, is the set of tuples of constants $(c_1, \ldots, c_n)$ occurring in D such that $(c_1^\mathcal{I}, \ldots, c_n^\mathcal{I}) \in q_\mathcal{O}^\mathcal{I}$ for each model $\mathcal{I}$ of Σ, where $\mathcal{I}$ is seen as a set of facts over $\mathcal{O}$. If Σ is inconsistent, then the set of certain answers of any query $q_\mathcal{O}$ over $\mathcal{O}$ w.r.t. Σ is simply the set of all possible tuples of constants occurring in D whose arity is the one of the query. We say that two queries q_1 and q_2 are equivalent w.r.t. an OBDM system $\Sigma = \langle J, D \rangle$ if $cert^D_{q_1, J} = cert^D_{q_2, J}$.

As for the mapping component of an OBDM system, in this paper we are interested in *Global-And-Local-As-View (GLAV)* assertions [40], which are assertions of the form $q_\mathcal{S} \rightarrow q_\mathcal{O}$, where $q_\mathcal{S}$ and $q_\mathcal{O}$ are CQs over $\mathcal{S}$ and over $\mathcal{O}$, respectively, with the same target list $\vec{x} = (x_1, \ldots, x_n)$. Special cases of GLAV assertions highly considered in the data integration literature are *Global-As-View (GAV)* and *Local-As-View (LAV)* assertions [41]: in a GAV (resp., LAV) assertion, $q_\mathcal{O}$ (resp., $q_\mathcal{S}$) is simply an atom without existential variables. A GLAV (resp., GAV, LAV, GAV∩LAV) mapping is a finite set of GLAV (resp., GAV, LAV, both GAV and LAV) assertions.

Given a GLAV mapping $\mathcal{M}$ relating a source schema $\mathcal{S}$ to an ontology $\mathcal{O}$, an interpretation $\mathcal{I}$ for $\mathcal{O}$, and an $\mathcal{S}$-database D, we say that $\langle \mathcal{I}, D \rangle \models \mathcal{M}$ if $(c_1, \ldots, c_n) \in q_\mathcal{S}^D$ implies $(c_1^\mathcal{I}, \ldots, c_n^\mathcal{I}) \in q_\mathcal{O}^\mathcal{I}$ for each mapping assertion $q_\mathcal{S} \rightarrow q_\mathcal{O}$ occurring in $\mathcal{M}$ and for each possible tuple $(c_1, \ldots, c_n)$ of constants occurring in D.

Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification where $\mathcal{O} = \emptyset$, i.e., $\mathcal{O}$ has no assertions, and $\mathcal{M}$ is a GLAV mapping. From results of [42, 43], it is well-known that, given a UCQ $q_\mathcal{O}$ over $\mathcal{O}$, by splitting the GLAV mapping

$\mathcal{M}$ into a GAV mapping followed by a LAV mapping over an intermediate alphabet, it is always possible to compute a UCQ over $\mathcal{S}$, denoted by $\mathsf{MapRef}(\mathcal{M}, q_\mathcal{O})$, such that $\mathsf{MapRef}(\mathcal{M}, q_\mathcal{O})^D = cert^D_{q_\mathcal{O},J}$ for each $\mathcal{S}$-database D.

Let now $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be an OBDM specification where $\mathcal{O}$ is a *DL-Lite*$_\mathcal{R}$ ontology and $\mathcal{M}$ is a GLAV mapping. For a UCQ $q_\mathcal{O}$ over $\mathcal{O}$, we denote by $\mathsf{rew}_{q_\mathcal{O},J}$ the following UCQ over $\mathcal{S}$: $\mathsf{rew}_{q_\mathcal{O},J} := \mathsf{MapRef}(\mathcal{M}, \mathsf{PerfectRef}(\mathcal{O}, q_\mathcal{O}))$. By combining the above observation with the correctness of the PerfectRef algorithm [37], we have that (*i*) $cert^D_{q_\mathcal{O},J} = \mathsf{rew}^D_{q_\mathcal{O},J}$, for each UCQ $q_\mathcal{O}$ over $\mathcal{O}$ and for each $\mathcal{S}$-database D such that $\langle J, D \rangle$ is consistent, and (*ii*) for each $\mathcal{S}$-database D, the OBDM system $\langle J, D \rangle$ is inconsistent if and only if $\mathsf{rew}^D_{V_\mathcal{O},J} = \{\langle\rangle\}$. We also note that *DL-Lite*$_\mathcal{R}$ is insensitive to the adoption of the unique name assumption for UCQ answering [7].

Canonical Structure. Given an $\mathcal{S}$-database D and a GLAV mapping $\mathcal{M}$ relating a source schema $\mathcal{S}$ to an ontology $\mathcal{O}$, the *chase* [44] of D with respect to $\mathcal{M}$, denoted by $\mathcal{M}(D)$, is the set of atoms computed as follows: (*i*) $\mathcal{M}(D)$ is initially set to the the empty set of atoms over $\mathcal{O}$; then (*ii*) for every GLAV assertion $\{\vec{x} \mid \exists \vec{y}.\phi_\mathcal{S}(\vec{x},\vec{y})\} \rightarrow \{\vec{x} \mid \exists \vec{z}.\varphi_\mathcal{O}(\vec{x},\vec{z})\}$ in $\mathcal{M}$ and for every tuple of constants $\vec{c}$ such that $(set(\phi_\mathcal{S}), \vec{x}) \rightarrow (D, \vec{c})$, we add to $\mathcal{M}(D)$ the image of the set of atoms $set(\varphi_\mathcal{O})$ under h', that is, we set $\mathcal{M}(D) := \mathcal{M}(D) \cup h'(\varphi_\mathcal{O}(\vec{x},\vec{z}))$, where h' extends h by assigning to each variable z occurring in $\vec{z}$ a different fresh variable of $\mathcal{V}$ still not present in $dom(\mathcal{M}(D))$. Observe that $\mathcal{M}(D)$ is guaranteed to be finite and can be always computed in exponential time.

We conclude this section with the following observation used in the technical development of the next sections. Let $\Sigma = \langle \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle, D \rangle$ be an OBDM system where $\mathcal{O}$ is a *DL-Lite*$_\mathcal{R}$ ontology and $\mathcal{M}$ is a GLAV mapping. We call the *canonical structure* of $\mathcal{O}$ with respect to $\mathcal{M}$ and D, denoted by $\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}$, the (possibly infinite) set of atoms obtained by first computing $\mathcal{M}(D)$ as described before, and then by chasing $\mathcal{M}(D)$ with respect to the inclusion assertions of $\mathcal{O}$ as described in [37, Definition 5] but using the alphabet $\mathcal{V}$ of variables whenever a new element is needed in the chase. Observe that this latter is a *fair* deterministic strategy, i.e., it is such that if at some point an assertion is applicable, then it will be eventually applied. By combining results of [45, Proposition 4.2] with [37, Theorem 29], it is well-known that, for a UCQ $q_\mathcal{O} = \{\vec{x_1} \mid \exists \vec{y_1}.\phi^1_\mathcal{O}(\vec{x_1},\vec{y_1})\} \cup \ldots \cup \{\vec{x_p} \mid \exists \vec{y_p}.\phi^p_\mathcal{O}(\vec{x_p},\vec{y_p})\}$ over $\mathcal{O}$ and a tuple of constants $\vec{c}$, if $\Sigma = \langle J, D \rangle$ is consistent, then we have $\vec{c} \in cert^D_{q_\mathcal{O},J}$ if and only if $(set(\phi^i_\mathcal{O}), \vec{x_i}) \rightarrow (\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{c})$ for some $i \in [1, p]$.

4. Framework

In what follows, we implicitly use $\Sigma = \langle J, D \rangle$ to denote an OBDM system where $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ is an OBDM specification and D is an $\mathcal{S}$-database. Intuitively, given two sets λ^+ and λ^- of tuples of constants occurring in D (i.e., λ^+ and λ^- are D-datasets) of positive and negative examples, respectively, we aim at finding a query $q_\mathcal{O}$ over the ontology $\mathcal{O}$ in a certain target query language $\mathcal{Q}$ that logically separates λ^+ and λ^- w.r.t. Σ. Since the evaluation of queries in OBDM systems is based on certain answers, we are naturally led to the following definition.

Definition 1. *$q_\mathcal{O} \in \mathcal{Q}$ is a* proper Σ-separation *of λ^+ and λ^- in the query language $\mathcal{Q}$, if both conditions (*i*) $\lambda^+ \subseteq cert^D_{q_\mathcal{O},J}$ and (*ii*) $cert^D_{q_\mathcal{O},J} \cap \lambda^- = \emptyset$ hold.*

The next example illustrates the above definition.

Example 1. *Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the OBDM specification in which $\mathcal{O}$ is the DL-Lite$_\mathcal{R}$ ontology $\mathcal{O} = \{\mathsf{MathStudent} \sqsubseteq \mathsf{Student}, \mathsf{ForeignStudent} \sqsubseteq \mathsf{Student}\}$, $\mathcal{S} = \{s_1, s_2, s_3, s_4, s_5\}$, and the mapping $\mathcal{M}$ consists of the following GAV assertions:*

$$\{(x) \mid s_1(x)\} \rightarrow \{(x) \mid \mathsf{Student}(x)\}$$

$$\{(x) \mid s_2(x)\} \rightarrow \{(x) \mid \mathsf{Student}(x)\}$$

$$\{(x_1,x_2) \mid s_3(x_1,x_2)\} \rightarrow \{(x_1,x_2) \mid \mathsf{EnrolledIn}(x_1,x_2)\}$$

$$\{(x) \mid \exists y.s_3(x,y) \wedge s_4(y)\} \rightarrow \{(x) \mid \mathsf{MathStudent}(x)\}$$

$$\{(x) \mid \exists y.s_3(x,y) \wedge s_5(y)\} \rightarrow \{(x) \mid \mathsf{ForeignStudent}(x)\}$$

Consider now the OBDM system $\Sigma = \langle J, D\rangle$, where J is the OBDM specification illustrated above and D is the $\mathcal{S}$-database $D = \{s_1(c_4), s_2(c_3), s_4(b_1), s_5(d_1), s_3(c_1, b_1), s_3(c_2, d_1), s_3(c_3, e_1), s_3(c_4, e_2), s_3(c_5, e_3)\}$. For the D-datasets $\lambda_1^+ = \{(c_1), (c_2), (c_3)\}$ and $\lambda_1^- = \{(c_5)\}$, one can verify that the CQ $q^1_{\mathcal{O}} = \{(x) \mid \mathsf{Student}(x)\}$ over $\mathcal{O}$ is a proper Σ-separation of λ_1^+ and λ_1^- in CQ because $cert^D_{q^1_{\mathcal{O}},J} = \{(c_1), (c_2), (c_3), (c_4)\}$.

Consider now a slight variation of the negative examples, i.e., let $\lambda^+ = \lambda_1^+$ and $\lambda^- = \lambda_1^- \cup \{(c_4)\}$. Since $q^1_{\mathcal{O}}$ and $q^2_{\mathcal{O}} = \{(x) \mid \exists y.\mathsf{EnrolledIn}(x, y)\}$ are such that $cert^D_{q^1_{\mathcal{O}},J} = \{(c_1), (c_2), (c_3), (c_4)\}$ and $cert^D_{q^2_{\mathcal{O}},J} = \{(c_1), (c_2), (c_3), (c_4), (c_5)\}$, and since $q^3_{\mathcal{O}} = \{(x) \mid \mathsf{MathStudent}(x)\}$ and $q^4_{\mathcal{O}} = \{(x) \mid \mathsf{ForeignStudent}(x)\}$ are such that $cert^D_{q^3_{\mathcal{O}},J} = \{(c_1)\}$ and $cert^D_{q^4_{\mathcal{O}},J} = \{(c_2)\}$, one can verify that no proper Σ-separation of λ^+ and λ^- in UCQ exists.

Clearly, the more expressive the target query language $\mathcal{Q}$, the more likely it is possible to distinguish (w.r.t. the OBDM system) the properties between the tuples in λ^+ and λ^- by means of the operators in $\mathcal{Q}$, and therefore the more likely the proper separation in $\mathcal{Q}$ exists. Unfortunately, the next example shows that, even in trivial cases and without any restriction on the target query language $\mathcal{Q}$, proper separations are not guaranteed to exist.

Example 2. *Let $\Sigma = \langle J, D\rangle$ be the following OBDM system: (*i*) $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ is the OBDM specification in which $\mathcal{O} = \emptyset$, i.e., $\mathcal{O}$ contains no assertions, $\mathcal{S} = \{s_1, s_2\}$, and $\mathcal{M} = \{m_1, m_2\}$ with $m_1 = \{(x) \mid s_1(x)\} \rightarrow \{(x) \mid A(x)\}$ and $m_2 = \{(x) \mid s_2(x)\} \rightarrow \{(x) \mid A(x)\}$; and (*ii*) D is the $\mathcal{S}$-database $D = \{s_1(c_1), s_2(c_2)\}$.*

For the D-datasets $\lambda^+ = \{(c_1)\}$ and $\lambda^- = \{(c_2)\}$, one can trivially verify that, whatever is the query language $\mathcal{Q}$, there can be no query $q_{\mathcal{O}} \in \mathcal{Q}$ over $\mathcal{O}$ for which $cert^D_{q_{\mathcal{O}},J}$ include the tuple (c_1) but does not include the tuple (c_2). To see this, note that $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)} = \{A(c_1), A(c_2)\}$. It follows that, whatever is the query language $\mathcal{Q}$, there can be no query $q_{\mathcal{O}} \in \mathcal{Q}$ over $\mathcal{O}$ which is a proper Σ-separation of λ^+ and λ^- in $\mathcal{Q}$.

Notice the importance of the role played by the mapping $\mathcal{M}$ in order to reach this conclusion. Indeed, if we replace m_2 with $\{(x) \mid s_2(x)\} \rightarrow \{(x) \mid B(x)\}$, then a proper Σ-separation of λ^+ and λ^- in UCQ would simply be the CQ $\{(x) \mid A(x)\}$ over the ontology $\mathcal{O}$.

Borrowing similar ideas from [17], to remedy situations where proper separations do not exist, we now introduce approximations of proper separations in terms of completeness and soundness. More specifically, a complete separation is a query that captures all the positive examples in its set of certain answers w.r.t. the OBDM system (i.e., satisfies condition (*i*) of Definition 1), whereas a sound separation is a query that contains none of the negative examples in its set certain answers w.r.t. the OBDM system (i.e., satisfies condition (*ii*) of Definition 1).

Definition 2. *$q_{\mathcal{O}} \in \mathcal{Q}$ is a* complete *(resp.,* sound*) Σ-*separation *of λ^+ and λ^- in the query language $\mathcal{Q}$, if $\lambda^+ \subseteq cert^D_{q_{\mathcal{O}},J}$ (resp., $cert^D_{q_{\mathcal{O}},J} \cap \lambda^- = \emptyset$).*

We observe that the condition for being a complete (resp., sound) separation does not look at λ^- (resp., λ^+).

Example 3. *Refer to Example 1. We have that $q^1_{\mathcal{O}}$ and $q^2_{\mathcal{O}}$ are complete Σ-separations of λ^+ and λ^- in UCQ, whereas $q^3_{\mathcal{O}}$ and $q^4_{\mathcal{O}}$ are sound Σ-separations of λ^+ and λ^- in UCQ.*

As the above example manifests, there may be several complete and sound separations. In those cases, the interest is unquestionably in those queries that approximate at *best* the proper one, at least relative to a target query language $\mathcal{Q}$. Informally, a $\mathcal{Q}$-minimally complete separation is a complete separation in $\mathcal{Q}$ containing a minimal (w.r.t. set containment) possible set of negative examples in its set of certain answers, whereas a $\mathcal{Q}$-maximally sound separation is a sound separation in $\mathcal{Q}$ capturing a maximal (w.r.t. set containment) possible set of positive examples in its set of certain answers.

Definition 3. *$q_{\mathcal{O}}$ is a* $\mathcal{Q}$-minimally complete *(resp., $\mathcal{Q}$-*maximally sound*) Σ-*separation *of λ^+ and λ^-, if $q_{\mathcal{O}}$ is a complete (resp., sound) Σ-separation of λ^+ and λ^- in $\mathcal{Q}$ and there is no $q'_{\mathcal{O}} \in \mathcal{Q}$ satisfying both the following two conditions:*

*(*i*) $q'_{\mathcal{O}}$ is a complete (resp., sound) Σ-separation of λ^+ and λ^- in $\mathcal{Q}$;*
*(*ii*) $cert^D_{q'_{\mathcal{O}},J} \cap \lambda^- \subset cert^D_{q_{\mathcal{O}},J} \cap \lambda^-$ (resp., $cert^D_{q_{\mathcal{O}},J} \cap \lambda^+ \subset cert^D_{q'_{\mathcal{O}},J} \cap \lambda^+$).*

Example 4. *Refer again to Example 1. One can verify that the CQ $q^1_{\mathcal{O}}$ is a UCQ-minimally complete Σ-separation of λ^+ and λ^-, whereas $q^2_{\mathcal{O}}$ is not. Moreover, both $q^3_{\mathcal{O}}$ and $q^4_{\mathcal{O}}$ are CQ-maximally sound Σ-separations of λ^+ and λ^-, but neither of them is a UCQ-maximally sound Σ-separation of λ^+ and λ^-. Indeed, a UCQ-maximally sound Σ-separation of λ^+ and λ^- is $q^5_{\mathcal{O}} = q^3_{\mathcal{O}} \cup q^4_{\mathcal{O}}$.*

We point out that all the above definitions are a generalization of the definitions illustrated in the conference paper [46] which deal only with datasets of positive examples, rather than datasets of both positive and negative examples as done here. More specifically, in [46], as well as in all the works addressing the separability task, when only a D-dataset λ^+ of positive examples is provided, to λ^- it is implicitly associated the D-dataset $\lambda^- = dom(D)^n \setminus \lambda^+$, where n is the arity of the tuples in λ^+. With this remark in mind, we can now specialize the above definitions for the only dataset of positive examples case and report the definitions given in [46].

Definition 4. *$q_{\mathcal{O}} \in \mathcal{Q}$ is a* proper *(resp.,* complete*,* sound*)* Σ-characterization *of λ^+ in the query language $\mathcal{Q}$, if $cert^D_{q_{\mathcal{O}},J} = \lambda^+$ (resp., $\lambda^+ \subseteq cert^D_{q_{\mathcal{O}},J}$, $cert^D_{q_{\mathcal{O}},J} \subseteq \lambda^+$).*

$q_{\mathcal{O}}$ is a $\mathcal{Q}$-minimally complete *(resp.,* $\mathcal{Q}$-maximally sound*)* Σ-characterization *of λ^+, if $q_{\mathcal{O}}$ is a complete (resp., sound) Σ-characterization of λ^+ in $\mathcal{Q}$ and there is no $q'_{\mathcal{O}} \in \mathcal{Q}$ satisfying both the following two conditions:*

*(*i*) $q'_{\mathcal{O}}$ is a complete (resp., sound) Σ-characterization of λ^+ in $\mathcal{Q}$;*
*(*ii*) $cert^D_{q'_{\mathcal{O}},J} \subset cert^D_{q_{\mathcal{O}},J}$ (resp., $cert^D_{q_{\mathcal{O}},J} \subset cert^D_{q'_{\mathcal{O}},J}$).*

In other words, given an OBDM system $\Sigma = \langle J, D\rangle$ with $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$, a D-dataset λ^+ of arity n, and a query $q_{\mathcal{O}} \in \mathcal{Q}$, we have that $q_{\mathcal{O}}$ is a proper in $\mathcal{Q}$ (resp., complete in $\mathcal{Q}$, sound in $\mathcal{Q}$, $\mathcal{Q}$-minimally complete, $\mathcal{Q}$-maximally sound) Σ-characterization of λ^+ if and only if $q_{\mathcal{O}}$ is a proper in $\mathcal{Q}$ (resp., complete in $\mathcal{Q}$, sound in $\mathcal{Q}$, $\mathcal{Q}$-minimally complete, $\mathcal{Q}$-maximally sound) Σ-separation of λ^+ and λ^-, where $\lambda^- = dom(D)^n \setminus \lambda^+$.

4.1. Relation with the Abstraction *reasoning task*

We now discuss the relation between the notion of separation in OBDM introduced here with the notion of *Abstraction* [18, 19], recently introduced in [47] and studied under various scenarios [17, 48, 49] for addressing several reverse engineering tasks in OBDM. In Abstraction, we are given an OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ and a query $q_{\mathcal{S}}$ over $\mathcal{S}$, and the aim is to find a query $q_{\mathcal{O}}$ over $\mathcal{O}$, called *a perfect J-abstraction of $q_{\mathcal{S}}$*, such that $cert^D_{q_{\mathcal{O}},J} = q^D_{\mathcal{S}}$ *for each $\mathcal{S}$-database D* for which $\langle J, D\rangle$ is consistent. Conversely, in the separation task also the $\mathcal{S}$-database D is given, and instead of a query $q_{\mathcal{S}}$ we have two set of tuples λ^+ and λ^- of constants taken from D, and the aim is to find a query $q_{\mathcal{O}}$ over $\mathcal{O}$ such that both $\lambda^+ \subseteq cert^D_{q_{\mathcal{O}},J}$ and $cert^D_{q_{\mathcal{O}},J} \cap \lambda^- = \emptyset$ hold.

Following [17], we also say that a query $q_{\mathcal{O}}$ over $\mathcal{O}$ is *a complete (resp., sound) J-abstraction of $q_{\mathcal{S}}$* if $q^D_{\mathcal{S}} \subseteq cert^D_{q_{\mathcal{O}},J}$ (resp., $cert^D_{q_{\mathcal{O}},J} \subseteq q^D_{\mathcal{S}}$) for each $\mathcal{S}$-database D for which $\langle J, D\rangle$ is consistent. The next proposition establishes a precise relationship between the notion of separation in OBDM introduced here and the notion of abstraction.

Proposition 1. *Let $\Sigma = \langle J, D\rangle$ be a consistent OBDM system, λ^+ and λ^- be two D-datasets, and $q_{\mathcal{S}}$ be a query that explains λ^+ and λ^- inside D. If a query $q_{\mathcal{O}} \in \mathcal{Q}$ is a perfect (resp., complete, sound) J-abstraction of $q_{\mathcal{S}}$, then $q_{\mathcal{O}}$ is a proper (resp., complete, sound) Σ-separation of λ^+ and λ^- in $\mathcal{Q}$.*

Proof. Suppose $q_{\mathcal{O}} \in \mathcal{Q}$ is a perfect (resp., complete, sound) J-abstraction of $q_{\mathcal{S}}$, i.e., $cert^{D'}_{q_{\mathcal{O}},J} = q^{D'}_{\mathcal{S}}$ (resp., $q^{D'}_{\mathcal{S}} \subseteq cert^{D'}_{q_{\mathcal{O}},J}$, $cert^{D'}_{q_{\mathcal{O}},J} \subseteq q^{D'}_{\mathcal{S}}$) for each $\mathcal{S}$-database D' for which $\langle J, D'\rangle$ is consistent. Since by assumption $\Sigma = \langle J, D\rangle$ is consistent, we have that $cert^D_{q_{\mathcal{O}},J} = q^D_{\mathcal{S}}$ (resp., $q^D_{\mathcal{S}} \subseteq cert^D_{q_{\mathcal{O}},J}$, $cert^D_{q_{\mathcal{O}},J} \subseteq q^D_{\mathcal{S}}$). Now, since both $\lambda^+ \subseteq q^D_{\mathcal{S}}$ and $q^D_{\mathcal{S}} \cap \lambda^- = \emptyset$ hold by the assumption that $q_{\mathcal{S}}$ explains λ^+ and λ^- inside D, we derive that both $\lambda^+ \subseteq cert^D_{q_{\mathcal{O}},J}$ and $cert^D_{q_{\mathcal{O}},J} \cap \lambda^- = \emptyset$ (resp., only $\lambda^+ \subseteq cert^D_{q_{\mathcal{O}},J}$, only $cert^D_{q_{\mathcal{O}},J} \cap \lambda^- = \emptyset$) hold, which means that $q_{\mathcal{O}}$ is a proper (resp., complete, sound) Σ-separation of λ^+ and λ^- in $\mathcal{Q}$. □

The next example shows that the converse of the above proposition does not necessarily hold, thus stressing the fact that the two problems are indeed different.

Example 5. *Let $\Sigma = \langle J, D\rangle$ be as follows. $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ is the OBDM specification in which $\mathcal{O} = \emptyset$, i.e., $\mathcal{O}$ contains no assertions, $\mathcal{S} = \{s_1, s_2, s_3\}$, and $\mathcal{M}$ consists of the following two GAV assertions:*

$$\{(x) \mid s_1(x) \wedge s_2(x)\} \rightarrow \{(x) \mid \mathit{Student}(x)\}$$
$$\{(x) \mid s_3(x)\} \rightarrow \{(x) \mid \mathit{Student}(x)\}$$

Let also the $\mathcal{S}$-database be $D = \{s_1(a), s_2(a), s_2(b)\}$ and the D-datasets λ^+ and λ^- be $\lambda^+ = \{(a)\}$ and $\lambda^- = \{(b)\}$. Consider, moreover, the CQ $q_{\mathcal{S}} = \{(x) \mid s_1(x)\}$. One can see that the query $q_{\mathcal{S}}$ explains λ^+ and λ^- inside D because $q_{\mathcal{S}}^D = \{(a)\}$. Consider now the CQ $q_{\mathcal{O}} = \{(x) \mid \mathit{Student}(x)\}$ over $\mathcal{O}$. One can see that $q_{\mathcal{O}}$ is a proper (and therefore, also a complete and a sound) Σ-separation of λ^+ and λ^- in CQ because $cert^D_{q_{\mathcal{O}},J} = \{(a)\}$.

*Notice, however, that for the $\mathcal{S}$-database $D' = \{s_1(a), s_3(b)\}$ we have that (*i*) $\langle J, D'\rangle$ is a consistent OBDM system, (*ii*) $q_{\mathcal{S}}^{D'} = \{(a)\}$, and (*iii*) $cert^{D'}_{q_{\mathcal{O}},J} = \{(b)\}$. Thus, $q_{\mathcal{O}}$ is neither a complete nor a sound (and therefore, nor a perfect) J-abstraction of $q_{\mathcal{S}}$. Indeed both $q_{\mathcal{S}}^{D'} \not\subseteq cert^{D'}_{q_{\mathcal{O}},J}$ (witnessing that $q_{\mathcal{O}}$ is not a complete J-abstraction of $q_{\mathcal{S}}$) and $cert^{D'}_{q_{\mathcal{O}},J} \not\subseteq q_{\mathcal{S}}^{D'}$ (witnessing that $q_{\mathcal{O}}$ is not a sound J-abstraction of $q_{\mathcal{S}}$) hold.*

4.2. Computational Problems associated within the framework

Given the general framework introduced so far, there are (at least) three computational problems to consider, with respect to an ontology language $\mathcal{L}_{\mathcal{O}}$, a mapping language $\mathcal{L}_{\mathcal{M}}$, and a query language $\mathcal{Q}$. Given an OBDM system $\Sigma = \langle\langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle, D\rangle$ and two D-datasets λ^+ and λ^-, where $\mathcal{O} \in \mathcal{L}_{\mathcal{O}}$ and $\mathcal{M} \in \mathcal{L}_{\mathcal{M}}$:

- *Verification:* given also a query $q_{\mathcal{O}} \in \mathcal{Q}$, check whether $q_{\mathcal{O}}$ is a proper (resp., complete, sound) Σ-separation of λ^+ and λ^- in $\mathcal{Q}$.
- *Computation:* compute any proper in $\mathcal{Q}$ (resp., any $\mathcal{Q}$-minimally complete, any $\mathcal{Q}$-maximally sound) Σ-separation of λ^+ and λ^-, provided it exists.
- *Existence:* check whether there exists a query $q_{\mathcal{O}} \in \mathcal{Q}$ that is a proper in $\mathcal{Q}$ (resp., a $\mathcal{Q}$-minimally complete, a $\mathcal{Q}$-maximally sound) Σ-separation of λ^+ and λ^-.

Analogous computational problems can be defined when in input we have only one D-dataset λ^+ of arity n, rather than two D-datasets λ^+ and λ^-, and we implicitly think of λ^- as $dom(D)^n \setminus \lambda^+$.

In what follows, if not otherwise stated, we refer to the following scenario which considers by far the most popular languages for the OBDM paradigm: (*i*) $\mathcal{L}_{\mathcal{O}}$ is *DL-Lite*$_{\mathcal{R}}$, (*ii*) $\mathcal{L}_{\mathcal{M}}$ is GLAV, and (*iii*) $\mathcal{Q}$ is UCQ. In this scenario, there are two interesting properties that are worth mentioning.

Proposition 2. *Let $\Sigma = \langle J, D\rangle$ be an OBDM system, and λ^+ and λ^- be two D-datasets of arity n such that $\lambda^+ \cup \lambda^- = dom(D)^n$. If q_1 and q_2 are UCQ-minimally complete (resp., UCQ-maximally sound) Σ-separations of λ^+ and λ^-, then they are equivalent w.r.t. Σ.*

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 Σ-separations of λ^+ and λ^- and suppose, for the sake of contradiction, that they are not equivalent w.r.t. Σ. This implies the existence of a tuple $\vec{c}$ such that $\vec{c} \notin cert^D_{q_1,J}$ and $\vec{c} \in cert^D_{q_2,J}$. Observe that, since $\lambda^+ \cup \lambda^- = dom(D)^n$, $\vec{c}$ must be such that $\vec{c} \in \lambda^+$, otherwise q_2 would not be a sound Σ-separation of λ^+ and λ^- in UCQ, thus reaching a contradiction. But then, the UCQ $Q = q_1 \cup q_2$ is such that (*i*) since both q_1 and q_2 are sound Σ-separations of λ^+ and λ^- in UCQ, Q is a sound Σ-separation of λ^+ and λ^- in UCQ as well, and (*ii*) $cert^D_{q_1,J} \cap \lambda^+ \subset cert^D_{Q,J} \cap \lambda^+$ holds because of tuple $\vec{c}$. Obviously, this contradicts the fact that q_1 is a UCQ-maximally sound Σ-separation of λ^+ and λ^-.

Assume now that q_1 and q_2 are UCQ-minimally complete Σ-separations of λ^+ and λ^- and suppose, for the sake of contradiction, that they are not equivalent w.r.t. Σ. This implies the existence of a tuple $\vec{c}$ such that $\vec{c} \in cert^D_{q_1,J}$ and $\vec{c} \notin cert^D_{q_2,J}$. Observe that, since $\lambda^+ \cup \lambda^- = dom(D)^n$, $\vec{c}$ must be such that $\vec{c} \in \lambda^-$, otherwise q_2 would not be a complete Σ-separation of λ^+ and λ^- in UCQ, thus reaching a contradiction. But then, consider the query Q such that $cert^D_{Q,J} = cert^D_{q_1,J} \bigcap cert^D_{q_2,J}$. Obviously, since q_1 and q_2 are UCQs, Q exists and can be expressed as a UCQ. Moreover, (*i*) since q_1 and q_2 are complete Σ-separations of λ^+ and λ^- in UCQ, Q is a complete Σ-separation of λ^+

and λ^- in UCQ as well, and (*ii*) $cert^D_{Q,J} \cap \lambda^- \subset cert^D_{q_1,J} \cap \lambda^-$ holds because of tuple $\vec{c}$. Obviously, this contradicts the fact that q_1 is a UCQ-minimally complete Σ-separation of λ^+ and λ^-. □

This also means that, in our scenario, for an OBDM system $\Sigma = \langle J, D\rangle$ and two D-datasets λ^+ and λ^- of arity n such that $\lambda^+ \cup \lambda^- = dom(D)^n$, if a proper Σ-separation of λ^+ and λ^- in UCQ exists, then it is unique up to equivalence w.r.t. Σ. Furthermore, for the characterizability case where λ^- is implicitly set to be $dom(D)^n \setminus \lambda^+$, proper in UCQ, UCQ-minimally complete, and UCQ-maximally sound Σ-characterizations of λ^+ are always unique up to equivalence w.r.t. Σ, provided they exist.

Secondly, in this scenario, as one may expect, proper separations are less likely to exist than explanations in the plain relational database case.

Proposition 3. *Let $\Sigma = \langle J, D\rangle$ be a consistent OBDM system, and λ^+ and λ^- be two D-datasets. If there exists a proper Σ-separation of λ^+ and λ^- in UCQ, then λ^+ and λ^- are UCQ-explainable inside D.*

Proof. Suppose there exists a proper Σ-separation of λ^+ and λ^- in UCQ, i.e., there is a UCQ $q_\mathcal{O}$ over $\mathcal{O}$ for which $\lambda^+ \subseteq cert^D_{q_\mathcal{O},J}$ and $cert^D_{q_\mathcal{O},J} \cap \lambda^- = \emptyset$. Recall from Section 3 that the UCQ $\mathsf{rew}_{q_\mathcal{O},J}$ over $\mathcal{S}$ is such that $cert^{D'}_{q_\mathcal{O},J} = \mathsf{rew}^{D'}_{q_\mathcal{O},J}$ for each $\mathcal{S}$-database D' for which $\langle J, D'\rangle$ is consistent. Since $\Sigma = \langle J, D\rangle$ is consistent by assumption, we have that $cert^D_{q_\mathcal{O},J} = \mathsf{rew}^D_{q_\mathcal{O},J}$. Thus, $\mathsf{rew}_{q_\mathcal{O},J}$ is such that both $\lambda^+ \subseteq \mathsf{rew}^D_{q_\mathcal{O},J}$ and $\mathsf{rew}^D_{q_\mathcal{O},J} \cap \lambda^- = \emptyset$ hold, from which immediately follows that $\mathsf{rew}_{q_\mathcal{O},J}$ explains λ^+ and λ^- inside D by definition, and thus λ^+ and λ^- are UCQ-explainable inside D. □

In general, the converse of the above proposition does not hold. Indeed, in Example 2, while there is no proper Σ-separation of λ^+ and λ^- in $\mathcal{Q}$, whatever is the target query language $\mathcal{Q}$, the CQ $q_\mathcal{S} = \{(x) \mid s_1(x)\}$ witnesses that λ^+ and λ^- are CQ-definable inside D.

5. Verification

We now define the verification problems for X-separability (X-SEP) and X-characterization (X-CHAR), where X={Proper, Complete, Sound}. These decision problems are parametric with respect to the ontology language $\mathcal{L}_\mathcal{O}$ to express the ontology $\mathcal{O}$, the mapping language $\mathcal{L}_\mathcal{M}$ to express the mapping $\mathcal{M}$, and the target query language $\mathcal{Q}$ to express the query $q_\mathcal{O}$ over $\mathcal{O}$.

X-VSEP($\mathcal{L}_\mathcal{O}$, $\mathcal{L}_\mathcal{M}$, $\mathcal{Q}$)

Input: An OBDM system $\Sigma = \langle\langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle, D\rangle$, two D-datasets λ^+ and λ^-, and a query $q_\mathcal{O} \in \mathcal{Q}$ over $\mathcal{O}$, where $\mathcal{O} \in \mathcal{L}_\mathcal{O}$ and $\mathcal{M} \in \mathcal{L}_\mathcal{M}$.
Question: Is $q_\mathcal{O}$ a **X** Σ-separation of λ^+ and λ^- in $\mathcal{Q}$?

X-VCHAR($\mathcal{L}_\mathcal{O}$, $\mathcal{L}_\mathcal{M}$, $\mathcal{Q}$)

Input: An OBDM system $\Sigma = \langle\langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle, D\rangle$, a D-dataset λ^+, and a query $q_\mathcal{O} \in \mathcal{Q}$ over $\mathcal{O}$, where $\mathcal{O} \in \mathcal{L}_\mathcal{O}$ and $\mathcal{M} \in \mathcal{L}_\mathcal{M}$.
Question: Is $q_\mathcal{O}$ a **X** Σ-characterization of λ^+ in $\mathcal{Q}$?

We also introduce two important special cases of the above decision problems, namely: X-VSTSEP($\mathcal{L}_\mathcal{O}$, $\mathcal{L}_\mathcal{M}$, $\mathcal{Q}$) and X-VSTCHAR($\mathcal{L}_\mathcal{O}$, $\mathcal{L}_\mathcal{M}$, $\mathcal{Q}$), which are special cases of X-VSEP($\mathcal{L}_\mathcal{O}$, $\mathcal{L}_\mathcal{M}$, $\mathcal{Q}$) and X-VCHAR($\mathcal{L}_\mathcal{O}$, $\mathcal{L}_\mathcal{M}$, $\mathcal{Q}$), respectively, in which the all the input D-datasets are singleton sets (i.e., they consist of just a single tuple).

In what follows, given a syntactic object x such as a query, an ontology, or a mapping, we denote by $\sigma(x)$ its size.

We start by analyzing the upper bounds for the case X=Complete. The proof of the next theorem relies on the fact that, given an OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ of our considered scenario and a UCQ $q_\mathcal{O}$ over $\mathcal{O}$, the size of each disjunct in $\mathsf{PerfectRef}(\mathcal{O}, q_\mathcal{O})$ is at most $\sigma(q_\mathcal{O})$ [37], and the size of each disjunct in $\mathsf{MapRef}(\mathcal{M}, q_\mathcal{O})$ is at most $\sigma(\mathcal{M}) \cdot \sigma(q_\mathcal{O})$ [43].

Theorem 1. *Complete-V*SEP*(DL-Lite$_\mathcal{R}$, GLAV, UCQ) and Complete-V*CHAR*(DL-Lite$_\mathcal{R}$, GLAV, UCQ) are in NP.*

Proof. We address Complete-VSEP(*DL-Lite*$_\mathcal{R}$, GLAV, UCQ). The case of Complete-VCHAR(*DL-Lite*$_\mathcal{R}$, GLAV, UCQ) can be addressed in exactly the same way (recall that λ^- is immaterial for the complete case). In particular, we now show how to check in NP whether $q_\mathcal{O}$ is a complete Σ-separation of λ^+ and λ^- in UCQ (i.e., $\lambda^+ \subseteq \mathit{cert}^D_{q_\mathcal{O},J}$), where $\Sigma = \langle J, D\rangle$ with $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ in which $\mathcal{O}$ is a *DL-Lite*$_\mathcal{R}$ ontology and $\mathcal{M}$ is a GLAV mapping.

Let n be the arity of the tuple(s) in the D-datasets λ^+ and λ^-. For each n-tuple of constants $\vec{c} \in \lambda^+$, we first guess (*i*) a CQ $q'_\mathcal{O}$ over $\mathcal{O}$ which is either of arity n and size at most $\sigma(q_\mathcal{O})$, or a boolean one capturing a disjointness assertion d (e.g., $\{() \mid \exists y.A_1(y) \wedge A_2(y)\}$ capturing $d = A_1 \sqsubseteq \neg A_2$); (*ii*) a sequence $\rho_\mathcal{O}$ of ontology assertions; (*iii*) a CQ $q_\mathcal{S}$ over $\mathcal{S}$ of size at most $\sigma(\mathcal{M}) \cdot \sigma(q'_\mathcal{O})$ which is either of arity n and of the form $\{\vec{x} \mid \exists \vec{y}.\phi_\mathcal{S}(\vec{x},\vec{y})\}$, or a boolean one of the form $\{() \mid \exists \vec{y}.\phi_\mathcal{S}(\vec{y})\}$; (*iv*) a sequence $\rho_\mathcal{M}$ of mapping assertions; and (*v*) a function f from the variables occurring in $q_\mathcal{S}$ to $\mathit{dom}(D)$.

Then, we check in polynomial time whether (*i*) by means of $\rho_\mathcal{O}$, either we can rewrite a disjunct of $q_\mathcal{O}$ into $q'_\mathcal{O}$ through $\mathcal{O}$ (i.e., $q'_\mathcal{O} \in \mathsf{PerfectRef}(\mathcal{O}, q_\mathcal{O})$), or we can rewrite a disjunct of $V_\mathcal{O}$ into $q'_\mathcal{O}$ through $\mathcal{O}$ (i.e., $q'_\mathcal{O} \in \mathsf{PerfectRef}(\mathcal{O}, V_\mathcal{O})$); (*ii*) by means of $\rho_\mathcal{M}$, we can rewrite $q'_\mathcal{O}$ into $q_\mathcal{S}$ through $\mathcal{M}$ (i.e., $q_\mathcal{S} \in \mathsf{MapRef}(\mathcal{M}, q'_\mathcal{O})$, and thus either $q_\mathcal{S} \in \mathsf{rew}_{q_\mathcal{O},J}$ or $q_\mathcal{S} \in \mathsf{rew}_{V_\mathcal{O},J}$); and finally (*iii*) f consists in a homomorphism witnessing either $(\mathit{set}(\phi_\mathcal{S}), \vec{x}) \rightarrow (D, \vec{c})$, i.e., $\vec{c} \in q^D_\mathcal{S}$ (and therefore $\vec{c} \in \mathsf{rew}^D_{q_\mathcal{O},J}$, which means $\vec{c} \in \mathit{cert}^D_{q_\mathcal{O},J}$), or $(\mathit{set}(\phi_\mathcal{S}), ()) \rightarrow (D, ())$, i.e., $q^D_\mathcal{S} = \{\langle\rangle\}$ (and therefore $\mathsf{rew}^D_{V_\mathcal{O},J} = \{\langle\rangle\}$, which means that Σ is inconsistent and thus $\vec{c} \in \mathit{cert}^D_{q_\mathcal{O},J}$ by definition of certain answers). □

By exploiting the above result, we now address the upper bounds for the case *X*=Sound.

Theorem 2. *Sound-V*SEP*(DL-Lite$_\mathcal{R}$, GLAV, UCQ) and Sound-V*CHAR*(DL-Lite$_\mathcal{R}$, GLAV, UCQ) are in coNP.*

Proof. We start with Sound-VSEP(*DL-Lite*$_\mathcal{R}$, GLAV, UCQ), and then we consider Sound-VCHAR(*DL-Lite*$_\mathcal{R}$, GLAV, UCQ). In particular, we now show how to check in NP whether $q_\mathcal{O}$ is not a sound Σ-separation of λ^+ and λ^- in UCQ (i.e., $\mathit{cert}^D_{q_\mathcal{O},J} \cap \lambda^- \neq \emptyset$), where $\Sigma = \langle J, D\rangle$ with $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ in which $\mathcal{O}$ is a *DL-Lite*$_\mathcal{R}$ ontology and $\mathcal{M}$ is a GLAV mapping.

We first guess (*i*) a tuple of constants $\vec{c}$, and, exactly as in the proof of Theorem 1, we also guess (*ii*) $q'_\mathcal{O}$, $\rho_\mathcal{O}$, $q_\mathcal{S}$, $\rho_\mathcal{M}$, and f. Then, we check in polynomial time whether (*i*) $\vec{c} \in \lambda^-$, and (*ii*) using $q'_\mathcal{O}$, $\rho_\mathcal{O}$, $q_\mathcal{S}$, $\rho_\mathcal{M}$, and f, we follow exactly the same polynomial time procedure described in the proof of Theorem 1 to check whether $\vec{c} \in \mathit{cert}^D_{q_\mathcal{O},J}$.

As for the Sound-VCHAR(*DL-Lite*$_\mathcal{R}$, GLAV, UCQ) case, it is sufficient to replace the check (*i*) $\vec{c} \in \lambda^-$ with the check $\vec{c} \in \mathit{dom}(D)^n \setminus \lambda^+$, where n is the arity of the tuple(s) in the D-dataset λ^+. Clearly this latter check can be done in polynomial time as well, by first checking that every constant of $\vec{c}$ effectively occurs in $\mathit{dom}(D)$ and then simply checking that $\vec{c} \notin \lambda^+$. □

We recall that the complexity class DP, introduced in [50], resides at the second level of the polynomial time hierarchy [51], and it is composed of all those decision problems that are the *intersection* of a decision problem in NP and a decision problem in coNP, that is, DP = NP $\wedge$ coNP = $\{P_1 \cap P_2 \mid P_1 \in \text{NP} \wedge P_2 \in \text{coNP}\}$.

Since $q_\mathcal{O}$ is a proper Σ-separation of λ^+ and λ^- in $\mathcal{Q}$ if and only if it is both a sound, and a complete Σ-separation of λ^+ and λ^- in $\mathcal{Q}$, we immediately derive the following upper bounds for *X*=Proper.

Corollary 1. *Proper-V*SEP*(DL-Lite$_\mathcal{R}$, GLAV, UCQ) and Proper-V*CHAR*(DL-Lite$_\mathcal{R}$, GLAV, UCQ) are in DP.*

We now provide matching lower bounds, proving that all of them already hold for the singleton datasets special cases. More specifically, we show that they already hold for the same fixed OBDM system $\Sigma = \langle J, D\rangle$, same fixed D-datasets λ^+ and λ^- (resp., D-dataset λ^+) containing only a single unary tuple, and for CQs as queries. Furthermore, the fixed OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ is such that $\mathcal{O} = \emptyset$ (i.e., $\mathcal{O}$ contains no ontology assertions) and $\mathcal{M}$ is a GAV$\cap$LAV mapping (i.e., $\mathcal{M}$ is both a GAV and a LAV mapping). To simplify the presentation, with a slight abuse of notation, from now on we denote by $\mathcal{L}_\mathcal{O} = \emptyset$ the ontology language allowing only for ontologies $\mathcal{O} = \emptyset$, i.e., ontologies $\mathcal{O}$ without assertions.

Theorem 3. *There is an OBDM system $\Sigma = \langle\langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle, D\rangle$ such that $\mathcal{O} = \emptyset$ and $\mathcal{M}$ is a GAV$\cap$LAV mapping, and two D-datasets λ^+ and λ^- (resp., a D-dataset λ^+) containing only one unary tuple for which:*

- *Complete-VST*SEP*(∅, GAV∩LAV, CQ) and Complete-VST*CHAR*(∅, GAV∩LAV, CQ) are NP-hard;*
- *Sound-VST*SEP*(∅, GAV∩LAV, CQ) and Sound-VST*CHAR*(∅, GAV∩LAV, CQ) are coNP-hard;*
- *Proper-VST*SEP*(∅, GAV∩LAV, CQ) and Proper-VST*CHAR*(∅, GAV∩LAV, CQ)) are DP-hard.*

Proof. Let $\Sigma = \langle J, D\rangle$ be the fixed OBDM system such that (*i*) $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ is an OBDM specification in which $\mathcal{O}$ contains no assertions and whose alphabet consists of two atomic roles P_1 and P_2, $\mathcal{S} = \{s_1, s_2\}$, and $\mathcal{M}$ consists of the following two GAV∩LAV assertions: $\{(x_1, x_2) \mid s_1(x_1, x_2)\} \rightarrow \{(x_1, x_2) \mid P_1(x_1, x_2)\}$ and $\{(x_1, x_2) \mid s_2(x_1, x_2)\} \rightarrow \{(x_1, x_2) \mid P_2(x_1, x_2)\}$, which simply mirrors source predicate s_i to atomic role P_i, for $i = [1, 2]$, and (*ii*) D is the $\mathcal{S}$-database composed of the following facts:

$$\{s_1(x, y) \mid x = \{r', g', b'\} \text{ and } y = \{r', g', b'\} \text{ and } x \neq y\} \cup$$
$$\{s_1(x, y) \mid x = \{r, g, b, o\} \text{ and } y = \{r, g, b, o\} \text{ and } x \neq y\} \cup$$
$$\{s_2(x, c_3) \mid x = \{r', g', b'\}\} \cup \{s_2(x, c_4) \mid x = \{r, g, b, o\}\}.$$

Let, moreover, λ^+ and λ^- be the fixed D-datasets $\lambda^+ = \{(c_4)\}$ and $\lambda^- = \{(c_3)\}$.

Let $G = (V, E)$ be a finite and undirected graph without loops[2] or isolated nodes, where $V = \{y_1, \ldots, y_n\}$. We define a CQ $q_G = \{(x) \mid \exists \vec{y}.\phi_{\mathcal{O}}(x, \vec{y})\}$ over $\mathcal{O}$ as follows:

$$\{(x) \mid \exists y_1, \ldots, y_n. \bigwedge_{(y_i, y_j) \in E} (P_1(y_i, y_j)) \wedge \bigwedge_{y_i \in V} (P_2(y_i, x))\}$$

Notice that q_G can be constructed in LOGSPACE from an input graph G as above.

By inspecting the OBDM system $\Sigma = \langle J, D\rangle$, for any graph G as above, the set of certain answers $cert^D_{q_G, J}$ must necessarily be an element of the power set of $\{(c_3), (c_4)\}$. More specifically, the following property holds:

Claim 1. *For both $i = 3$ and $i = 4$, we have that a graph $G = (V, E)$ is i-colourable if and only if $(c_i) \in cert^D_{q_G, J}$.*

Proof. First of all, notice that $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ is composed of the following facts:

$$\{P_1(x, y) \mid x = \{r', g', b'\} \text{ and } y = \{r', g', b'\} \text{ and } x \neq y\} \cup$$
$$\{P_1(x, y) \mid x = \{r, g, b, o\} \text{ and } y = \{r, g, b, o\} \text{ and } x \neq y\} \cup$$
$$\{P_2(x, c_3) \mid x = \{r', g', b'\}\} \cup \{P_2(x, c_4) \mid x = \{r, g, b, o\}\}.$$

"**Only-if part:**" Suppose $G = (V, E)$ is 3-colourable (resp., 4-colourable), that is, there exists a function $f : V \rightarrow \{r', g', b'\}$ (resp., $f : V \rightarrow \{r, g, b, o\}$) such that $f(y_i) \neq f(y_j)$ for each $(y_i, y_j) \in E$. Let $\phi_{\mathcal{O}}$ be the body of q_G, and consider the extension of f which assigns to the distinguished variable x of q_G the constant c_3 (resp., c_4). It can be readily seen that f consists in a homomorphism from $set(\phi_{\mathcal{O}})$ to $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}$ such that $f(x) = c_3$ (resp., $f(x) = c_4$). In other words, f witnesses that $(set(\phi_{\mathcal{O}}), (x)) \rightarrow (\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}, (c_3))$ (resp., $(set(\phi_{\mathcal{O}}), (x)) \rightarrow (\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}, (c_4))$). Thus, $(c_3) \in cert^D_{q_G, J}$ (resp., $(c_4) \in cert^D_{q_G, J}$), as required.

"**If part:**" Suppose $G = (V, E)$ is not 3-colourable (resp., not 4-colourable), that is, each possible function $f : V \rightarrow \{r', g', b'\}$ (resp., $f : V \rightarrow \{r, g, b, o\}$) is such that $f(y_i) = f(y_j)$ for some $(y_i, y_j) \in E$. Clearly, this implies that $(set(\phi_{\mathcal{O}}), (x)) \not\rightarrow (\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}, (c_3))$ (resp., $(set(\phi_{\mathcal{O}}), (x)) \not\rightarrow (\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}, (c_4))$). Thus, $(c_3) \notin cert^D_{q_G, J}$ (resp., $(c_4) \notin cert^D_{q_G, J}$), as required. □

With the above property in hand, and the fact that $cert^D_{q_G, J}$ is an element of the power set of $\{(c_3), (c_4)\}$ for each possible graph $G = (V, E)$, we are now ready to prove the claimed lower bounds.

[2] In graph theory, a loop is an edge that connects a vertex with itself [52].

As for the complete case, the proof of NP-hardness is by a LOGSPACE reduction from the *4-colourability problem*, which is NP-complete [53]. In particular, from the above claim a graph G is 4-colourable if and only if $(c_4) \in cert^D_{q_G,J}$, i.e., if and only if $\lambda^+ \subseteq cert^D_{q_G,J}$, which is the condition for q_G to be a complete Σ-separation of λ^+ and λ^- in CQ (resp., a complete Σ-characterization of λ^+ in CQ).

As for the sound case, the proof of coNP-hardness is by a LOGSPACE reduction from the *complement of the 3-colourability problem*, which is coNP-complete [53]. In particular, from the above claim a graph G is not 3-colourable if and only if $(c_3) \notin cert^D_{q_G,J}$, i.e., if and only if $cert^D_{q_G,J} \cap \lambda^- = \emptyset$ (resp., $cert^D_{q_G,J} \subseteq \lambda^+$), which is the condition for q_G to be a sound Σ-separation of λ^+ and λ^- in CQ (resp., a sound Σ-characterization of λ^+ in CQ).

Finally, as for the proper case, the proof of DP-hardness is by a LOGSPACE reduction from the *exact-4-colourability problem*, which is DP-complete [54]. In particular, a graph G is exact-4-colourable (i.e., 4-colourable and not 3-colourable) if and only if $cert^D_{q_G,J} = \{(c_4)\}$, i.e., if and only if $cert^D_{q_G,J} \subseteq \lambda^+$ and $cert^D_{q_G,J} \cap \lambda^- = \emptyset$ (resp., $cert^D_{q_G,J} = \lambda^+$), which is the condition for q_G to be a proper Σ-separation of λ^+ and λ^- in CQ (resp., a proper Σ-characterization of λ^+ in CQ). □

For the scenario under investigation in this paper, we can now establish the precise computational complexity of all the Verification decision problems introduced at the beginning of this section.

Corollary 2. *The following holds:*

- *Complete-V*SEP*(DL-Lite$_\mathcal{R}$, GLAV, UCQ) and Complete-V*CHAR*(DL-Lite$_\mathcal{R}$, GLAV, UCQ) are NP-complete. The hardnesses already hold for Complete-VST*SEP*(∅, GAV∩LAV, CQ) and Complete-VST*CHAR*(∅, GAV∩LAV, CQ);*
- *Sound-V*SEP*(DL-Lite$_\mathcal{R}$, GLAV, UCQ) and Sound-V*CHAR*(DL-Lite$_\mathcal{R}$, GLAV, UCQ) are coNP-complete. The hardnesses already hold for Sound-VST*SEP*(∅, GAV∩LAV, CQ) and Sound-VST*CHAR*(∅, GAV∩LAV, CQ);*
- *Proper-V*SEP*(DL-Lite$_\mathcal{R}$, GLAV, UCQ) and Proper-V*CHAR*(DL-Lite$_\mathcal{R}$, GLAV, UCQ) are DP-complete. The hardnesses already hold for Proper-VST*SEP*(∅, GAV∩LAV, CQ) and Proper-VST*CHAR*(∅, GAV∩LAV, CQ).*

Finally, from the lower bounds given in Theorem 3, we can derive two interesting novel results also in the context of explainability and definability in the plain relational database case. More specifically, since the fixed OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ used in the proof that theorem is such that $\mathcal{O} = \emptyset$ and $\mathcal{M}$ is both a GAV and a LAV mapping, the proof can be straightforwardly adapted also for the plain relational database case. Thus, given a schema $\mathcal{S}$, an $\mathcal{S}$-database D, two D-datasets λ^+ and λ^- (resp., a D-dataset λ^+), and a UCQ $q_\mathcal{S}$ over $\mathcal{S}$, it is DP-complete the problem of deciding whether $q_\mathcal{S}$ explains λ^+ and λ^- (resp., defines λ^+) inside D (the DP membership of these decision problems directly follows from Corollary 1).

6. Computation

In this section, we address the Computation problem. We start by considering the case when the given OBDM system Σ at hand is inconsistent. Given an inconsistent OBDM system $\Sigma = \langle J, D \rangle$ with $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and two D-datasets λ^+ and λ^- (resp., a D-dataset λ^+) of arity n, we point out that any query $q_\mathcal{O} \in \mathcal{Q}$ of arity n over $\mathcal{O}$ is a $\mathcal{Q}$-minimally complete Σ-separation (resp., Σ-characterization) of λ^+ and λ^- (resp., of λ^+). This is so because, by definition, the certain answers of any query $q_\mathcal{O}$ of arity n w.r.t. an inconsistent OBDM system Σ is the set of all possible n-tuples of constants occurring in D, i.e., $dom(D)^n$. Furthermore, if $\lambda^+ = dom(D)^n$ and $\lambda^- = \emptyset$, then any query $q_\mathcal{O} \in \mathcal{Q}$ of arity n over $\mathcal{O}$ is also a $\mathcal{Q}$-maximally sound (and therefore a proper in $\mathcal{Q}$) Σ-separation (resp., Σ-characterization) of λ^+ and λ^- (resp., of λ^+); otherwise, no sound (and therefore, no $\mathcal{Q}$-maximally sound and no proper in $\mathcal{Q}$) Σ-separation (resp., Σ-characterization) of λ^+ and λ^- (resp., of λ^+) exists. Since, however, for OBDM systems of our scenario it is always possible to check whether they are inconsistent or not (cf. Section 3), from the above observations one can trivially derive suitable algorithms for the Computation problem in all the cases in which the input OBDM system Σ is inconsistent.

Having thoroughly covered the case of inconsistent OBDM systems, in what follows in this section, unless otherwise stated, we implicitly assume to deal only with consistent OBDM systems.

Specifically, we now provide two exponential time algorithms that, given a consistent OBDM system $\Sigma = \langle J, D\rangle$ and two D-datasets λ^+ and λ^- (resp., a D-dataset λ^+), return, respectively, a UCQ-minimally complete and a UCQ-maximally sound Σ-separation (resp., Σ-characterization) of λ^+ and λ^- (resp., of λ^+), thus proving that, in our investigated scenario, they always exist. In fact, the algorithms we provide focus only on the Separability case. Algorithms for the Characterizability case can be immediately derived from the ones we provide by simply setting $\lambda^- = dom(D)^n \setminus \lambda^+$, where n is the arity of the tuple(s) in λ^+.

Before delving into the details of the two algorithms, we provide some crucial properties about the canonical structure that will be used to establish the correctness of such algorithms.

Proposition 4. *Let $\Sigma = \langle J, D\rangle$ with $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ be an OBDM system, $q_\mathcal{O}$ be a UCQ over $\mathcal{O}$, and $\vec{c}$ and $\vec{b}$ be two tuples of constants such that $(\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{c}) \to (\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{b})$. If $\vec{c} \in cert^D_{q_\mathcal{O},J}$, then $\vec{b} \in cert^D_{q_\mathcal{O},J}$.*

Proof. If Σ is inconsistent, the claim is trivial. If Σ is consistent, from Section 3 we know that $\vec{c} \in cert^D_{q_\mathcal{O},J}$ implies the existence of a disjunct $q = \{\vec{x} \mid \exists\vec{y}.\phi(\vec{x},\vec{y})\}$ in $q_\mathcal{O}$ for which $(set(\phi), \vec{x}) \to (\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{c})$. Let h be the homomorphism witnessing that $(set(\phi), \vec{x}) \to (\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{c})$, and let h' be the homomorphism witnessing that $(\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{c}) \to (\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{b})$, which exists by the premises of the proposition. The composite function $h'' = h' \circ h$ is then a homomorphism witnessing that $(set(\phi), \vec{x}) \to (\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{b})$. It follows that $\vec{b} \in cert^D_{q_\mathcal{O},J}$, as required. □

Proposition 5. *Let $\Sigma = \langle J, D\rangle$ with $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ be a consistent OBDM system, $\vec{b}$ and $\vec{c}$ be two tuples of constants, and $q_{\vec{c}}$ be the CQ $q_{\vec{c}} = query(\mathcal{M}(D), \vec{c})$ over $\mathcal{O}$. We have that $\vec{b} \in cert^D_{q_{\vec{c}},J}$ if and only if $(\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{c}) \to (\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{b})$.*

Proof. Suppose that $(\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{c}) \to (\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{b})$, and let h be the homomorphism witnessing this. Consider the query $q_{\vec{c}} = query(\mathcal{M}(D), \vec{c}) = \{\vec{x} \mid \exists\vec{y}.\phi(\vec{x},\vec{y})\}$. Observe that $set(\phi)$ is obtained from $\mathcal{M}(D)$ by appropriately replacing each occurrence of each constant $c \in dom(\mathcal{M}(D))$ either with a distinguished variable $x_c \in \vec{x}$ or with an existential variable $y_c \in \vec{y}$. This means that h can be immediately transformed into a homomorphism witnessing that $(set(\phi), \vec{x}) \to (\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{b})$, thus implying that $\vec{b} \in cert^D_{q_{\vec{c}},J}$.

Suppose now that $\vec{b} \in cert^D_{q_{\vec{c}},J}$. Since Σ is consistent, it follows that there is a homomorphism h witnessing that $(set(\phi), \vec{x}) \to (\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{b})$, where $q_{\vec{c}} = query(\mathcal{M}(D), \vec{c}) = \{\vec{x} \mid \exists\vec{y}.\phi(\vec{x},\vec{y})\}$. By considering again the relationship between $set(\phi)$ and $\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}$, the homomorphism h can be immediately transformed into a homomorphism h' that witnesses $(\mathcal{M}(D), \vec{c}) \to (\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{b})$. By construction of the canonical structure $\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}$, it is now trivial to verify that h' can be properly extended into a homomorphism h'' witnessing that $(\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{c}) \to (\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)}, \vec{b})$. □

We are now ready to present our algorithms for the Computation problem. We start with the complete case, and provide the algorithm MinCompSeparation for computing UCQ-minimally complete separations.

Algorithm MinCompSeparation

Input: Consistent OBDM system $\Sigma = \langle J, D\rangle$ with $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$; D-dataset $\lambda^+ = \{\vec{c_1}, \ldots, \vec{c_m}\}$; D-dataset λ^-
Output: UCQ $q_\mathcal{O}$ over $\mathcal{O}$

1: Compute $\mathcal{M}(D)$
2: $q_\mathcal{O} \leftarrow query(\mathcal{M}(D), \vec{c_1}) \cup \ldots \cup query(\mathcal{M}(D), \vec{c_m})$
3: **return** $q_\mathcal{O}$

Informally, for each positive example $\vec{c_i} \in \lambda^+$, the algorithm obtains from the set of atoms $\mathcal{M}(D)$ the CQ $query(\mathcal{M}(D), \vec{c_i})$. Then, the output query $q_\mathcal{O}$ is the union of all the CQs obtained in such a way.

Example 6. *Let $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be the same OBDM specification of Example 1. One can verify that for the $\mathcal{S}$-database $D = \{s_1(c_1), s_3(c_2, b), s_3(c_3, b)\}$ and the D-datasets $\lambda^+ = \{(c_1), (c_2)\}$ and $\lambda^- = \{(c_3)\}$, MinCompSeparation($\langle J, D \rangle, \lambda^+, \lambda^-$) returns the UCQ $q_\mathcal{O} = query(\mathcal{M}(D), (c_1)) \cup query(\mathcal{M}(D), (c_2))$, where $query(\mathcal{M}(D), (c_1)) = \{(x_{c_1}) \mid \exists y_{c_2}, y_{c_3}, y_b.\mathsf{Student}(x_{c_1}) \wedge \mathsf{EnrolledIn}(y_{c_2}, y_b) \wedge \mathsf{EnrolledIn}(y_{c_3}, y_b)\}$ and $query(\mathcal{M}(D), (c_2)) = \{(x_{c_2}) \mid \exists y_{c_1}, y_{c_3}, y_b.\mathsf{EnrolledIn}(x_{c_2}, y_b) \wedge \mathsf{EnrolledIn}(y_{c_3}, y_b) \wedge \mathsf{Student}(y_{c_1})\}$. Note that the query $q_\mathcal{O}$ returned by the algorithm is the UCQ-minimally complete Σ-separation of λ^+ and λ^-, where $\Sigma = \langle J, D \rangle$.*

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

Theorem 4. *Let $\Sigma = \langle J, D \rangle$ with $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ be a consistent OBDM system and λ^+ and λ^- be two D-datasets. We have that MinCompSeparation($\Sigma, \lambda^+, \lambda^-$) terminates and returns a UCQ-minimally complete Σ-separation of λ^+ and λ^-.*

Proof. Termination of the algorithm as well as completeness of the UCQ $q_\mathcal{O}$ returned are straightforward. In particular, due to Proposition 5, it is obvious that each $\vec{c_i} \in \lambda^+$ is such that $\vec{c_i} \in cert^D_{q_{\vec{c_i}}, J}$, where $q_{\vec{c_i}} = query(\mathcal{M}(D), \vec{c_i})$.

To prove that $q_\mathcal{O}$ is also a UCQ-minimally complete Σ-separation of λ^+ and λ^-, note that it is enough to show that any query q over $\mathcal{O}$ that is a complete Σ-separation of λ^+ and λ^- in UCQ is such that $cert^D_{q_\mathcal{O}, J} \subseteq cert^D_{q, J}$. We do this by contraposition. Let q be a UCQ over $\mathcal{O}$ for which $cert^D_{q_\mathcal{O}, J} \not\subseteq cert^D_{q, J}$, i.e., for a tuple of constants $\vec{b}$ we have $\vec{b} \notin cert^D_{q, J}$ but $\vec{b} \in cert^D_{q_\mathcal{O}, J}$. This latter means that $\vec{b} \in cert^D_{q_{\vec{c_i}}, J}$ for some $q_{\vec{c_i}} = query(\mathcal{M}(D), \vec{c_i})$ with $\vec{c_i} \in \lambda^+$ occurring in $q_\mathcal{O}$. By Proposition 5, one can see that $\vec{b} \in cert^D_{q_{\vec{c_i}}, J}$ implies $(\mathcal{C}^{\mathcal{M}(D)}_\mathcal{O}, \vec{c_i}) \rightarrow (\mathcal{C}^{\mathcal{M}(D)}_\mathcal{O}, \vec{b})$. By Proposition 4, it follows that each UCQ q' over $\mathcal{O}$ containing the tuple $\vec{c_i}$ in its set of certain answers w.r.t. Σ must contain also tuple $\vec{b}$ in such a set, i.e., $\vec{c_i} \in cert^D_{q', J}$ implies $\vec{b} \in cert^D_{q', J}$ for any UCQ q' over $\mathcal{O}$. Thus, since $\vec{b} \notin cert^D_{q, J}$, we derive that $\vec{c_i} \notin cert^D_{q, J}$ as well. Now, since $\vec{c_i} \in \lambda^+$ and $\vec{c_i} \notin cert^D_{q, J}$, this immediately implies that q is not a complete Σ-separation of λ^+ and λ^- in UCQ, as required. □

We now turn to the sound case, and provide the algorithm MaxSoundSeparation for computing UCQ-maximally sound Σ-separations.

Algorithm MaxSoundSeparation

Input: Consistent OBDM system $\Sigma = \langle J, D \rangle$ with $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$; D-dataset $\lambda^+ = \{\vec{c_1}, \ldots, \vec{c_m}\}$ of arity n; D-dataset λ^- of arity n
Output: UCQ $q_\mathcal{O}$ over $\mathcal{O}$

1: $q_\mathcal{O} \leftarrow \{(x_1, \ldots, x_n) \mid \bot(x_1) \wedge \ldots \wedge \bot(x_n)\}$, where $\vec{x} = (x_1, \ldots, x_n)$
2: Compute $\mathcal{M}(D)$
3: **for** each $i \leftarrow 1, \ldots, m$ **do**
4: $\quad q_{\vec{c_i}} \leftarrow query(\mathcal{M}(D), \vec{c_i})$
5: $\quad$ **if** $cert^D_{q_{\vec{c_i}}, J} \cap \lambda^- = \emptyset$ **then**
6: $\quad\quad q_\mathcal{O} \leftarrow q_\mathcal{O} \cup q_{\vec{c_i}}$
7: $\quad$ **end if**
8: **end for**
9: **return** $q_\mathcal{O}$

Intuitively, starting from the query $query(\mathcal{M}(D), \vec{c_1}) \cup \ldots \cup query(\mathcal{M}(D), \vec{c_m})$, which is a UCQ-minimally complete Σ-separation of λ^+ and λ^-, the algorithm simply discards all those disjuncts whose set of certain answers w.r.t. Σ contain at least a tuple $\vec{b} \in \lambda^-$. We recall from Section 3 that the set of certain answers of a CQ $q_{\vec{c_i}}$ w.r.t. a consistent OBDM system $\Sigma = \langle J, D \rangle$ can be always computed by first obtaining its reformulation $\mathsf{rew}_{q_{\vec{c_i}}, J}$ over the source schema $\mathcal{S}$, and then by evaluating this latter query directly over the $\mathcal{S}$-database D. In other words, the if-condition of line 5 can be equivalently reformulated as: $\mathsf{rew}^D_{q_{\vec{c_i}}, J} \cap \lambda^- = \emptyset$.

Example 7. *Refer to Example 6. Since the certain answers of the CQ query*$(\mathcal{M}(D),(c_2))$ *w.r.t.* $\Sigma = \langle J, D\rangle$ *include also* $(c_3) \in \lambda^-$*, MaxSoundSeparation*$(\Sigma,\lambda^+,\lambda^-)$ *returns the CQ* $q_\mathcal{O} = query(\mathcal{M}(D),(c_1))$*. Note that* $q_\mathcal{O}$ *is the UCQ-maximally sound* Σ*-separation of* λ^+ *and* λ^-*.*

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

Theorem 5. *Let* $\Sigma = \langle J, D\rangle$ *with* $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ *be a consistent OBDM system, and* λ^+ *and* λ^- *be two D-datasets. We have that MaxSoundSeparation*$(\Sigma,\lambda^+,\lambda^-)$ *terminates and returns a UCQ-maximally sound* Σ*-separation of* λ^+ *and* λ^-*.*

Proof. Termination of the algorithm as well as soundness of the UCQ $q_\mathcal{O}$ returned are straightforward. In particular, by construction, all the disjuncts $q_{\vec{c}_i}$ of $q_\mathcal{O}$ are such that there is no tuple $\vec{b} \in \lambda^-$ for which $\vec{b} \in cert^D_{q_{\vec{c}_i},J}$.

To prove that $q_\mathcal{O}$ is also a UCQ-maximally sound Σ-separation of λ^+ and λ^-, note that it is enough to show that any query q over $\mathcal{O}$ that is a sound Σ-separation of λ^+ and λ^- in UCQ is such that $cert^D_{q,J} \cap \lambda^+ \subseteq cert^D_{q_\mathcal{O},J} \cap \lambda^+$. We do this by contraposition. Let q be a UCQ over $\mathcal{O}$ for which $cert^D_{q,J} \cap \lambda^+ \not\subseteq cert^D_{q_\mathcal{O},J} \cap \lambda^+$, i.e., for a tuple of constants $\vec{b} \in \lambda^+$ we have $\vec{b} \in cert^D_{q,J}$ but $\vec{b} \notin cert^D_{q_\mathcal{O},J}$. Since $\vec{b} \notin cert^D_{q_\mathcal{O},J}$ and $\vec{b} \in \lambda^+$, it is easy to see that the algorithm discarded the disjunct $q_{\vec{b}} = query(\mathcal{M}(D),\vec{b})$ (otherwise, we would trivially derive that $\vec{b} \in cert^D_{q_{\vec{b}},J}$, and thus $\vec{b} \in cert^D_{q_\mathcal{O},J}$, which is a contradiction to the fact that $\vec{b} \notin cert^D_{q_\mathcal{O},J}$). By construction of the algorithm, one can see that the only reason $q_{\vec{b}}$ was discarded is because $\vec{g} \in cert^D_{q_{\vec{b}},J}$ for at least a tuple $\vec{g} \in \lambda^-$. By Proposition 5, one can see that $\vec{g} \in cert^D_{q_{\vec{b}},J}$ implies $(\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)},\vec{b}) \to (\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)},\vec{g})$. By Proposition 4, it follows that each UCQ q' over $\mathcal{O}$ containing tuple $\vec{b}$ in its set of certain answers w.r.t. Σ must contain also tuple $\vec{g}$ in such a set, i.e., $\vec{b} \in cert^D_{q',J}$ implies $\vec{g} \in cert^D_{q',J}$ for any UCQ q' over $\mathcal{O}$. Thus, since $\vec{b} \in cert^D_{q,J}$, we derive that $\vec{g} \in cert^D_{q,J}$ as well. Now, since $\vec{g} \in \lambda^-$ and $\vec{g} \in cert^D_{q,J}$, this immediately implies that q is not a sound Σ-separation of λ^+ and λ^- in UCQ, as required. □

Notice that, in all the cases in which a proper separation exists, it is clear that both the above algorithms return the same query $query(\mathcal{M}(D),\vec{c_1}) \cup \ldots \cup query(\mathcal{M}(D),\vec{c_m})$. Therefore, as a direct consequence of both Theorem 4 and Theorem 5, we get the following result.

Corollary 3. *Let* $\Sigma = \langle\langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle, D\rangle$ *be an OBDM specification, and* $\lambda^+ = \{\vec{c_1},\ldots,\vec{c_m}\}$ *and* λ^- *two D-datasets. Either the UCQ* $q_\mathcal{O} = query(\mathcal{M}(D),\vec{c_1}) \cup \ldots \cup query(\mathcal{M}(D),\vec{c_m})$ *is a proper* Σ*-separation of* λ^+ *and* λ^- *in UCQ, or no proper* Σ*-separation of* λ^+ *and* λ^- *in UCQ exists.*

Finally, the combination of Corollary 3 and Proposition 5 allows us to provide the following semantic tests for the existence of proper separations and proper characterizations in UCQ in the OBDM setting, which can be seen as the analogous of the semantic tests given in [11] and [21] for the plain relational database setting and the ontology-enriched query answering setting, respectively.

- SEP test for UCQs in OBDM: given a consistent OBDM system $\Sigma = \langle\langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle, D\rangle$ and two D-datasets λ^+ and λ^-, there exists a proper Σ-separation of λ^+ and λ^- in UCQ if and only if it is the case that $(\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)},\vec{c}) \not\to (\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)},\vec{b})$ for all $\vec{c} \in \lambda^+$ and all $\vec{b} \in \lambda^-$.
- CHAR test for UCQs in OBDM: given a consistent OBDM system $\Sigma = \langle\langle \mathcal{O}, \mathcal{S}, \mathcal{M}\rangle, D\rangle$ and a D-dataset λ^+ of arity n, there exists a proper Σ-characterization of λ^+ in UCQ if and only if it is the case that $(\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)},\vec{c}) \not\to (\mathcal{C}_\mathcal{O}^{\mathcal{M}(D)},\vec{b})$ for all $\vec{c} \in \lambda^+$ and all $\vec{b} \in dom(D)^n \setminus \lambda^+$.

7. Existence

We now address the existence problem. First of all, for the scenario under consideration in this paper, the existence problem for both UCQ-minimally complete and UCQ-maximally sound separations (and also characterizations) is trivial, since by Theorems 4 and 5 they always exist. Thus, in this section we only consider the remaining proper case, by defining a variant of the decision problems as defined in [21], where also a mapping in some mapping language $\mathcal{L}_\mathcal{M}$ is given as input.

SEP($\mathcal{L}_\mathcal{O}$, $\mathcal{L}_\mathcal{M}$, $\mathcal{Q}$)

Input: An OBDM system $\Sigma = \langle\langle\mathcal{O}, \mathcal{S}, \mathcal{M}\rangle, D\rangle$ and two D-datasets λ^+ and λ^-, where $\mathcal{O} \in \mathcal{L}_\mathcal{O}$ and $\mathcal{M} \in \mathcal{L}_\mathcal{M}$.
Question: Does there exist a proper Σ-separation of λ^+ and λ^- in $\mathcal{Q}$?

CHAR($\mathcal{L}_\mathcal{O}$, $\mathcal{L}_\mathcal{M}$, $\mathcal{Q}$)

Input: An OBDM system $\Sigma = \langle\langle\mathcal{O}, \mathcal{S}, \mathcal{M}\rangle, D\rangle$ and a D-dataset λ^+, where $\mathcal{O} \in \mathcal{L}_\mathcal{O}$ and $\mathcal{M} \in \mathcal{L}_\mathcal{M}$.
Question: Does there exist a proper Σ-characterization of λ^+ in $\mathcal{Q}$?

We also introduce two important special cases of the above decision problems, namely: STSEP($\mathcal{L}_\mathcal{O}$, $\mathcal{L}_\mathcal{M}$, $\mathcal{Q}$) and STCHAR($\mathcal{L}_\mathcal{O}$, $\mathcal{L}_\mathcal{M}$, $\mathcal{Q}$), which are special cases of SEP($\mathcal{L}_\mathcal{O}$, $\mathcal{L}_\mathcal{M}$, $\mathcal{Q}$) and CHAR($\mathcal{L}_\mathcal{O}$, $\mathcal{L}_\mathcal{M}$, $\mathcal{Q}$), respectively, in which the all the input D-datasets are singleton sets (i.e., they consist of just a single tuple).

In what follows, we show that the computational complexity of the above decision problems differ only depending on the mapping language $\mathcal{L}_\mathcal{M}$ adopted. A key difference between GLAV and the special cases GAV and LAV is in the size of $\mathcal{M}(D)$. In GLAV mappings $\mathcal{M}$, the set of atoms $\mathcal{M}(D)$ can be exponentially large due to the simultaneous presence in mapping assertions of queries in the left-hand side of the assertions involving multiple source predicates, and join existential variables in the right-hand side of the assertions. As an example, take $D = \{s_i(0), s_i(1) \mid 1 \leqslant i \leqslant n\}$ and $\mathcal{M}$ containing the GLAV assertion: $\{(x_1, \ldots, x_n) \mid s_1(x_1) \wedge \ldots \wedge s_n(x_n)\} \rightarrow \{(x_1, \ldots, x_n) \mid \exists y. P(x_1, y) \wedge \ldots \wedge P(x_n, y)\}$. One can see that the number of atoms occurring in the set $\mathcal{M}(D)$ is 2^n. Conversely, in both LAV and GAV mappings, $\mathcal{M}(D)$ is always polynomially bounded since the former does not allow for multiple source predicates in the left-hand side of mapping assertions, whereas the latter does not allow for existential variables in the right-hand side of mapping assertions and the arity of ontology predicates is fixed to at most 2.

GAV and LAV mappings, however, differ for the effort in computing $\mathcal{M}(D)$. While in LAV mappings $\mathcal{M}(D)$ can be always computed in polynomial time, in GAV mappings there are CQs on the left-hand side of mapping assertions that need to be evaluated, and so $\mathcal{M}(D)$ cannot be computed in polynomial time (unless PTIME = NP).

We start by characterizing the computational complexity of SEP and CHAR (and their respective subproblems) in the simplest LAV case, then we address the GAV case, and finally we focus on the most general GLAV case. Interestingly, all the provided matching lower bounds hold even for fixed ontologies $\mathcal{O} = \emptyset$, i.e., ontologies without assertions, and fixed D-datasets containing only a single unary tuple.

Importantly, for the scenario under consideration, from the results of the previous section, the questions in SEP and CHAR can be reformulated equivalently as follows: "is $q_\mathcal{O} = \mathit{query}(\mathcal{M}(D), \vec{c_1}) \cup \ldots \cup \mathit{query}(\mathcal{M}(D), \vec{c_m})$ also a sound (and so, a proper) Σ-separation of $\lambda^+ = \{\vec{c_1}, \ldots, \vec{c_m}\}$ and λ^- in UCQ?" and "is $q_\mathcal{O} = \mathit{query}(\mathcal{M}(D), \vec{c_1}) \cup \ldots \cup \mathit{query}(\mathcal{M}(D), \vec{c_m})$ also a sound (and so, a proper) Σ-characterization of $\lambda^+ = \{\vec{c_1}, \ldots, \vec{c_m}\}$ in UCQ?", respectively.

Theorem 6. *SEP(DL-Lite$_\mathcal{R}$, LAV, UCQ) and CHAR(DL-Lite$_\mathcal{R}$, LAV, UCQ) are coNP-complete. The hardnesses already hold for STSEP($\emptyset$, GAV$\cap$LAV, UCQ) and STCHAR($\emptyset$, GAV$\cap$LAV, UCQ).*

Proof. **Upper bound:** We only mention SEP(*DL-Lite*$_\mathcal{R}$, LAV, UCQ). The CHAR(*DL-Lite*$_\mathcal{R}$, LAV, UCQ) case is similar and therefore not discussed. Given an OBDM system $\Sigma = \langle J, D\rangle$ with $J = \langle\mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ in which $\mathcal{O}$ is a *DL-Lite*$_\mathcal{R}$ ontology and $\mathcal{M}$ is a LAV mapping, and two D-datasets $\lambda^+ = \{\vec{c_1}, \ldots, \vec{c_m}\}$ and λ^-, we first compute $\mathcal{M}(D)$ in polynomial time (recall that $\mathcal{M}$ is a LAV mapping), and therefore also the query $q_\mathcal{O} = \mathit{query}(\mathcal{M}(D), \vec{c_1}) \cup \ldots \cup \mathit{query}(\mathcal{M}(D), \vec{c_m})$ which is the UCQ-minimally complete Σ-separation of λ^+ and λ^-. Then, exactly as illustrated in Theorem 2, we can check in coNP whether $\mathit{query}(\mathcal{M}(D), \vec{c_1}) \cup \ldots \cup \mathit{query}(\mathcal{M}(D), \vec{c_m})$ is also a sound (and so, a proper) Σ-separation of λ^+ and λ^- in UCQ.

Lower bound: We start with STSEP($\emptyset$, GAV$\cap$LAV, UCQ), and then we address the STCHAR($\emptyset$, GAV$\cap$LAV, UCQ) case. The proof of coNP-hardness is by a LOGSPACE reduction from the complement of the *3-colourability problem*. We define a fixed OBDM specification $J = \langle\mathcal{O}, \mathcal{S}, \mathcal{M}\rangle$ in which $\mathcal{O}$ contains no assertions and whose alphabet consists of an atomic role e and an atomic concept A, $\mathcal{S} = \{s_e, s\}$, and $\mathcal{M}$ consists of the following two

GAV∩LAV mapping assertions: $\{(x_1, x_2) \mid s_e(x_1, x_2)\} \rightarrow \{(x_1, x_2) \mid e(x_1, x_2)\}$ and $\{(x) \mid s(x)\} \rightarrow A(x)\}$, which simply mirrors source predicates s_e and s to e and A, respectively.

Let $G = (V, E)$ be a finite and undirected graph without loops or isolated nodes, where $V = \{c_1, \ldots, c_n\}$. Without loss of generality, we may assume that $V \neq \emptyset$ and that G is *connected*, i.e., there is a path from c to c' for any pair of nodes $(c, c') \in V^2$. Then, we define an $\mathcal{S}$-database D_G composed of the following facts:

$$\{s_e(c, c') \mid (c, c') \in E\} \cup \{s(c_1)\} \cup$$
$$\{s_e(x, y) \mid x = \{r, g, b\} \text{ and } y = \{r, g, b\} \text{ and } x \neq y\} \cup \{s(r)\}.$$

Let, moreover, λ^+ and λ^- be the fixed D_G-datasets $\lambda^+ = \{(c_1)\}$ and $\lambda^- = \{(r)\}$.

Notice that the $\mathcal{S}$-database D_G can be constructed in LOGSPACE from an input graph G as above. Let the OBDM system be $\Sigma_G = \langle J, D_G \rangle$. We now show that a graph G is not 3-colourable if and only if there exists a proper Σ_G-separation of λ^+ and λ^- in UCQ, thus proving the claimed lower bound.

Claim 2. *Given a graph G, there exists a proper Σ_G-separation of λ^+ and λ^- in UCQ if and only if G is not 3-colourable.*

Proof. First of all, note that $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D_G)} = \{e(c, c') \mid (c, c') \in E\} \cup \{A(c_1)\} \cup \{e(x, y) \mid x = \{r, g, b\} \text{ and } y = \{r, g, b\} \text{ and } x \neq y\} \cup \{A(r)\}$. We recall that a proper Σ_G-separation of λ^+ and λ^- in UCQ exists if and only if the CQ $q_G = \mathit{query}(\mathcal{M}(D_G), (c_1))$ is also a sound (and so, a proper) Σ_G-separation of λ^+ and λ^- in UCQ.

"**Only-if part:**" Suppose $G = (V, E)$ is 3-colourable, that is, there exists a function $f : V \rightarrow \{r, g, b\}$ such that $f(c) \neq f(c')$ for each $(c, c') \in E$. Without loss of generality, we may assume that $f(c_1) = r$ (indeed, the existence of f clearly implies the existence of a function f' with $f'(c_1) = r$ and such that $f'(c) \neq f'(c')$ for each $(c, c') \in E$ holds as well). Consider the extension h of f assigning $h(x) = x$ to each $x \in \{r, g, b\}$. It can be readily seen that h consists in a homomorphism witnessing that $(\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D_G)}, (c_1)) \rightarrow (\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D_G)}, (r))$, which directly implies that $(r) \in \mathit{cert}_{q_G, J}^{D_G}$. It follows that q_G is not a sound Σ_G-separation of λ^+ and λ^- in UCQ, and therefore no proper Σ_G-separation of λ^+ and λ^- in UCQ exists.

"**If part:**" Suppose there exists no proper Σ_G-separation of λ^+ and λ^- in UCQ, i.e., the CQ q_G is such that $(r) \in \mathit{cert}_{q_G, J}^{D_G}$. It follows that there exists a homomorphism h witnessing that $(\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D_G)}, (c_1)) \rightarrow (\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D_G)}, (r))$. Now, since the graph G is connected and since $h(c_1) = h(r)$, by construction of $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D_G)}$ the homomorphism h must necessarily be such that $h(c) \in \{r, g, b\}$ for each constant c representing a node $c \in V$. But then, due to the fact that none of $e(r, r)$, $e(g, g)$, and $e(b, b)$ occur in $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D_G)}$, we derive that h is such that $h(c) \neq h(c')$ for each $e(c, c') \in \mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D_G)}$, and therefore for each $(c, c') \in E$ as well. Thus, we can conclude that G is 3-colourable. □

As for the coNP-hardness of STCHAR($\emptyset$, GAV∩LAV, UCQ), it is possible to use exactly the same reduction provided above by discarding λ^- and considering only $\lambda^+ = \{(c_1)\}$. In particular, due to the fact that $A(c)$ and $A(r)$ are the only A-facts occurring in $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D_G)}$, the set of certain answers of the query q_G w.r.t. Σ_G is always either $\{(c_1)\}$ or $\{(c_1), (r)\}$, and which one of the two depends on the 3-colourability of G. □

We now turn to consider GAV mappings. We recall that the complexity class Θ_2^p has many characterizations: $\Theta_2^p = \mathrm{P}^{\mathrm{NP}[\mathrm{O}(\log n)]} = \mathrm{P}$ *with a constant number of rounds of parallel queries to an oracle for a decision problem in* NP [55] (we refer the reader to [56] for further characterizations of such complexity class). By a round of parallel queries, it is intended that the Turing machine can ask for polynomially many *non-adaptive queries* to the NP oracle.

Theorem 7. SEP*(DL-Lite$_{\mathcal{R}}$, GAV, UCQ) and* CHAR*(DL-Lite$_{\mathcal{R}}$, GAV, UCQ) are Θ_2^p-complete. The hardnesses already hold for ST*SEP*($\emptyset$, GAV, UCQ) and ST*CHAR*($\emptyset$, GAV, UCQ).*

Proof. **Upper bound:** We only mention SEP(*DL-Lite$_{\mathcal{R}}$*, GAV, UCQ). The CHAR(*DL-Lite$_{\mathcal{R}}$*, GAV, UCQ) case is similar and therefore not discussed. Given an OBDM system $\Sigma = \langle J, D \rangle$ with $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ in which $\mathcal{O}$ is a *DL-Lite$_{\mathcal{R}}$* ontology and $\mathcal{M}$ is a GAV mapping, and two D-datasets $\lambda^+ = \{\vec{c_1}, \ldots, \vec{c_m}\}$ and λ^-, as a first step we compute $\mathcal{M}(D)$ in polynomial time with a single round of parallel queries to an NP oracle. More specifically, for

each pair of constants $(c_1, c_2) \in dom(D)^2$ (resp., constant $c \in dom(D)$) and for each atomic role P (resp., concept A) in the alphabet of $\mathcal{O}$ we ask, all together with a single round of parallel queries to an NP oracle, whether $P(c_1, c_2) \in \mathcal{M}(D)$ (resp., $A(c) \in \mathcal{M}(D)$). It is clear that deciding whether $P(c_1, c_2) \in \mathcal{M}(D)$ (resp., $A(c) \in \mathcal{M}(D)$) for a given pair of constants $(c_1, c_2) \in dom(D)^2$ (resp., constant $c \in dom(D)$), a GAV mapping $\mathcal{M}$, and a database D is an NP-complete problem because the left-hand side of mapping assertions are CQs [33].

Once obtained $\mathcal{M}(D)$ as described above, we construct in polynomial time the UCQ $q_{\mathcal{O}} = query(\mathcal{M}(D), \vec{c_1}) \cup \ldots \cup query(\mathcal{M}(D), \vec{c_m})$. Then, due to Theorem 2, with a second and final round of parallel queries, we can ask with a single query to an NP oracle whether $q_{\mathcal{O}}$ is also a sound (and so, a proper) Σ-separation of $\lambda^+ = \{\vec{c_1}, \ldots, \vec{c_m}\}$ and λ^- in UCQ.

Lower bound: We start with STSEP($\emptyset$, GAV, UCQ), and then we address the STCHAR($\emptyset$, GAV, UCQ) case. The proof of Θ_2^p-hardness is by a LOGSPACE reduction from the *odd clique problem*, which is a Θ_2^p-complete problem [57]. *Odd clique* is the problem of deciding, given a finite and undirected graph $G = (V, E)$ without loops, whether the maximum clique size of G is an odd number. Without loss of generality, we may assume that E contains at least an edge and that the cardinality of V is an even number (indeed, it is always possible to add fresh isolated nodes to the graph G without changing its maximum clique size).

Given a graph $G = (V, E)$ as above with $V = \{v_1, \ldots, v_n\}$, we define an OBDM system $\Sigma_G = \langle J_G, D_G \rangle$ as follows: $J_G = \langle \mathcal{O}, \mathcal{S}_G, \mathcal{M}_G \rangle$ is an OBDM specification in which $\mathcal{O}$ contains no assertions, $\mathcal{S}_G = \{e, s_1, \ldots, s_n\}$, and, for each odd $i \in [1, n]$, the mapping $\mathcal{M}_G$ has the following two GAV assertions:

$$\{(x) \mid \exists y_1, \ldots, y_i . s_i(x) \wedge cl_i\} \rightarrow \{(x) \mid A_i(x)\}$$
$$\{(x) \mid \exists y_1, \ldots, y_{i+1} . s_{i+1}(x) \wedge cl_{i+1}\} \rightarrow \{(x) \mid A_i(x)\}$$

where A_i is an atomic concept in the alphabet of $\mathcal{O}$ and, for each natural number p, cl_p is the conjunction of atoms:

$$cl_p = \bigwedge_{\{(k,j) \mid 1 \leqslant k < j \leqslant p\}} e(y_k, y_j)$$

.

Intuitively, cl_p asks whether G contains a clique of size p. Finally, D_G is the $\mathcal{S}_G$-database $D_G = \{e(x_1, x_2) \mid (x_1, x_2) \in E\} \cup \{e(x_2, x_1) \mid (x_1, x_2) \in E\} \cup \{s_i(c) \mid 1 \leqslant i \leqslant n \text{ and } i \text{ is odd}\} \cup \{s_i(c') \mid 2 \leqslant i \leqslant n \text{ and } i \text{ is even}\}$. Let, moreover, λ^+ and λ^- be the fixed D_G-datasets $\lambda^+ = \{(c)\}$ and $\lambda^- = \{(c')\}$.

Notice that λ^+ and λ^- are fixed, whereas the OBDM system Σ_G can be constructed in LOGSPACE from an input graph G as above.

The correctness of the reduction is mainly based on the next property:

Claim 3. *Let $i \in [1, n]$ be an odd number. We have that:*

1. $A_i(c) \in \mathcal{C}_{\mathcal{O}}^{\mathcal{M}_G(D_G)}$ *if and only if G contains a clique of size i.*
2. $A_i(c') \in \mathcal{C}_{\mathcal{O}}^{\mathcal{M}_G(D_G)}$ *if and only if G contains a clique of size $i+1$.*

Proof. As for 1, since $s_i(c) \in D_G$, it is easy to see that the query $q_i = \{(x) \mid \exists y_1, \ldots, y_i . s_i(x) \wedge cl_i\}$ is such that $(c) \in q_i^{D_G}$ if and only if G has a clique of size i. Thus, due to the GAV assertion $q_i \rightarrow \{(x) \mid A_i(x)\}$ occurring in $\mathcal{M}_G$, we have $A_i(c) \in \mathcal{C}_{\mathcal{O}}^{\mathcal{M}_G(D_G)}$ if and only if G has a clique of size i.

As for 2, since $s_{i+1}(c') \in D_G$, it is easy to see that the query $q_{i+1} = \{(x) \mid \exists y_1, \ldots, y_{i+1} . s_{i+1}(x) \wedge cl_{i+1}\}$ is such that $(c') \in q_{i+1}^{D_G}$ if and only if G has a clique of size $i+1$. Thus, due to the GAV assertion $q_{i+1} \rightarrow \{(x) \mid A_i(x)\}$ occurring in $\mathcal{M}_G$, if G has a clique of size $i+1$, then $A_i(c') \in \mathcal{C}_{\mathcal{O}}^{\mathcal{M}_G(D_G)}$. Conversely, suppose that G has not a clique of size $i+1$. On the one hand, the assertion $q_{i+1} \rightarrow \{(x) \mid A_i(x)\}$ does not make $A_i(c')$ true in $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}_G(D_G)}$. On the other hand, since $s_i(c') \notin D_G$, not even the assertion $\{(x) \mid \exists y_1, \ldots, y_i . s_i(x) \wedge cl_i\} \rightarrow \{(x) \mid A_i(x)\}$ makes $A_i(c')$ true in $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}_G(D_G)}$. Thus, $A_i(c') \notin \mathcal{C}_{\mathcal{O}}^{\mathcal{M}_G(D_G)}$. □

With the above property in hand, we can now prove that the maximum clique size of a graph G is an odd number if and only if the CQ $q_{\mathcal{O}} = query(\mathcal{M}_G(D_G), (c))$ is also a sound (and so, a proper) Σ_G-separation of λ^+ and λ^- in UCQ, thus showing the claimed lower bound.

Claim 4. *The maximum clique size of a graph G is an odd number if and only if the CQ $q_{\mathcal{O}} = query(\mathcal{M}_G(D_G), (c))$ is also a sound (and so, a proper) Σ_G-separation of λ^+ and λ^- in UCQ.*

Proof. "**Only-if part:**" Suppose that the maximum clique size of G is p, where p is an odd number. Due to Claim 3, we have that $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}_G(D_G)} = \{A_1(c), A_1(c'), A_3(c), A_3(c'), \ldots, A_p(c)\}$. Observe that $A_p(c') \notin \mathcal{C}_{\mathcal{O}}^{\mathcal{M}_G(D_G)}$ because G has not a clique of size $p+1$ by assumption. Thus, $q_{\mathcal{O}} = query(\mathcal{M}_G(D_G), (c)) = \{(x_c) \mid \exists y_{c'}.\phi_{\mathcal{O}}(x_c, y_{c'})\}$, where $\phi_{\mathcal{O}}(x_c, y_{c'}) = A_1(x_c) \wedge A_1(y_{c'}) \wedge A_3(x_c) \wedge A_3(y'_c) \wedge \ldots \wedge A_p(x_c)$. It is straightforward to verify that $(set(\phi_{\mathcal{O}}), (x_c)) \rightarrow (\mathcal{C}_{\mathcal{O}}^{\mathcal{M}_G(D_G)}, (c))$ but $(set(\phi_{\mathcal{O}}), (x_c)) \not\rightarrow (\mathcal{C}_{\mathcal{O}}^{\mathcal{M}_G(D_G)}, (c'))$. It follows that $cert_{q_{\mathcal{O}}, J_G}^{D_G} = \{(c)\}$, i.e., $q_{\mathcal{O}}$ is a proper Σ_G-separation of λ^+ and λ^- in UCQ.

"**If part:**" Suppose that the maximum clique size of G is r, where r is an even number. Due to Claim 3, we have that $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}_G(D_G)} = \{A_1(c), A_1(c'), A_3(c), A_3(c'), \ldots, A_{r-1}(c), A_{r-1}(c')\}$. Observe that $A_{r-1}(c') \in \mathcal{C}_{\mathcal{O}}^{\mathcal{M}_G(D_G)}$ and $A_{r+1}(c) \notin \mathcal{C}_{\mathcal{O}}^{\mathcal{M}_G(D_G)}$ because by assumption G has a clique of size r but not of size $r+1$. Thus, $q_{\mathcal{O}} = query(\mathcal{M}_G(D_G), (c)) = \{(x_c) \mid \exists y_{c'}.\phi_{\mathcal{O}}(x_c, y_{c'})\}$, where $\phi_{\mathcal{O}}(x_c, y_{c'}) = A_1(x_c) \wedge A_1(y_{c'}) \wedge A_3(x_c) \wedge A_3(y'_c) \wedge \ldots \wedge A_{r-1}(x_c) \wedge A_{r-1}(y_{c'})$. It is straightforward to verify that $(set(\phi_{\mathcal{O}}), (x_c)) \rightarrow (\mathcal{C}_{\mathcal{O}}^{\mathcal{M}_G(D_G)}, (c'))$. It follows that $(c') \in cert_{q_{\mathcal{O}}, J_G}^{D_G}$, and therefore $q_{\mathcal{O}}$ is not a sound (and so, not a proper) Σ_G-separation of λ^+ and λ^- in UCQ. □

As for the Θ_2^p-hardness of STCHAR($\emptyset$, GAV, UCQ), it is possible to use exactly the same reduction provided above by discarding λ^- and considering only $\lambda^+ = \{(c)\}$. In particular, the set of certain answers of the query $q_{\mathcal{O}} = query(\mathcal{M}_G(D_G), (c))$ w.r.t. Σ_G is always either $\{(c)\}$ or $\{(c), (c')\}$, and which one of the two depends on the parity of the maximum clique size of G. □

We conclude this section by addressing the remaining more general GLAV mappings.

Theorem 8. SEP*(DL-Lite$_{\mathcal{R}}$, GLAV, UCQ) and* CHAR*(DL-Lite$_{\mathcal{R}}$, GLAV, UCQ) are* coNEXPTIME*-complete. The hardnesses already hold for ST*SEP*($\emptyset$, GLAV, UCQ) and ST*CHAR*($\emptyset$, GLAV, UCQ).*

Proof. **Upper bound:** We only mention SEP(*DL-Lite*$_{\mathcal{R}}$, GLAV, UCQ). The CHAR(*DL-Lite*$_{\mathcal{R}}$, GLAV, UCQ) case is similar and therefore not discussed. In particular, we show how to check in NEXPTIME whether $q_{\mathcal{O}} = query(\mathcal{M}(D), \vec{c_1}) \cup \ldots \cup query(\mathcal{M}(D), \vec{c_m})$ is not a sound (and so, not a proper) Σ-separation of $\lambda^+ = \{\vec{c_1}, \ldots, \vec{c_m}\}$ and λ^- in UCQ.

As a first step, we compute $q_{\mathcal{O}} = query(\mathcal{M}(D), \vec{c_1}) \cup \ldots \cup query(\mathcal{M}(D), \vec{c_m})$ in exponential time (note that $\mathcal{M}(D)$ can be exponentially large, and so also the UCQ $q_{\mathcal{O}}$). Then, we can proceed similarly as in the proof of Theorem 2. We guess (*i*) a tuple of constants $\vec{c}$, and (*ii*) $q'_{\mathcal{O}}$, $\rho_{\mathcal{O}}$, $q_{\mathcal{S}}$, $\rho_{\mathcal{M}}$, and f, which now can be objects of exponential size. Finally, we first check whether (*i*) $\vec{c} \in \lambda^-$, and then we employ the same procedure used in the proof of Theorem 1 to check whether (*ii*) $\vec{c} \in cert_{q_{\mathcal{O}}, J}^{D}$, which, as already shown can be carried out in polynomial time in the size of the guessed objects (and therefore in exponential time with respect to the size of the input).

Lower bound: We start with STSEP($\emptyset$, GLAV, UCQ), and then we address the STCHAR($\emptyset$, GLAV, UCQ) case. Moreover, to simplify the readability of the proof, we first illustrate the main idea behind it. In particular, as a first step we provide an alternative proof of coNP-hardness of UCQ-explainability in the plain relational database case. The (alternative) coNP-hardness proof is by a LOGSPACE reduction from the *complement of the clique problem*. We employ a schema $\mathcal{S} = \{e, P\}$. Let $\langle G, k\rangle$ be an instance of the *clique problem*, where $G = (V, E)$ is a finite and undirected graph without loops and with $E \neq \emptyset$, and k is a natural number *written in unary* (note indeed that *clique* is *strongly NP-complete* [58]). Starting from G, we define an $\mathcal{S}$-database $D = D_k \cup D_G$ as follows:

$$D_k = \{e(y_i, y_j) \mid 1 \leqslant i < j \leqslant k\} \cup \{P(c, y_i) \mid 1 \leqslant i \leqslant k\}$$

$$D_G = \{e(v_i, v_j) \mid (v_i, v_j) \in E\} \cup \{e(v_j, v_i) \mid (v_i, v_j) \in E\} \cup \{P(c', v) \mid v \in V\}$$

Finally, λ^+ and λ^- are the fixed D-datasets $\lambda^+ = \{(c)\}$ and $\lambda^- = \{(c')\}$. Intuitively, D_k represents a clique of size k, whereas D_G represents the graph G.

Notice that D can be constructed in LOGSPACE from G. Let $q_k = \mathit{query}(D_k, (c))$, which is equivalent to:

$$\{(x) \mid \exists y_1, \ldots, y_k . P(x, y_1) \wedge \ldots \wedge P(x, y_k) \wedge \bigwedge_{1 \leqslant i < j \leqslant k} (e(y_i, y_j))\}.$$

One can easily verify that λ^+ and λ^- are UCQ-explainable inside D if and only if $q_k^D = \{(c)\}$. Indeed, q_k is similar to the canonical UCQ $\mathit{query}(D, (c))$ [11], where, however, only constants *reachable* from c are taken into account in the query. More specifically, by construction, the evaluation of q_k over D is either $\{(c)\}$, or $\{(c), (c')\}$. Indeed, q_k^D always contain the tuple (c), and it contains the tuple (c') if and only if $(D_k, (c)) \to (D_G, (c'))$ (obviously, $(D_k, (c)) \to (D_G, (c'))$ if and only if $(\mathit{set}(\phi), (x)) \to (D, (c'))$, where ϕ is the body of q_k). In the next property, we are going to show that this latter is the case if and only if G has a clique of size k, thus proving the claimed coNP-hardness.

Claim 5. *A graph G has a clique of size k if and only if $(D_k, (c)) \to (D_G, (c'))$.*

Proof. "**Only-if part:**" Suppose G has a clique of size k, i.e., there are k nodes in G forming a clique. This immediately implies the existence of a function h from $\mathit{dom}(D_k)$ to $\mathit{dom}(D_G)$ mapping (*i*) c to c', and (*ii*) constant y_i to a constant v_i representing a node in such a clique, for each possible $i \in [1, k]$. Thus, h is a homomorphism witnessing that $(D_k, (c)) \to (D_G, (c'))$, as required.

"**If part:**" Suppose $(D_k, (c)) \to (D_G, (c'))$, and let h be the homomorphism witnessing this. Note that $h(c) = c'$. This immediately implies that the set of facts $\{e(h(y_i), h(y_j)) \mid e(y_i, y_j) \in D_k\}$, which must occur in D_G due to the assumption that h witnesses $(D_k, (c)) \to (D_G, (c'))$, denotes a clique of size k inside the graph G. □

By extending the above reduction, we are now ready to prove that SEP($\emptyset$, GLAV, UCQ) is coNEXPTIME-hard. The proof of coNEXPTIME-hardness is by a polynomial time reduction from the *complement of the succinct clique problem*. Given a succinct representation of a graph C_G representing a finite and undirected graph $G = (V, E)$ without loops and with $E \neq \emptyset$, and given a natural number k written in unary, *succinct clique* is the problem of deciding whether the graph represented by C_G has a clique of size k. *Succinct clique* is known to be NEXPTIME-complete [59].

For a succinct representation C_G of a graph $G = (V, E)$ with m nodes, without loss of generality, we implicitly mean a circuit using $2 \cdot b$ input gates $\vec{x} = (x_1, \ldots, x_b, x_{b+1}, \ldots, x_{2 \cdot b})$ (where $2^b = m$) for which on input $(a_1, \ldots, a_b, a_{b+1}, \ldots, a_{2 \cdot b})$ circuit C_G outputs true if and only if the two nodes v_i, v_j in V represented by $v_i = (a_1, \ldots, a_b)$ and $v_j = (a_{b+1}, \ldots, a_{2 \cdot b})$ are such that $(v_i, v_j) \in E$ (see [60] for further details). Moreover, from a circuit C_G with $\vec{x}$ as input gates, we denote by $F_{C_G}(\vec{x}, \vec{w})$ the 3-CNF formula obtained by applying the *Tseitin transformation* [61], where $\vec{w}$ are the fresh variables introduced by the transformation. We recall that such transformation is linear in the size of C_G. And, among the introduced variables $\vec{w}$ introduced by the linear transformation, there is one variable in $\vec{w}$, denoted by w, which represents the output gate of the circuit. More formally, the transformation is such that if on input $\vec{a} = (a_1, \ldots, a_b, a_{b+1}, \ldots, a_{2 \cdot b})$ circuit C_G outputs true, there there is exactly one satisfying assignment of formula $F_{C_G}(\vec{a}, \vec{w})$ with 1 as truth assignment to variable w, otherwise (i.e., C_G outputs false on $\vec{a}$) there is no satisfying assignment of $F_{C_G}(\vec{a}, \vec{w})$ with 1 as truth assignment to variable w [61].

Let $\langle C_G, k \rangle$ be an instance of the succinct clique problem, where C_G is a circuit with $2 \cdot b$ input gates succinctly representing a graph $G = (V, E)$ of $m = 2^b$ nodes, and k is a natural number written in unary. Let $F_{C_G}(\vec{x}, \vec{w}) = p_1 \wedge \ldots \wedge p_r$ be the 3-CNF formula associated to C_G, where each clause p_i is a disjunction of three literals, each literal being either a variable in $\vec{x} \cup \vec{w}$ or its negated. For $i \in [1, p]$, we denote by $o_{i_1}, o_{i_2}, o_{i_3}$ the first, the second, and the third, respectively, variable appearing (either positive or negated) in clause p_i.

Starting from C_G, we define an OBDM system $\Sigma = \langle J, D \rangle$ and two D-datasets λ^+ and λ^- as follows, where $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ is an OBDM specification and D is an $\mathcal{S}$-database. First, $\mathcal{O} = \emptyset$ is an empty set of assertions containing in its alphabet binary predicates e, P, and V_i for each $i \in [1, b]$. Intuitively, as in the above reduction from normal clique, e denotes the edges of graphs and P connects constants c and c' to nodes of the k-clique graph and the input graph G, respectively. The additional predicates V_i's are used to encode nodes of the graphs. Second,

$\mathcal{S} = \mathcal{S}_k \cup \mathcal{S}_G$, where $\mathcal{S}_k = \{s_e, s_p, s_v\}$ and $\mathcal{S}_G = \{s', s_w, p_1, \ldots, p_r\}$ with s' and s_w unary predicates, s_e and s_p binary predicates, p_1, p_2, …, and p_r ternary predicates, and s_v a $b+1$ predicate. Third, the $\mathcal{S}$-database $D = D_k \cup D_G$ is as follows. For each $i \in [1,k]$, D_k contains (*i*) $k-1$ constants $y_i^1, \ldots, y_i^{k-1}$ used to represent the clique without ever repeating the same constant, and (*ii*) b constants $d_i^1, \ldots, d_i^b$ used to encode node y_i and ensuring that the distinct constants $y_i^1, \ldots, y_i^{k-1}$ actually denote the same node y_i. That is:

$$D_k = \{s_e(y_i^j, y_l^i) \mid 1 \leqslant i \leqslant j < k \text{ and } l = j+1\} \cup \{s_p(c, y_i^l) \mid 1 \leqslant i \leqslant k \text{ and } 1 \leqslant l < k\} \cup$$
$$\{s_v(y_i^l, d_i^1, \ldots, d_i^b) \mid 1 \leqslant i \leqslant k \text{ and } 1 \leqslant l < k\}$$

As an explanatory example, if we have $k = 4$ and $b = 2$, then we have $D_k = \{s_e(y_1^1, y_2^1), s_e(y_1^2, y_3^1), s_e(y_1^3, y_4^1), s_e(y_2^2, y_3^2), s_e(y_2^3, y_4^2), s_e(y_3^3, y_4^3)\} \cup \{s_p(c, y_i^1) \mid 1 \leqslant i \leqslant 4\} \cup \{s_p(c, y_i^2) \mid 1 \leqslant i \leqslant 4\} \cup \{s_p(c, y_i^3) \mid 1 \leqslant i \leqslant 4\} \cup \{s_v(y_i^1, d_i^1, d_i^2) \mid 1 \leqslant i \leqslant 4\} \cup \{s_v(y_i^2, d_i^1, d_i^2) \mid 1 \leqslant i \leqslant 4\} \cup \{s_v(y_i^3, d_i^1, d_i^2) \mid 1 \leqslant i \leqslant 4\}$. Informally, e contains a clique of size $k = 4$ without ever repeating a node, while relation s_v ensures that, for each $i \in [1,4]$, the distinct constants y_i^1, y_i^2, y_i^3 actually denote the same node y_i because they are connected with same elements d_i^1, d_i^2 in s_v.

As for D_G, we have $s'(c') \in D_G$ and $s_w(1) \in D_G$. Observe that for each clause p_i there are exactly seven satisfying truth assignments of the clause, each of the form $(t_{j_1}^i, t_{j_2}^i, t_{j_3}^i)$, where $j \in [1,7]$ and each $t_{j_l}^i$ is the truth assignment (i.e., either constant 0 or constant 1) given to the variable o_{i_l} by j. The predicate p_i in D_G associated with the clause simply lists such satisfying assignments:

$$D_G = \{p_i(t_{j_1}^i, t_{j_2}^i, t_{j_3}^i) \mid 1 \leqslant i \leqslant r \text{ and } 1 \leqslant j \leqslant 7\} \cup \{s'(c')\} \cup \{s_w(1)\}$$

For example, if the circuit C_G succinctly representing G corresponds to formula x_1 XOR x_2 (note $b = 1$), then the 3-CNF formula is $F_{C_G} = \exists x_1, x_2, w.(\neg x_1 \vee \neg x_2 \vee \neg w) \wedge (x_1 \vee x_2 \vee \neg w) \wedge (x_1 \vee \neg x_2 \vee w) \wedge (\neg x_1 \vee x_2 \vee w)$ (with w the output gate being the only fresh variable introduced by the Tseitin transformation) and D_G is as follows:

$$D_G = \{s'(c'), s_w(1),$$
$$p_1(0,0,0), p_1(0,0,1), p_1(0,1,0), p_1(0,1,1), p_1(1,0,0), p_1(1,0,1), p_1(1,1,0),$$
$$p_2(0,0,0), p_2(0,1,0), p_2(0,1,1), p_2(1,0,0), p_2(1,0,1), p_2(1,1,0), p_2(1,1,1),$$
$$p_3(0,0,0), p_3(0,0,1), p_3(0,1,1), p_3(1,0,0), p_3(1,0,1), p_3(1,1,0), p_3(1,1,1)$$
$$p_4(0,0,0), p_4(0,0,1), p_4(0,1,0), p_4(0,1,1), p_4(1,0,1), p_4(1,1,0), p_4(1,1,1)\}$$

The fixed D-datasets λ^+ and λ^- are $\lambda^+ = \{(c)\}$ and $\lambda^- = \{(c')\}$. It remains to describe the GLAV mapping $\mathcal{M}$ in the OBDM system Σ. We have $\mathcal{M} = \mathcal{M}_k \cup \mathcal{M}_G$, where $\mathcal{M}_k = \{m_k^1, m_k^2, m_k^3\}$ and $\mathcal{M}_G = \{m_G\}$ such that $\mathcal{M}_k$ is simply as follows:

$$m_k^1 : \{(x', x'') \mid s_e(x', x'')\} \rightarrow \{(x', x'') \mid e(x', x')\}$$
$$m_k^2 : \{(x, x') \mid s_p(x, x')\} \rightarrow \{(x, x') \mid P(x, x')\}$$
$$m_k^3 : \{(x, x_1, \ldots, x_b) \mid s_v(x, x_1, \ldots, x_b)\} \rightarrow \{(x, x_1, \ldots, x_b) \mid V_1(x, x_1) \wedge \ldots \wedge V_b(x, x_b)\}$$

and $m_G \in \mathcal{M}_G$ is the following GLAV assertion:

$$\{(x, x_1, \ldots, x_{2 \cdot b}) \mid \exists \vec{w}.s'(x) \wedge p_1(o_{1,1}, o_{1,2}, o_{1,3}) \wedge \ldots \wedge p_r(o_{r,1}, o_{r,2}, o_{r,3}) \wedge s_w(w)\} \rightarrow$$
$$\{(x, x_1, \ldots, x_{2 \cdot b}) \mid \exists z_1, z_2.e(z_1, z_2) \wedge e(z_2, z_1) \wedge P(x, z_1) \wedge P(x, z_2) \wedge$$
$$V_1(z_1, x_1) \wedge \ldots \wedge V_b(z_1, x_b) \wedge V_1(z_2, x_{b+1}) \wedge \ldots \wedge V_b(z_2, x_{2 \cdot b})\}$$

For example, for the circuit C_G given before, the GLAV assertion m_G is:

$$\{(x, x_1, x_2) \mid \exists w_1.s'(x) \wedge p_1(x_1, x_2, w) \wedge p_2(x_1, x_2, w) \wedge p_3(x_1, x_2, w) \wedge p_4(x_1, x_2, w) \wedge s_w(w)\} \rightarrow$$
$$\{(x, x_1, x_2) \mid \exists z_1, z_2.e(z_1, z_2) \wedge e(z_2, z_1) \wedge P(x, z_1) \wedge P(x, z_2) \wedge V_1(z_1, x_1) \wedge V_1(z_2, x_2)\}$$

Observe that $\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)} = \mathcal{M}(D) = \mathcal{M}_k(D_k) \cup \mathcal{M}_G(D_G)$. Informally, the extension of predicate e in $\mathcal{M}_k(D_k)$ describes a clique of size k without ever repeating a node, while the extensions of predicates V_i's in $\mathcal{M}_k(D_k)$ ensure that, for each $i \in [1,k]$, the distinct constants $y_i^1, \ldots, y_i^{k-1}$ actually denote the same node y_i because they are connected with same elements $d_i^1, \ldots, d_i^b$. Finally, the extension of predicate P in $\mathcal{M}_k(D_k)$ simply contains (c, y_i^l) for each constant y_i^l occurring in the extension of e in $\mathcal{M}_k(D_k)$.

As for $\mathcal{M}_G(D_G)$, let $\vec{a} = (a_1, \ldots, a_b, a_{b+1}, a_{2\cdot b})$ be an input to the circuit C_G. By construction, one can easily verify that C_G outputs true if and only if $(c, a_1, \ldots, a_b, a_{b+1}, a_{2\cdot b})$ is a tuple in the evaluation of the left-hand side query of m_G over D_G. Thus, for each edge $((a_1, \ldots, a_b), (a_{b+1}, a_{2\cdot b})) \in E$ represented by circuit C_G, the chase $\mathcal{M}_G(D_G)$ produces two fresh variables z_1 and z_2 that simulates, respectively, $v_1 = (a_1, \ldots, a_b)$ and $v_2 = (a_{b+1}, \ldots, a_{2\cdot b})$ and connect them trough predicate e (simulating so the edge $(v_1, v_2) \in E$ described by circuit C_G). Moreover, both the freshly introduced variables z_1 and z_2 are connected through the extensions of predicates V_i's in the following way: $V_1(z_1, a_1), \ldots, V_b(z_1, a_b), V_1(z_2, a_{b+1}), \ldots, V_b(z_2, a_{2\cdot b})$. This ensures that two distinct variables z and z' that actually denote the same node in G, denote the same node also in $\mathcal{M}_G(D_G)$ because they are connected with same elements. Finally, the extension of predicate P in $\mathcal{M}_G(D_G)$ simply contains (c', z) for each freshly introduced variable z by the application of the chase.

Notice that $\lambda^+ = \{(c)\}$ and $\lambda^- = \{(c')\}$ are two fixed D-datasets and both the OBDM specification $J = \langle \mathcal{O}, \mathcal{S}, \mathcal{M} \rangle$ and the $\mathcal{S}$-database D can be always constructed in polynomial time from C_G. Let $q_k = \mathit{query}(\mathcal{M}_k(D_k), (c))$, which is equivalent to:

$$\{(x) \mid \exists y_1^1, \ldots, y_1^{k-1}, \ldots, y_k^1, \ldots, y_k^{k-1}, d_1^1, \ldots, d_1^b, \ldots, d_k^1, \ldots, d_k^b.$$
$$\bigwedge_{1 \leqslant i \leqslant j < k \text{ and } l = j+1} (e(y_i^j, y_l^i)) \wedge \bigwedge_{1 \leqslant i \leqslant k \text{ and } 1 \leqslant l < k} (P(x, y_i^l)) \wedge \bigwedge_{1 \leqslant i \leqslant k \text{ and } 1 \leqslant l < k} (V_1(y_i^l, d_i^1) \wedge \ldots \wedge V_b(y_i^l, d_i^b))\}$$

One can easily verify that a proper Σ-separation of λ^+ and λ^- in UCQ exists if and only if $\mathit{cert}_{q_k,J}^D = \{(c)\}$. Indeed, q_k is similar to $\mathit{query}(\mathcal{M}(D), (c))$ (i.e., the UCQ-minimally complete Σ-separation of λ^+ and λ^-), where, however, only elements *reachable* from c in $\mathcal{M}(D)$ are taken into account in the query. More specifically, by construction, the set of certain answers of q_k with respect to $\Sigma = \langle J, D \rangle$ is either $\{(c)\}$, or $\{(c), (c')\}$. Indeed, $\mathit{cert}_{q_k,J}^D$ always contain the tuple (c), and it contains the tuple (c') if and only if $(\mathcal{M}_k(D_k), (c)) \rightarrow (\mathcal{M}_G(D_G), (c'))$. Clearly, $(\mathcal{M}_k(D_k), (c)) \rightarrow (\mathcal{M}_G(D_G), (c'))$ if and only if $(\mathit{set}(\phi), (x)) \rightarrow (\mathcal{C}_{\mathcal{O}}^{\mathcal{M}(D)}, (c'))$, where ϕ is the body of q_k (recall that this latter condition is equivalent to $(c') \in \mathit{cert}_{q_k,J}^D$). In the next property, we are going to show that $(\mathcal{M}_k(D_k), (c)) \rightarrow (\mathcal{M}_G(D_G), (c'))$ if and only if the graph G represented by the circuit C_G has a clique of size k, thus proving the claimed coNEXPTIME-hardness and concluding the proof.

Claim 6. *A graph G represented by a circuit C_G has a clique of size k if and only if $(\mathcal{M}_k(D_k), (c)) \rightarrow (\mathcal{M}_G(D_G), (c'))$.*

Proof. "**Only-if part:**" Suppose the graph G represented by circuit C_G has a clique of size k. Let $(v_1, v_2, \ldots, v_k)$ be the nodes forming such clique in G, where each node v_i is encoded by $v_i = (a_{i,1}, \ldots, a_{i,b})$ in C_G (each $a_{i,x}$ is either 0 or 1). Consider any edge $(v_i, v_l) \in E$ (or $(v_l, v_i) \in E$) which by assumption exists for each pair (i, l) with $1 \leqslant i < l \leqslant k$. Due to the GLAV assertion $m_G \in \mathcal{M}_G$, by construction, $\mathcal{M}_G(D_G)$ introduces two fresh variables (without loss of generality, let denote them by z_i^j and z_l^i, where $j = l - 1$) such that (*i*) $\{V_1(z_i^j, a_{i,1}), \ldots, V_b(z_i^j, a_{i,b})\} \subseteq \mathcal{M}_G(D_G)$, (*ii*) $\{V_1(z_l^i, a_{l,1}), \ldots, V_b(z_l^i, a_{l,b})\} \subseteq \mathcal{M}_G(D_G)$, and (*iii*) $\{e(z_i^j, z_l^i), e(z_l^i, z_i^j)\} \subseteq \mathcal{M}_G(D_G)$. Observe that, for each node v_i, in this way there are $k-1$ fresh variables $z_i^1, z_i^2, \ldots, z_i^{k-1}$ representing v_i, i.e., $\{V_1(z_i^j, a_{i,1}), \ldots, V_b(z_i^j, a_{i,b})\} \subseteq \mathcal{M}_G(D_G)$ for each $j \in [1, k-1]$.

Consider now the function h from $dom(\mathcal{M}_k(D_k))$ to $dom(\mathcal{M}_G(D_G))$ such that (*i*) $h(c) = c'$, (*ii*) $h(y_i^l) = z_i^l$ for each $i \in [1,k]$ and for each $l \in [1,k-1]$, and (*iii*) $h(d_i^j) = a_{i,j}$ for each $i \in [1,k]$ and for each $j \in [1,b]$. It is straightforward to verify that h is a homomorphism witnessing that $(\mathcal{M}_k(D_k),(c)) \to (\mathcal{M}_G(D_G),(c'))$, as required.

"**If part:**" Suppose $(\mathcal{M}_k(D_k),(c)) \to (\mathcal{M}_G(D_G),(c'))$, and let h be the homomorphism witnessing this. Note that $h(c) = c'$ and, by construction, $(h(d_i^1),\ldots,h(d_i^b))$ is the encoding of a node v_i in circuit C_G for each possible $i \in [1,k]$. Furthermore, for each $i \in [1,k]$, ontology predicates $V_1,\ldots,V_b$ ensure that elements $y_i^1,\ldots,y_i^{k-1}$ are mapped to distinct fresh variables introduced by $\mathcal{M}_G(D_G)$ that actually denote the same node, because it must be the case that $V_1(h(y_i^1),h(d_i^1)) \in \mathcal{M}_G(D_G),\ldots,V_b(h(y_i^1),h(d_i^b)) \in \mathcal{M}_G(D_G),\ \ldots,\ V_1(h(y_i^{k-1}),h(d_i^1)) \in \mathcal{M}_G(D_G),\ldots,V_b(h(y_i^{k-1}),h(d_i^b)) \in \mathcal{M}_G(D_G)$. Now, it is easy to verify that if in $\mathcal{M}_G(D_G)$ we replace each variable z introduced by $\mathcal{M}_G(D_G)$ with the value $v = (a_1,\ldots,a_b)$ for which $V_1(z,a_1) \in \mathcal{M}_G(D_G),\ldots,V_b(z,a_b) \in \mathcal{M}_G(D_G)$, then, by looking at the extension of e, we obtain exactly the graph G represented by circuit C_G.

From the above observations, and the fact that the set of facts $\{e(h(y_i^j),h(y_l^i)) \mid 1 \leqslant i \leqslant j < k \text{ and } l = j+1\}$ must occur in $\mathcal{M}_G(D_G)$ due to the assumption that h witnesses $(\mathcal{M}_k(D_k),(c)) \to (\mathcal{M}_G(D_G),(c'))$, we immediately derive that the graph G represented by circuit C_G contains a clique of size k, as required. □

As for the coNEXPTIME-hardness of STCHAR(∅, GLAV, UCQ) case, it is possible to use exactly the same reduction provided above by discarding λ^- and considering only $\lambda^+ = \{(c)\}$. In particular, the set of certain answers of the query $q_k = query(\mathcal{M}_k(D_k),(c))$ w.r.t. Σ is always either $\{(c)\}$ or $\{(c),(c')\}$, and which one of the two depends on whether the graph represented by circuit C_G has a clique of size k or not. □

8. Conclusion and Future Work

In this paper, we have studied logical separability in OBDM. As a first contribution, we have illustrated a general framework for separability in OBDM, by also proposing natural relaxations of the classical separability notion (called here proper separation). In the general envision of separability as a tool for providing global post-hoc explanations of black-box models, we argue that such relaxations (especially the best approximations) are crucial in order to always be able to provide meaningful explanations even when proper separations do not exist. As a second contribution, by instantiating the general framework with the most common languages used in OBDM, we have provided a comprehensive study of three computational problems associated with the framework, namely Verification, Computation, and Existence. For the decision problems related to Verification and Existence, we have provided tight complexity results, whereas for the Computation problem we have devised two algorithms for computing the two forms of best approximated separations considered in this paper, thus proving they always exist.

We conclude the paper by discussing some interesting avenues for future work that deserve more investigation.

More expressive scenarios. It would be interesting to study extensions of the scenario considered in this paper for the computational problems. For example, one may consider more expressive target query languages that go beyond UCQ, in order to capture proper separations in more cases. Natural candidates for this are UCQ$^{\neq}$, i.e., UCQ with *inequalities*, First-order logic, and *EQL-Lite* [62]. However, differently from our case, by extending the considered scenario with one of these query languages, adopting or not the UNA makes a difference.

As other interesting extensions of the considered scenario, one may investigate highly expressive ontology languages, such as $\mathcal{SHIQ}$ [63] or even $\mathcal{SROIQ}$ [64], where the separability task has not yet been studied even in the ontology-enriched query answering setting (i.e., without considering mapping assertions).

Strong separability. In [3], separability comes into two flavours: *weak separability* and *strong separability*. In weak separability, which is the one we addressed in this paper, the requirement on the negative examples is that none of them is included in the set of certain answers of the separating query. On the other hand, in strong separability, the requirement on the negative examples is that all of them are included in the set of the certain answers of the *negation of the separating query*. Thus, a natural problem to be addressed in our considered OBDM scenario is strong separability, by considering relaxations also in this case.

Quantitative metrics for best approximations. In this work, we have defined best approximations of the proper separation notion by adopting the set-inclusion metric. Another common choice that would be interesting to investigate is the cardinality criteria. Moreover, for a more fine-grained usage of separability as a tool for explanation, it would be natural to assign *weights* to both positive and negative examples, and to consider such weights when computing best approximations of proper separations.

Restricted signature. A possible constraint that can be imposed when finding a separating query is that the query expression uses only a specific subset of the predicates available in the alphabet of the ontology $\mathcal{O}$. This may be important to meet some users' requirements regarding the separating query, as for example a user may ask for a separating query that does not make use of a specific concept. The separating task under restricted signature has been studied in [65] in the ontology-enriched query answering setting, in the particular case of expressive ontology languages, such as $\mathcal{ALC}$ or $\mathcal{ALCO}$. Typically, the computational complexity of checking for a proper separation increases under the restricted signature constraint. Thus, another notable direction is to study separability in the considered OBDM scenario under the restricted signature constraint.

Intelligible Queries. We point out that the separating query (or their best approximated versions) may sometimes be very hard to understand from an end-user perspective, because, for example, it may involve too many atoms in the body of its disjuncts. Thus, it would be natural to consider also the length of each disjunct, as well as the number of disjuncts, as additional parameters when considering approximations. This requires to consider novel definitions and techniques that may allow to obtain, from end users' perspectives, more intelligible separating queries.

Implementation. Finally, we mention that we are currently implementing the algorithms and techniques proposed in this paper using the OBDM engine Mastro [66]. The implementation we are working on follows the recently introduced *Human-in-the-loop Artificial Intelligence* (HitAI) paradigm [67]. By running some experiments with a first prototype in real-world settings, we observed as the above-mentioned issues for future work turn out to be crucial, especially the ones concerning intelligible queries and quantitative metrics.

References

[1] D.M.L. Martins, Reverse engineering database queries from examples: State-of-the-art, challenges, and research opportunities, *Information Systems* **83** (2019), 89–100. doi:https://doi.org/10.1016/j.is.2019.03.002.

[2] D. Mottin, M. Lissandrini, Y. Velegrakis and T. Palpanas, New Trends on Exploratory Methods for Data Analytics, *Proc. VLDB Endow.* **10**(12) (2017), 1977–1980–.

[3] J.C. Jung, C. Lutz, H. Pulcini and F. Wolter, Logical Separability of Incomplete Data under Ontologies, in: *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece*, 2020.

[4] M. Lenzerini, Ontology-based data management, in: *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM*, 2011.

[5] M. Lenzerini, Managing Data through the Lens of an Ontology, *AI Magazine* **39**(2) (2018), 65–74.

[6] G. Cima, F. Croce and M. Lenzerini, Query Definability and Its Approximations in Ontology-based Data Management, in: *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management*, ACM, 2021.

[7] A. Artale, D. Calvanese, R. Kontchakov and M. Zakharyaschev, The *DL-Lite* Family and Relations, *Journal of Artificial Intelligence Research* **36** (2009), 1–69.

[8] M.M. Zloof, Query-by-example: The Invocation and Definition of Tables and Forms, in: *Proceedings of the 1st International Conference on Very Large Data Bases*, VLDB '75, ACM, New York, NY, USA, 1975, pp. 1–24. ISBN 978-1-4503-3920-9. doi:10.1145/1282480.1282482.

[9] Q.T. Tran, C.-Y. Chan and S. Parthasarathy, Query Reverse Engineering, *The VLDB Journal* **23**(5) (2014).

[10] M. Arenas and G.I. Diaz, The Exact Complexity of the First-Order Logic Definability Problem, *ACM Trans. Database Syst.* **41**(2) (2016). doi:10.1145/2886095.

[11] P. Barceló and M. Romero, The Complexity of Reverse Engineering Problems for Conjunctive Queries, in: *20th International Conference on Database Theory (ICDT 2017)*, M. Benedikt and G. Orsi, eds, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 68, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2017, pp. 7:1–7:17. ISSN 1868-8969. ISBN 978-3-95977-024-8. doi:10.4230/LIPIcs.ICDT.2017.7. http://drops.dagstuhl.de/opus/volltexte/2017/7052.

[12] D.V. Kalashnikov, L.V.S. Lakshmanan and D. Srivastava, FastQRE: Fast Query Reverse Engineering, in: *Proceedings of the 2018 International Conference on Management of Data*, Association for Computing Machinery, 2018.

[13] D.M.L. Martins, Reverse engineering database queries from examples: State-of-the-art, challenges, and research opportunities, *Information Systems* **83** (2019).

[14] B. ten Cate and V. Dalmau, Conjunctive Queries: Unique Characterizations and Exact Learnability, in: *24th International Conference on Database Theory (ICDT 2021)*, Vol. 186, 2021.
[15] Y.Y. Weiss and S. Cohen, Reverse Engineering SPJ-Queries from Examples, in: *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, 2017.
[16] D. Angluin, Learning regular sets from queries and counterexamples, *Information and Computation* (1987).
[17] G. Cima, M. Lenzerini and A. Poggi, Semantic Characterization of Data Services through Ontologies, in: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI 2019)*, 2019, pp. 1647–1653.
[18] G. Cima, Abstraction in Ontology-based Data Management, PhD thesis, Sapienza University of Rome, 2020.
[19] G. Cima, M. Console, M. Lenzerini and A. Poggi, Abstraction in Data Integration, in: *Proceedings of the Thirty-Sixth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2021)*, IEEE, 2021, pp. 1–11.
[20] M. Arenas, G.I. Diaz and E.V. Kostylev, Reverse Engineering SPARQL Queries, in: *Proceedings of the 25th International Conference on World Wide Web*, 2016.
[21] M. Ortiz, Ontology-Mediated Queries from Examples: a Glimpse at the DL-Lite Case, in: *Proceedings of the Fifth Global Conference on Artificial Intelligence*, EPiC Series in Computing, Vol. 65, 2019, pp. 1–14.
[22] V. Gutiérrez-Basulto, J.C. Jung and L. Sabellek, Reverse Engineering Queries in Ontology-Enriched Systems: The Case of Expressive Horn Description Logic Ontologies, in: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 1847–1853. doi:10.24963/ijcai.2018/255.
[23] L. Bühmann, J. Lehmann, P. Westphal and S. Bin, DL-Learner Structured Machine Learning on Semantic Web Data, in: *Companion Proceedings of the The Web Conference 2018*, WWW '18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2018, pp. 467–471. ISBN 978-1-4503-5640-4. doi:10.1145/3184558.3186235.
[24] M. Funk, J.C. Jung, C. Lutz, H. Pulcini and F. Wolter, Learning Description Logic Concepts: When can Positive and Negative Examples be Separated?, in: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 1682–1688. doi:10.24963/ijcai.2019/233.
[25] J.C. Jung, C. Lutz, H. Pulcini and F. Wolter, Separating Data Examples by Description Logic Concepts with Restricted Signatures, in: *Proceedings of the Eighteenth International Conference on Principles of Knowledge Representation and Reasoning*, International Joint Conferences on Artificial Intelligence Organization, 2021.
[26] J. Lehmann and P. Hitzler, Concept learning in description logics using refinement operators, *Mach. Learn.* (2010).
[27] R. Confalonieri, T. Weyde, T.R. Besold and F.M. del Prado Martín, Using ontologies to enhance human understandability of global post-hoc explanations of black-box models, *Artif. Intell.* (2021).
[28] M.K. Sarker, N. Xie, D. Doran, M. Raymer and P. Hitzler, Explaining Trained Neural Networks with Semantic Web Technologies: First Steps, *Proceedings of the Twelfth International Workshop on Neural-Symbolic Learning and Reasoning.* (2017).
[29] G. Ciravegna, F. Giannini, M. Gori, M. Maggini and S. Melacci, Human-Driven FOL Explanations of Deep Learning, in: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, 2020.
[30] J. Liartis, E. Dervakos, O.M. Mastromichalakis, A. Chortaras and G. Stamou, Semantic Queries Explaining Opaque Machine Learning Classifiers, in: *Proceedings of the Workshop on Data meets Applied Ontologies in Explainable AI*, 2021.
[31] M. Law, A. Russo and K. Broda, Inductive Learning of Answer Set Programs, in: *Logics in Artificial Intelligence*, 2014.
[32] M. Funk, J.C. Jung and C. Lutz, Actively Learning ELI Queries under DL-Lite Ontologies, in: *Proceedings of the 34th International Workshop on Description Logics (DL 2021) part of Bratislava Knowledge September (BAKS 2021), Bratislava, Slovakia, September 19th to 22nd, 2021*, 2021.
[33] S. Abiteboul, R. Hull and V. Vianu, *Foundations of Databases*, Addison Wesley Publ. Co., 1995.
[34] F. Baader, D. Calvanese, D. McGuinness, D. Nardi and P.F. Patel-Schneider (eds), *The Description Logic Handbook: Theory, Implementation and Applications*, Cambridge University Press, 2003.
[35] M. Lenzerini, Ontology-based Data Management, in: *Proceedings of the Twentieth International Conference on Information and Knowledge Management (CIKM 2011)*, 2011, pp. 5–6. doi:10.1145/2063576.2063582.
[36] B. ten Cate and V. Dalmau, The Product Homomorphism Problem and Applications, in: *Proceedings of the Eigthteenth International Conference on Database Theory (ICDT 2015)*, LIPIcs, Vol. 31, 2015, pp. 161–176.
[37] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini and R. Rosati, Tractable Reasoning and Efficient Query Answering in Description Logics: The *DL-Lite* Family, *Journal of Automated Reasoning* **39**(3) (2007), 385–429.
[38] B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue and C. Lutz, OWL 2 Web Ontology Language Profiles (Second Edition), W3C Recommendation, World Wide Web Consortium, 2012, Available at http://www.w3.org/TR/owl2-profiles/.
[39] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini and R. Rosati, Linking Data to Ontologies, *Journal on Data Semantics* **X** (2008), 133–173.
[40] A. Doan, A.Y. Halevy and Z.G. Ives, *Principles of Data Integration*, Morgan Kaufmann, 2012. ISBN 978-0-12-416044-6.
[41] M. Lenzerini, Data Integration: A Theoretical Perspective., in: *Proceedings of the Twentyfirst ACM SIGACT SIGMOD SIGART Symposium on Principles of Database Systems (PODS 2002)*, 2002, pp. 233–246.
[42] M. Friedman, A. Levy and T. Millstein, Navigational Plans for Data Integration, in: *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI 1999)*, AAAI Press, 1999, pp. 67–73.
[43] D. Calvanese, G. De Giacomo, M. Lenzerini and M.Y. Vardi, Query Processing under GLAV Mappings for Relational and Graph Databases, *Proceedings of the Very Large Database Endowment* **6**(2) (2012), 61–72.
[44] A. Calì, G. Gottlob and M. Kifer, Taming the Infinite Chase: Query Answering under Expressive Relational Constraints, *Journal of Artificial Intelligence Research* **48** (2013), 115–174.

[45] R. Fagin, P.G. Kolaitis, R.J. Miller and L. Popa, Data Exchange: Semantics and Query Answering, *Theoretical Computer Science* **336**(1) (2005), 89–124.
[46] G. Cima, F. Croce and M. Lenzerini, QDEF and Its Approximations in OBDM, CoRR, arXiv.org e-Print archive, 2021.
[47] G. Cima, Preliminary Results on Ontology-based Open Data Publishing, in: *Proceedings of the Thirtieth International Workshop on Description Logics (DL 2017)*, CEUR Electronic Workshop Proceedings, http://ceur-ws.org/, Vol. 1879, 2017.
[48] C. Lutz, J. Marti and L. Sabellek, Query Expressibility and Verification in Ontology-based Data Access, in: *Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference (KR 2018)*, 2018, pp. 389–398.
[49] G. Cima, M. Lenzerini and A. Poggi, Non-Monotonic Ontology-based Abstractions of Data Services, in: *Proceedings of the Seventeenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2020)*, 2020, pp. 243–252.
[50] C.H. Papadimitriou and M. Yannakakis, The Complexity of Facets (and Some Facets of Complexity), *Journal of Computer and System Sciences* **28**(2) (1984), 244–259.
[51] L.J. Stockmeyer, The Polynomial-Time Hierarchy, *Theoretical Computer Science* **3**(1) (1976), 1–22.
[52] A. Bondy and M.R. Murty, *Graph Theory*, Graduate Texts in Mathematics, Springer, 2008.
[53] M.R. Garey, D.S. Johnson and L.J. Stockmeyer, Some Simplified NP-Complete Graph Problems, *Theoretical Computer Science* **1**(3) (1976), 237–267.
[54] J. Rothe, Exact complexity of Exact-Four-Colorability, *Information Processing Letters* **87**(1) (2003), 7–12.
[55] S.R. Buss and L. Hay, On Truth-Table Reducibility to SAT, *Information and Computation* **91**(1) (1991), 86–102.
[56] K.W. Wagner, Bounded Query Classes, *SIAM Journal on Computing* **19**(5) (1990), 833–846.
[57] K.W. Wagner, More Complicated Questions About Maxima and Minima, and Some Closures of NP, *Theoretical Computer Science* **51** (1987), 53–80.
[58] M.R. Garey and D.S. Johnson, "Strong" NP-Completeness Results: Motivation, Examples, and Implications, *Journal of the ACM* **25**(3) (1978), 499–508.
[59] C.H. Papadimitriou and M. Yannakakis, A Note on Succinct Representations of Graphs, *Information and Computation* **71**(3) (1986), 181–185.
[60] H. Galperin and A. Wigderson, Succinct Representations of Graphs, *Information and Computation* **56**(3) (1983), 183–198.
[61] G.S. Tseitin, On the Complexity of Derivation in Propositional Calculus, in: *Automation of Reasoning: 2: Classical Papers on Computational Logic 1967–1970*, 1983, pp. 466–483.
[62] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini and R. Rosati, EQL-Lite: Effective First-Order Query Processing in Description Logics, in: *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI 2007)*, 2007, pp. 274–279.
[63] I. Horrocks, U. Sattler and S. Tobies, Reasoning with Individuals for the Description Logic $\mathcal{SHIQ}$, in: *Proceedings of the Seventeenth International Conference on Automated Deduction (CADE 2000)*, D. McAllester, ed., Lecture Notes in Computer Science, Vol. 1831, Springer, 2000, pp. 482–496.
[64] I. Horrocks, O. Kutz and U. Sattler, The Even More Irresistible $\mathcal{SROIQ}$, in: *Proceedings of the Tenth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2006)*, 2006, pp. 57–67.
[65] J.C. Jung, C. Lutz, H. Pulcini and F. Wolter, Separating Data Examples by Description Logic Concepts with Restricted Signatures, in: *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, KR 2021, Online event, November 3-12, 2021*, 2021, pp. 390–399. doi:10.24963/kr.2021/37.
[66] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, R. Rosati, M. Ruzzi and D.F. Savo, The Mastro System for Ontology-based Data Access, *Semantic Web Journal* **2**(1) (2011), 43–53.
[67] F.M. Zanzotto, Viewpoint: Human-in-the-loop Artificial Intelligence, *Journal of Artificial Intelligence Research* **64** (2019), 243–252. doi:10.1613/jair.1.11345.